MODIFICATIONS TO PSU-WOPWOP FOR ENHANCED NOISE PREDICTION CAPABILITIES

A Thesis in
Aerospace Engineering
by
Benjamin Ariel Goldman

© 2013 Benjamin Ariel Goldman

Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

May 2013
The thesis of Benjamin Ariel Goldman was reviewed and approved* by the following:

Kenneth S. Brentner  
Professor of Aerospace Engineering  
Thesis Advisor

Phillip J. Morris  
Boeing/A. D. Welliver Professor of Aerospace Engineering

George A Lesieutre  
Professor of Aerospace Engineering  
Head of the Department of Aerospace Engineering

*Signatures are on file in the Graduate School.
Abstract

The motivation behind the research described in this thesis is to improve the science of acoustic prediction by expanding the range of predictable noise sources and developing more efficient computational methods for general prediction. These improvements have been implemented in the aeroacoustic prediction code PSU-WOPWOP.

This thesis focused on two general problems. The first problem deals with memory management for cases with large amounts of data. As the size and power of computers have grown, so too have the scale and resolution of domains applied to aeroacoustic prediction codes over the past several decades. Various methods have been used to overcome these memory demands, yet these methods simply bypass the underlying issue of inefficient data management. This thesis describes the development and validation of a computational algorithm that provides a solution to this problem by working with data from only a few source time steps at a time.

The second problem deals with the development and implementation of a package within PSU-WOPWOP that allows for rapid testing and analysis of rotors for FAA certification of civil helicopters. The metric used by the FAA for certification is the effective perceived noise level (EPNL) which is computed from a vehicle’s acoustic time history for a given flight condition. A significant contribution to the noise profile of such cases is the broadband noise component. Distinct from discrete-frequency noise sources, the physical mechanisms responsible for broadband noise are stochastic in nature and current prediction approaches are primarily semi-empirical. The use of multiple empirical models of varying fidelity can provide reasonable solutions across a wide range of cases while also forming an explicit hierarchy with which to compare with first-principle predictions as they are developed. The new noise certification package developed in PSU-WOPWOP consists of EPNL prediction, which includes
broadband noise prediction, as well as an easy-to-use flight path system developed to shorten the setup time for cases. Details of the package’s design and implementation are discussed in this thesis. In particular, the implementation of two semi-empirical broadband noise prediction models in PSU-WOPWOP is described in detail.
Table of Contents

List of Figures ix
List of Tables xii
List of Symbols xiii
Acknowledgments xvii

Chapter 1
Introduction 1
1.1 Problem Description ................................. 2
1.1.1 Algorithm Development .......................... 2
1.1.2 FAA Noise Certification .......................... 3
1.1.3 Broadband Noise Prediction ...................... 4
1.2 Motivation ........................................ 4
1.3 Contributions ..................................... 5
1.4 Reader’s Guide .................................... 6

Chapter 2
The Foundations of PSU-WOPWOP 8
2.1 Background ......................................... 8
2.1.1 The Ffowcs Williams-Hawkings Equation .......... 15
2.1.2 Permeable Surfaces ............................. 17
2.1.2.1 Permeable Surface Requirements .............. 17
2.2 PSU-WOPWOP Design ............................. 19
2.2.1 Data Organization ............................... 19
2.2.2 Data Types .................................... 20
2.3 Summary .......................................... 20
### Chapter 3
**Algorithm Development for Reduced Memory Requirements and Real Time Noise Prediction**

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Problem Description</td>
<td>23</td>
</tr>
<tr>
<td>3.2</td>
<td>Acoustic Prediction Algorithms</td>
<td>23</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Time in Acoustics</td>
<td>23</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Solutions to the Retarded Time Formulation</td>
<td>24</td>
</tr>
<tr>
<td>3.2.2.1</td>
<td>Run Time Efficiency</td>
<td>25</td>
</tr>
<tr>
<td>3.2.2.2</td>
<td>Memory Efficiency</td>
<td>26</td>
</tr>
<tr>
<td>3.3</td>
<td>An Algorithm for Reducing Memory Requirements</td>
<td>28</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Design</td>
<td>29</td>
</tr>
<tr>
<td>3.3.1.1</td>
<td>Difference Equations in Discrete Time</td>
<td>29</td>
</tr>
<tr>
<td>3.3.1.2</td>
<td>Analysis of Terms in Formulation 1A</td>
<td>30</td>
</tr>
<tr>
<td>3.4</td>
<td>Implementation of a New Computational Algorithm</td>
<td>31</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Consideration of Data Types</td>
<td>32</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Observer Time Interpolation</td>
<td>32</td>
</tr>
<tr>
<td>3.4.2.1</td>
<td>A New Interpolation Method in PSU-WOPWOP</td>
<td>34</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Parallel Computing</td>
<td>35</td>
</tr>
<tr>
<td>3.5</td>
<td>Results</td>
<td>36</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Setup</td>
<td>36</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Results</td>
<td>37</td>
</tr>
<tr>
<td>3.5.2.1</td>
<td>Run Time Efficiency</td>
<td>37</td>
</tr>
<tr>
<td>3.5.2.2</td>
<td>Memory Efficiency</td>
<td>39</td>
</tr>
<tr>
<td>3.6</td>
<td>Summary</td>
<td>40</td>
</tr>
</tbody>
</table>

### Chapter 4
**Broadband Noise Prediction**

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Problem Description</td>
<td>41</td>
</tr>
<tr>
<td>4.2</td>
<td>Semi-Empirical Broadband Noise Prediction Methods</td>
<td>42</td>
</tr>
<tr>
<td>4.2.1</td>
<td>The Pegg Method</td>
<td>42</td>
</tr>
<tr>
<td>4.2.1.1</td>
<td>Derivation of Pegg’s Method</td>
<td>43</td>
</tr>
<tr>
<td>4.2.1.2</td>
<td>Input Data</td>
<td>48</td>
</tr>
<tr>
<td>4.2.1.3</td>
<td>Implementation</td>
<td>50</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Method of Brooks, Pope and Marcolini</td>
<td>53</td>
</tr>
<tr>
<td>4.2.2.1</td>
<td>Derivation of BPM’s Method</td>
<td>53</td>
</tr>
<tr>
<td>4.2.2.2</td>
<td>Input Data</td>
<td>62</td>
</tr>
<tr>
<td>4.2.2.3</td>
<td>Implementation</td>
<td>67</td>
</tr>
</tbody>
</table>
# List of Figures

2.1 A cut-away view of a permeable acoustic data surface surrounding a helicopter rotor blade. .................................... 18
2.2 CFD grid and flow data to be used with a permeable ADS [10]... 19
2.3 An example of a blade formed by four structured patches. ...... 20

3.1 Schematic of source-time-dominant algorithm. .................... 25
3.2 Schematic of retarded-time algorithm. ............................ 25
3.3 Comparison of operation count between the retarded-time and source-time-dominant algorithms for ratio $nt/n\tau=3$ [12]. ... 27
3.4 Comparison of operation count between the retarded-time and source-time-dominant algorithms for ratio $nt/n\tau=0.2$ [12]. ... 27
3.5 Scalar function of source time plotted on an iso-surface of observer time. ......................................................... 28
3.6 Impact of source time steps and source nodes on memory requirements. ............................................................. 30
3.7 Flowchart of the numerical algorithm. ............................. 33
3.8 Example of signals arriving at different observer times from data at a single source time. ...................................... 34
3.9 Acoustic pressure matrix prior to PSU-WOPWOP v3.4 .......... 35
3.10 Acoustic pressure matrix updated in PSU-WOPWOP v3.4 ..... 35
3.11 Comparison of computation time for periodic data. ............ 38
3.12 Comparison of computation time for aperiodic data. .......... 38

4.1 Variation of broadband noise with Mach number and spanwise location on rotor blade. ........................................ 43
4.2 Determining the angle between the negative hub axis and the vector between the hub and the observer. ....................... 50
4.3 Normalized rotor broadband noise empirically determined spectrum shape. [19] .................................................. 52
4.4 Self noise mechanisms. [3] ........................................... 54
4.5 One-third-octave spectral shapes as functions of Strouhal and Reynolds numbers. \( R_c \) is the Reynolds number based on the airfoil chord.

4.6 Scaled 1/3-octave spectra of tripped BL airfoils at \( \alpha_t = 0^\circ (\alpha_* = 0^\circ) \). Symbols indicate approximate spectral peak locations. \( \alpha_t \) is the AOA referenced to the tunnel streamwise axis (deg), \( \alpha_* \) is the effective AOA corrected for open wind tunnel effects (deg).

4.7 Peak scaled levels for bluntness noise versus thickness ratio \( h/\delta^* \) determined from Fig. 4.15.

4.8 Spectral shape functions for TE bluntness noise.

4.9 Scaled 1/3-octave spectra for untripped boundary-layer airfoils at \( \alpha_t = 0^\circ (\alpha_* = 0^\circ) \). Symbols indicate approximate LBL-VS reference peak locations.

4.10 Scaled 1/3-octave spectra for untripped boundary-layer airfoils at various angles of attack. \( U=71.3 \text{ m/s}. \) Symbols indicate approximate LBL-VS reference peak locations.

4.11 One-third-octave spectral shape function \( G_1 \) for LBL-VS noise.

4.12 Peak scaled levels for LBL-VS noise versus Reynolds number. Data symbols are values of \( \alpha_* \) rounded off to the nearest degree.

4.13 Normalization of LBL-VS noise peak scaled levels by functions \( G_2 \). Data from Fig. 4.12.

4.14 Illustration of trailing-edge extensions and plates. Smooth surface transition was provided for all geometries.

4.15 TE-bluntness–vortex-shedding noise, data for untripped BL, and data with plate extensions (Fig. 4.14) attached.

4.16 Spectral density for TE noise for 60.96-cm-chord airfoil with various degrees of TE bluntness. Tripped BL; \( \alpha_t = 0^\circ; \Theta_e = 90^\circ \). \( \Theta_e \) is the angle from the source streamwise axis \( x \) to the observer, see Fig. 4.17. \( S(f) \) is the spectrum of self-noise with units Pa\(^2\)/Hz. The level is referenced to 1-Hz bandwidth. Data is from [25].

4.17 Flat plate in rectilinear motion.

4.18 Comparison of results for Pegg’s Method.

4.19 NACA 23012 and Boeing VR-12 airfoils.

4.20 Comparison of results for case 1 of BPM’s method. Comparison is between closed symbols.

4.21 Comparison of results for case 2 of BPM’s method. Comparison is between open symbols.

4.22 Comparison of results for case 3 of BPM’s method. Comparison is between closed symbols.
4.23 Comparison of results for case 4 of BPM’s method. Comparison is between open symbols. ........................................... 79
4.24 Comparison of results for case 5 of BPM’s method. Comparison is between open symbols. ........................................... 80
4.25 Comparison of results for case 6 of BPM’s method. Comparison is between open symbols. ........................................... 81
4.26 Comparison of results for case 7 of BPM’s method. Comparison is between closed symbols. ........................................... 82

5.1 FAA noise certification flight paths. ........................................... 85
5.2 Flight path motion versus aperiodic motion. ................................. 86
5.3 Perceived noise level as a function of total perceived noisiness [26]. 88
5.4 Example of flyover noise time history [26]. ................................. 89
5.5 Example of observer time-range extension. ................................. 91
5.6 Example of flight path extension. ........................................... 91
5.7 Example of default environment/global settings (left) and observer settings (right). ........................................... 92
5.8 Expanded view of the initial demonstration of the FAA civil noise certification package without observer time expansion plotted on the same scale as the full-length demonstration. ................. 95
5.9 Demonstration of the FAA civil noise certification package with observer time expansion. ........................................... 96

6.1 Routine currently used by PSU-WOPWOP to calculate pressure solutions. ........................................... 102
6.2 Alternative routine for calculating pressure solutions. .................... 102
List of Tables

4.1 Flight conditions for test case 1 with Pegg’s method . . . . . . . . 71
4.2 Flight conditions for test case 2 with Pegg’s method . . . . . . . . 72
4.3 Flight conditions for test cases with BPM’s method . . . . . . . . . 74
List of Symbols

\(c\)  
Sound speed in quiescent medium, m/s

\(\rho\)  
Density of quiescent medium, kg/m\(^3\)

\(\rho'\)  
Density of perturbation, kg/m\(^3\)

\(p'\)  
Acoustic pressure, \(p - p_0\), \((p' \approx c^2 \rho'\), outside of source region\), Pa

\(f = 0\)  
Function that describes the source surface, e.g., a rotor blade

\(dS\)  
Element of acoustic data surface

\(t\)  
Observer time, s

\(\tau\)  
Source time, s

\(x\)  
Observer position vector, with components \(x_i\), m

\(y\)  
Source position vector, with components \(y_i\), m

\(r\)  
Distance between observer and source, \(r = |x - y|\), m

\(\hat{n}\)  
Unit outward normal vector to surface, with components \(\hat{n}_i\)

\(P_{ij}\)  
Compressive stress tensor, Pa

\(l_i\)  
Components of local force intensity that act on the fluid, \(l_i = P_{ij}n_j\), Pa

\(l_r\)  
\(l_i\hat{n}_i\), Pa

\(\dot{l}_r\)  
\(\dot{l}_i\hat{n}_i\), Pa/s
\( l_r \)  \( \hat{l_i \hat{r}_i}, \text{Pa} \)

\( u_i \) Components of local fluid velocity, m/s

\( v_n \) Local normal velocity of source surface, m/s

\( \hat{n}_i \) Components of unit normal vector

\( \hat{v}_n \)  \( \hat{v}_i \hat{n}_i, \text{m/s}^2 \)

\( v_{\hat{n}} \)  \( v_i \hat{n}_i, \text{m/s}^2 \)

\( M \) Mach number of source with respect to a frame fixed to the undisturbed medium

\( M_i \)  \( \partial M_i / \partial \tau \)

\( \hat{r}_i \) Components of unit radiation vector

\( M_r \) Mach number of source in radiation direction, \( M_i \hat{r}_i \)

\( \hat{M}_r \)  \( \hat{M}_i \hat{r}_i \)

\( U \) Free-stream velocity, m/s

\( \alpha_t \) Airfoil angle of attack referenced to tunnel streamwise axis, deg

\( \alpha_* \) Effective aerodynamic angle of attack, corrected for open wind tunnel effects, deg

\( \sigma \) Rotor solidity, Blade area / rotor disk area

\( R_c \) Reynolds number based on chord length, \( cU / \nu \)

\( A_b \) Blade area, m²

\( A_c \) Correlation area, m²

\( A_t \) Blade tip area of vortex interactions, m²

\( c_L \) Local lift coefficient

\( C_L \) Average lift coefficient

\( V_{\text{tip}} \) Rotor blade tip velocity

\( T \) Thrust, N
$C_T$  Thrust coefficient, $\frac{T}{\rho A_b V_{tip}}$

$\delta$  Boundary-layer thickness, m

$\delta^*$  Boundary-layer displacement thickness, m

$\tilde{D}_h$  Directivity function for trailing-edge noise

$\tilde{D}_l$  Directivity function for translating dipole

$St$  Strouhal number defined for TBL-TE and separation noise scaling

$St'$  Strouhal number defined for LBL-VS noise

$St''$  Strouhal number defined for tip vortex formation noise

$St'''$  Strouhal number defined for TE-bluntness–VS noise

$\Psi$  Angle parameter related to surface slope at trailing-edge, deg

$\Theta$  Angle from source streamwise axis $x$ to observer, deg

$\Phi$  Angle from source lateral axis $y$ to observer, deg

**Abbreviations**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BL</td>
<td>Boundary layer</td>
</tr>
<tr>
<td>LBL</td>
<td>Laminar boundary layer</td>
</tr>
<tr>
<td>LE</td>
<td>Leading edge of airfoil blade</td>
</tr>
<tr>
<td>SPL</td>
<td>Sound pressure level, spectrum, dB (re $2 \times 10^{-5}$ Pa)</td>
</tr>
<tr>
<td>TBL</td>
<td>Turbulent boundary layer</td>
</tr>
<tr>
<td>TE</td>
<td>Trailing edge of airfoil blade</td>
</tr>
<tr>
<td>VS</td>
<td>Vortex shedding</td>
</tr>
</tbody>
</table>

**Subscripts**

$\text{ret}$  Quantity is evaluated at the retarded time, $\tau = t - r/c$
\[0\] Denotes fluid variable in quiescent medium

\[T\] Thickness noise component

\[L\] Loading noise component

\[i\] Array index denoting location in time

\[j\] Array index denoting location in space

TIP Tip of blade

TOT Total

e Retarded coordinate

p pressure side of airfoil

s Suction side of airfoil
Acknowledgments

I would not be where I am today without the support of my family. They have always encouraged and motivated me to be the best person I can be in every aspect of life. I am eternally grateful to them. To my mother for always reminding me to live and work as an honest man. To my father for the invaluable guidance he has provided throughout my academic career. To my sister for her love. And for trying to keep me humble.

I would like to thank my advisor Dr. Ken Brentner for his incredible mentoring. Never one to allow imprecise knowledge or work, his demand for quality has driven me to be a better engineer than I could have ever imagined.

This research and my contributions to PSU-WOPWOP would not have been possible without the work of its earlier authors: Guillaume Brès, Guillaume Perez, Leonard Lopes, Hsuan-Nien Chen, Christopher Hennes, Rui Cheng, Seongkyu Lee and James Erwin. Their talented work certainly made my job easier than it might have been and taught me an incredible amount about coding along the way. I am indebted to all of them. In particular, I would like to thank Chris Hennes for his tireless efforts to help me with every, and I do mean every, question and problem I have struggled with in my time at Penn State. I have never seen him lose his patience with me, even after hours of inquiry, although I’m fairly certain that would not be true had our communication not been over Skype. Aggravation is more easily concealed when texting than speaking.

I would like to thank my friends and colleagues at Penn State for their help and advice. The people at Penn State are amazing and supportive and I could not have completed this degree without their help. In particular, I would like to thank Jeswanth Mentey and Mihir Mistry for all their help and time spent discussing and exchanging ideas. True scholars, they never tired of technical discussions and always provided helpful comments and criticism.

I would also like to thank Dr. Philip J. Morris for his work in reading and editing this thesis.

Finally, I would like to thank Bell Helicopter Textron Inc. for providing fund-
ing for the research on the implementation of semi-empirical broadband noise methods and an FAA civil noise certification package in PSU-WOPWOP.
Dedication

To my parents. You have never stopped believing in me.
Chapter 1

Introduction

This thesis presents work done on two projects that expand the capabilities of the aeroacoustic prediction code PSU-WOPWOP. The first project deals with the development and implementation of an algorithm for reduced computer memory and real time noise prediction. The motivation for this research is to resolve the problems that arise from the large memory requirements typical of cases using permeable acoustic data surfaces (ADS). In recent years, the size of such cases has increased to the point that storage of the data in memory has become a considerable problem.

Various methods have been used in industry to manage this problem, but these methods provide only temporary, and often complicated, solutions without actually dealing with the underlying problem directly. A more permanent solution to the reduction of memory requirements must involve analysis of the fundamental requirements for aeroacoustic prediction. But the methods used to reduce memory requirements are very similar to those needed for real-time noise prediction. A new algorithm has been developed that solves both problems simultaneously.

The second project deals with the implementation and validation of an FAA civil noise certification package in PSU-WOPWOP. This package was designed with helicopters in mind but it can be used for any FAA civil certification case, including fixed wing aircraft, since the only difference in the certification requirements is the flight paths required for each vehicle type. This project also requires the implementation and validation of two semi-empirical broadband
noise prediction methods. The broadband noise predicted in these two models is generated from mechanisms excited by stochastic loading on a lifting surface such as a rotor blade or propeller. Prediction of these noise mechanisms from first principles is possible, but current methods are both computationally expensive and demanding. Semi-empirical methods sacrifice some level of generality, providing accurate predictions over a limited range of cases, for greatly reduced computational demands. As a first level of analysis, such methods offer reasonable predictive accuracy without needing expensive, time-consuming and bulky CFD solutions. Additionally, implementation of these models provides the groundwork for adding first-principles broadband noise prediction capability in future work.

1.1 Problem Description

This thesis presents solutions to the memory problems typical of cases with large computational domains. Also presented is the design for a comprehensive package which can quickly test and validate a rotor for FAA noise certification, which includes two semi-empirical broadband noise methods.

1.1.1 Algorithm Development

As the field of computational aeroacoustics has matured and grown in the last 20 years, so too has the scale and complexity of aeroacoustic computations. In particular, use of the Ffowcs Williams-Hawkins (FW-H) equation [1] for the computation of airframe noise and other broadband noise sources through the use of the permeable surface formulation of the FW-H equation has led to much larger computational demands. In these problems, while the data required by the FW-H code is still only needed on a surface (acoustic data surface or ADS), both high spatial and temporal resolution of the data are needed for accurate noise predictions. One of the biggest challenges has been the available computer memory. Indeed, some permeable surface noise prediction cases can require tens of gigabytes (GB) of data to describe all of the flow field information. Fortunately, not all of the data is needed in memory at the same time and re-
duction in the maximum computer memory used for a noise computation is possible. As an example, consider the prediction of noise in real time: relatively small portions of information are needed to predict an acoustic signal at any one time but those small portions, if saved up over time, would amass to become a large amount of data. This is the problem at hand: to try to reduce the amount of data stored in memory to what is need at the current time in the computation, rather than for all time.

In this thesis, methods to both reduce algorithm memory requirements and predict noise in real time will be discussed. As it turns out, these two topics are complementary as the capability to predict noise in real time calls for many of the same procedures needed to reduce the memory requirements of a FW-H solver.

1.1.2 FAA Noise Certification

An FAA noise certification package has been built into PSU-WOPWOP that enables rapid testing and evaluation of helicopters for FAA civil noise certification. This package predicts a vehicle’s effective perceived noise level (EPNL), a single-valued measure of a vehicle’s noise profile used by the FAA for civil noise certification. Distinct from discrete frequency noise, broadband noise provides a significant contribution to the total noise signal of modern helicopters. An accurate assessment of a helicopter’s noise profile in civil certification is incomplete without it. Yet prediction of helicopter broadband noise by first-principle methods and computational fluid dynamics models has not been widely attempted due to computational demands and complexity. To provide EPNL prediction that accounts for broadband noise, two semi-empirical broadband prediction methods have been included in the FAA noise certification package. The package also incorporates a new flight path system designed to help minimize the time needed to setup a case. Details of the package’s design and implementation will be discussed in this thesis.
1.1.3 Broadband Noise Prediction

Semi-empirical methods are very computationally efficient but their predictive capabilities are limited to the range of data upon which they are based. For preliminary analysis this approach is ideal. Two semi-empirical broadband prediction methods have been implemented into PSU-WOPWOP in this project: one by Pegg [2] and a second, more detailed, method by Brooks, Pope and Marcolini (BPM) [3]. These methods were chosen with the intention of being used as the first and second levels of analysis before a more rigorous analysis with data from computational fluid dynamics (CFD) is performed. A particular benefit with these methods is the fact that a majority of the information they require can be obtained from the geometry and loading data already used by PSU-WOPWOP. In this thesis, the implementation and validation of the two semi-empirical broadband methods will be discussed.

1.2 Motivation

Aircraft and rotorcraft are increasingly present in urban environments. Accurate noise prediction is of ever-greater importance and noise prediction tools that are both robust and easy-to-use are in high demand. The growing scale of computational domains applied to acoustic prediction codes has made the management of available computer memory a critical requirement. But memory management has implications beyond simply limiting data storage: real time noise prediction programs require a similar management system since they have to deal with a constant inflow of data. Thus, an algorithm that predicts noise while also limiting memory requirements is an ideal foundation for a real time noise prediction program.

Accurate noise prediction is also important for civil noise certification that requires that various measures of “acceptability” with respect to noise exposure are met. A significant noise source for civil certification is broadband noise, yet it is not currently well predicted using first-principles methods. Specifically, broadband noise must be included for accurate prediction of a vehicle’s effective perceived noise level (EPNL), the key metric in FAA civil noise certification. A
key goal in the present work has been to make the task of noise certification assessment less arduous. Much of the work required for noise certification tests can be automated since so many of the steps are always the same. Broadband noise can be included in this work by using semi-empirical methods that trade generality for significant reductions in computational demands and expense.

This thesis is concerned with providing both a solution to the memory management problem and a design methodology that can be generalized and applied to aeroacoustic prediction codes regardless of their framework. This thesis is also concerned with the development of a comprehensive FAA civil noise certification package that incorporates broadband noise prediction while generating quick and accurate noise predictions valid for FAA noise certification.

1.3 Contributions

The main contributions of this thesis are described below:

- **Algorithm for reduced memory requirements and real-time noise prediction**
  An algorithm that drastically reduces the memory requirements of aeroacoustic prediction codes has been developed. This algorithm takes advantage of the numerical methods used in the discrete-time implementation of the FW-H equation to reduce a case’s necessary memory requirements to their fundamental limit. In conjunction with modifications to the observer time interpolation routine, this algorithm also allows for real-time noise prediction.

- **Performance verification of algorithm through time trials**
  A series of time trials have been run to test the performance of the new algorithm. Results of the trials show that the algorithm maintains the speeds seen in previous versions of PSU-WOPWOP while drastically reducing the memory requirements.

- **Implementation of semi-empirical broadband noise prediction methods**
  Two semi-empirical broadband noise methods have been built into PSU-
These methods offer reasonable predictive capability with significantly lower computational demands. Implementation of the semi-empirical noise prediction methods also defines the framework in which a future first-principles broadband noise prediction method can be added to PSU-WOPWOP.

- **Design and implementation of FAA noise certification package**
  A package has been built into PSU-WOPWOP which allows for rapid testing and evaluation of rotors for FAA noise certification. The package has been designed to make the process of building and running cases for FAA noise certification as autonomous as possible.

### 1.4 Reader’s Guide

The remainder of this thesis is organized as follows:

- **Chapter 2** provides background on the acoustic theories and formulations used in the aeroacoustic prediction code PSU-WOPWOP in which the coding elements of this research were implemented.

- **Chapter 3** describes the analysis and development of the new algorithm for reduced memory and real time noise prediction. The design and implementation processes are also described. Also provided are a justification and the derivation of the algorithm. Lastly, results of the time trials designed to test the new algorithm are analyzed both for changes in run time and memory usage.

- **Chapter 4** describes the implementation and validation of the two semi-empirical broadband noise prediction methods. Discussion of the limitations of each method with respect to implementation in PSU-WOPWOP is also provided. The results from cases used to test the semi-empirical broadband methods are also presented.

- **Chapter 5** describes the design and implementation of an FAA civil noise certification package that incorporates the aforementioned semi-empirical
broadband noise methods. Descriptions of the simulations designed to test the FAA noise certification package are presented. Finally, results from each set of tests are presented.

- **Chapter 6** summarizes the results of this research and offers recommendations for future work.
The prediction of acoustic signals has been approached in numerous ways ranging from methods based entirely upon empirical results to those based upon more fundamental theories such as the Navier-Stokes (N-S) equation. Many of the acoustic prediction methods used today are based upon the Ffowcs Williams-Hawkings (FW-H) equation [1] which is an exact rearrangement of the N-S equations and the continuity equation into a wave equation. The components of this thesis were implemented and validated in a program which employs such a method. The program, PSU-WOPWOP, is a robust tool for aeroacoustic prediction and uses Farassat’s Formulation 1A of the FW-H equation to calculate acoustic pressure. Details of the derivation of Formulation 1A, may be found in Refs. [4], [5] and [6].

2.1 Background

The origins of aeroacoustic prediction can be traced back to Lighthill’s derivation of his acoustic analogy [7]. This approach is founded on the notion that the properties of a fluctuating fluid flow are never known very accurately, which inherently limits any estimation of sound levels. The analogy aims to quantify the strength of quadrupole (volume) source terms by comparing the behaviors of a real fluid and a uniform acoustic medium at rest, which have both been set to fluctuate by some driving force. For a real fluid, changes in pressure are
described through the combination of real stresses $p_{ij}$ and fluctuating Reynolds stresses $\rho v_i v_j$. In an acoustic medium, changes in pressure are proportional to the medium’s density variations where the proportionality constant is given by the square of the speed of sound $c^2$. The critical insight made by Lighthill is that density fluctuations in a real flow can be described exactly by those which would occur in a uniform acoustic medium being stressed by an external system whose strength is given the difference between the effective stresses in the real flow and those in the uniform acoustic medium at rest. This difference is called the Lighthill stress tensor and is defined as

$$T_{ij} = \rho v_i v_j + p_{ij} - c^2 \rho \delta_{ij} \tag{2.1}$$

This approach has a number of distinct advantages:

- Use of a uniform acoustic medium at rest as the free system avoids the need to consider modifications made to the medium after the sound has been produced such as sound propagation at variable speeds and its convection with the turbulent flow.

- The analogy is an exactly valid one since it assumes changes of mass and momentum are described by the conservation of mass (continuity) and momentum equations, respectively. It also makes no simplifying assumptions regarding real stresses or their relation to strain rates.

- The Lighthill stress tensor $T_{ij}$ accounts for the generation of sound, its convection with the flow, propagation with variable speed, and dissipation by both conduction and viscosity.

If external forces and sources of mass are ignored, sound propagation in a uniform medium is governed by the equations

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho v_i) = 0 \tag{2.2a}$$

$$\frac{\partial}{\partial t} (\rho v_i) + c^2 \frac{\partial \rho}{\partial x_i} = 0 \tag{2.2b}$$
\[ \frac{\partial^2 \rho}{\partial t^2} - c^2 \nabla^2 \rho = 0 \]  
(2.2c)

Equations 2.2a and 2.2b are the exact continuity equation, and an approximate equation of momentum, respectively. Equation 2.2c is the homogeneous wave equation, found by elimination of the momentum density term \( \rho v_i \) from the other two equations. The exact equation of momentum in an arbitrary continuous medium under no external forces is given in Reynolds’ form by

\[ \frac{\partial}{\partial t} (\rho v_i) + \frac{\partial}{\partial x_j} (\rho v_i v_j + p_{ij}) = 0 \]  
(2.3)

Thus for arbitrary fluid motion Eqns. 2.2 can be rewritten as

\[ \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho v_i) = 0 \]  
(2.4a)

\[ \frac{\partial}{\partial t} (\rho v_i) + c^2 \frac{\partial \rho}{\partial x_i} = \frac{\partial T_{ij}}{\partial x_j} \]  
(2.4b)

\[ \frac{\partial^2 \rho}{\partial t^2} - c^2 \nabla^2 \rho = \frac{\partial^2 T_{ij}}{\partial x_i \partial x_j} \]  
(2.4c)

where Eqn. 2.4c is an inhomogeneous wave equation.

In order to relate Lighthill’s stress tensor to the generation of sound by externally applied stresses, the theory of sound generation by simpler mechanisms must be reviewed. Eqns. 2.2 assume that the total mass of a system is constant but this assumption is not necessary. If fluctuating sources of additional mass are continuously distributed throughout part of a medium such that mass \( Q(x, t) \) per unit volume per unit time is introduced at \( x \) at time \( t \) then Eqns. 2.2 become

\[ \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho v_i) = Q(x, t) \]  
(2.5a)

\[ \frac{\partial}{\partial t} (\rho v_i) + c^2 \frac{\partial \rho}{\partial x_i} = 0 \]  
(2.5b)

\[ \frac{\partial^2 \rho}{\partial t^2} - c^2 \nabla^2 \rho = \frac{\partial}{\partial t} Q(x, t) \]  
(2.5c)
and the density field is given by the volume integral

$$\rho - \rho_0 = \frac{1}{4\pi c^2} \int_V \frac{\partial}{\partial t} Q(y, t - \frac{|x-y|}{c}) \frac{dy}{|x-y|}$$  \hspace{1cm} (2.6)$$

If, instead, sources of mass are not present and the sound is generated by a fluctuating external force field $F_i$ per unit volume in part of the medium, then Eqns. 2.2 become

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho v_i) = 0 \hspace{1cm} (2.7a)$$

$$\frac{\partial}{\partial t} (\rho v_i) + c^2 \frac{\partial \rho}{\partial x_i} = F_i(x, t) \hspace{1cm} (2.7b)$$

$$\frac{\partial^2 \rho}{\partial t^2} - c^2 \nabla^2 \rho = -\frac{\partial}{\partial x_i} F_i(x, t) \hspace{1cm} (2.7c)$$

Therefore, in terms of the density fluctuations produces, such a fluctuating force field is equivalent to a source distribution whose strength per unit volume is equal to the flux of force inwards, $-\partial F_i/\partial x_i$. The sound field generated is actually a field of dipoles with its axis in the $x_1$ direction and strength $F_1$ per unit volume, together with two similar fields whose axes lie in the $x_2$ and $x_3$ directions. For a general volume distribution of dipoles, the density field is given by

$$\rho - \rho_0 = -\frac{1}{4\pi c^2} \frac{\partial}{\partial x_i} \int_V F_i(y, t - \frac{|x-y|}{c}) \frac{dy}{|x-y|}$$  \hspace{1cm} (2.8)$$

In an unbounded medium, an expression analogous to Eqn. 2.8 can then be written for the field of a continuous distribution of quadrupoles with tensor strength density $T_{ij}$ as

$$\rho - \rho_0 = \frac{1}{4\pi c^2} \frac{\partial^2}{\partial x_i \partial x_j} \int T_{ij}(y, t - \frac{|x-y|}{c}) \frac{dy}{|x-y|}$$  \hspace{1cm} (2.9)$$

A crucial result, Eqn. 2.9 shows that, if $T_{ij}$ is known, then the density field and ultimately the acoustic field for a distribution of quadrupole sources can be determined.

Although important, Eqn. 2.9 is not itself sufficient for general acoustic prediction because it explicitly assumes the acoustic medium to be unbounded,
that is, without solid/physical boundaries to influence the acoustic field. Curle extended Lighthill’s analogy to include solid boundaries [8]. Acoustic fields are influenced by solid boundaries in three ways:

1. Sound generated by quadrupoles will be reflected and diffracted by solid boundaries.

2. Quadrupoles are no longer distributed over all of space, but rather only throughout the region external to the solid boundaries.

3. Interaction between quadrupoles and the solid boundaries will result in a distribution of dipoles at the boundaries.

The inclusion of solid boundaries in the domain means that the influence of both dipoles and quadrupoles must be accounted for. The most general solution to Eqns. 2.4 is was provided by Stratton [9]:

\[
\rho - \rho_0 = \frac{1}{4\pi c^2} \int_V \frac{\partial^2 T_{ij}}{\partial y_i \partial y_j} \frac{dy}{|x - y|} + \frac{1}{4\pi} \int_S \left\{ \frac{1}{r} \frac{\partial \rho}{\partial n} + \frac{1}{r^2} \frac{\partial^2 \rho}{\partial n^2} + \frac{1}{cr} \frac{\partial^2 \rho}{\partial n \partial t} \right\} dS(y)
\]

(2.10)

The first integral is taken over the volume \( V \) external to the solid boundaries which accounts for the contribution from the quadrupole while the second integral is taken over the surface \( S \) of the solid boundaries to capture the dipole contribution. The radiation distance from a source to an observer is given by \( r = |x - y| \). It follows that the time taken for a signal to propagate this distance is given by \( r/c = t - \tau \) where \( c \) is the speed of sound, \( t \) is the signal’s arrival time and \( \tau \) is the signal’s retarded/source time. In Eqn. 2.10 the quantities \( \frac{\partial^2 T_{ij}}{\partial y_i \partial y_j} \), \( \frac{\partial \rho}{\partial n} \), \( \rho \) and \( \frac{\partial \rho}{\partial t} \) are taken at the retarded time \( \tau = t - r/c \).

The volume integral in Eqn. 2.10 can be manipulated by twice applying the divergence theorem, in effect splitting up the quadrupole’s contribution between its influence in the volume and its influence on solid boundaries. The first application of the divergence theorem gives

\[
\int_V \frac{\partial^2 T_{ij}}{\partial y_i \partial y_j} \frac{dy}{r} - \frac{\partial}{\partial x_i} \int_V \frac{\partial T_{ij}}{\partial y_j} \frac{dy}{r} = \int_V \frac{\partial}{\partial y_i} \left[ \frac{\partial T_{ij}}{\partial y_j} / r \right] dy
\]

(2.11)
where \( l_i \) are the direction cosines of the outward normal from the fluid, \( \mathbf{n} = (l_1, l_2, l_3) \). Application of the divergence theorem a second time gives

\[
\int_V \frac{\partial T_{ij}}{\partial y_j} \frac{d\mathbf{y}}{\mathbf{r}} - \frac{\partial}{\partial x_i} \int_V T_{ij} \frac{d\mathbf{y}}{\mathbf{r}} = \int_S l_i T_{ij} \frac{dS(\mathbf{y})}{\mathbf{r}} \tag{2.12}
\]

Rearrangement of Eqns. 2.11 and 2.12 yields

\[
\int_V \frac{\partial^2 T_{ij}}{\partial y_i \partial y_j} \frac{d\mathbf{y}}{\mathbf{r}} = \frac{\partial^2}{\partial x_i \partial x_j} \int_V \frac{T_{ij}(\mathbf{y}, t - \frac{r}{c})}{\mathbf{r}} d\mathbf{y} \tag{2.13}
\]

\[
+ \frac{\partial}{\partial x_i} \int_S l_i T_{ij}(\mathbf{y}, t - \frac{r}{c}) \frac{dS(\mathbf{y})}{\mathbf{r}} + \int_S l_i \frac{\partial T_{ij}}{\partial y_j} \frac{dS(\mathbf{y})}{\mathbf{r}}
\]

The surface integral in Eqn. 2.10 can be expressed in a more convenient form:

\[
\int_S \left\{ \frac{1}{\mathbf{r}} \frac{\partial \rho}{\partial \mathbf{n}} + \frac{1}{\mathbf{r}^2} \frac{\partial}{\partial \mathbf{n}} \rho + \frac{1}{c \mathbf{r}} \frac{\partial \rho}{\partial t} \right\} dS(\mathbf{y}) = \int_S l_i \left\{ \frac{1}{\mathbf{r}} \frac{\partial \rho}{\partial y_i} + \frac{1}{\mathbf{r}^2} \frac{\partial}{\partial y_i} \rho + \frac{1}{c \mathbf{r}} \frac{\partial \rho}{\partial t} \right\} dS(\mathbf{y}) \tag{2.14}
\]

\[
= \int_S l_i \frac{\partial}{\partial y_i} (\rho \delta_{ij}) dS(\mathbf{y}) - \int_S l_i \left\{ \frac{1}{\mathbf{r}^2} \frac{\partial}{\partial x_i} \rho + \frac{1}{c \mathbf{r}} \frac{\partial \rho}{\partial t} \right\} dS(\mathbf{y})
\]

The last integral is found due to the fact that retarded times apply to all foregoing work, as well as

\[
\frac{\partial}{\partial x_i} \left\{ \frac{1}{\mathbf{r}} f \left( t - \frac{\mathbf{r}}{c} \right) \right\} = - \left\{ \frac{1}{\mathbf{r}^2} f + \frac{1}{c \mathbf{r}} f' \right\} \frac{\partial r}{\partial x_i} \tag{2.15}
\]
Equations 2.13 and 2.14 can be substituted into Eqn. 2.10 to give

\[ \rho - \rho_0 = \frac{1}{4\pi c^2} \frac{\partial^2}{\partial x_i \partial x_j} \int_V \frac{T_{ij}(y, t - \frac{r}{c})}{r} \, dy \]

\[ + \frac{1}{4\pi c^2} \int_S l_i \frac{1}{r} \frac{\partial}{\partial y_j} (T_{ij} + c\rho \delta_{ij}) \, dS(y) \]

\[ + \frac{1}{4\pi c^2} \frac{\partial}{\partial x_i} \int_S l_j \frac{1}{r} (T_{ij} + c^2 \rho \delta_{ij}) \, dS(y) \]

which can be simplified slightly by substituting for \( T_{ij} = \rho v_i v_j + p_{ij} - c^2 \rho \delta_{ij} \)

\[ \rho - \rho_0 = \frac{1}{4\pi c^2} \frac{\partial^2}{\partial x_i \partial x_j} \int_V \frac{T_{ij}(y, t - \frac{r}{c})}{r} \, dy \]

\[ + \frac{1}{4\pi c^2} \int_S l_i \frac{1}{r} \frac{\partial}{\partial y_j} (\rho v_i v_j + p_{ij}) \, dS(y) \]

\[ + \frac{1}{4\pi c^2} \frac{\partial}{\partial x_i} \int_S l_j \frac{1}{r} (\rho v_i v_j + p_{ij}) \, dS(y) \]

Rearrangement of the conservation of momentum equation shows that

\[ l_i \frac{\partial}{\partial y_j} (\rho v_i v_j + p_{ij}) = -l_i \frac{\partial}{\partial t} (\rho v_i) \]

(2.18)

and provided that there is zero normal velocity at the solid boundary then \( l_i v_i \equiv 0 \). With this assumption Eqn. 2.17 reduces to

\[ \rho - \rho_0 = \frac{1}{4\pi c^2} \frac{\partial^2}{\partial x_i \partial x_j} \int_V \frac{T_{ij}(y, t - \frac{r}{c})}{r} \, dy + \frac{1}{4\pi c^2} \frac{\partial}{\partial x_i} \int_S \frac{1}{r} l_j p_{ij} \, dS(y) \]

\[ = \frac{1}{4\pi c^2} \frac{\partial^2}{\partial x_i \partial x_j} \int_V \frac{T_{ij}(y, t - \frac{r}{c})}{r} \, dy - \frac{1}{4\pi c^2} \frac{\partial}{\partial x_i} \int_S \frac{1}{r} l_j p_{ij} \, dS(y) \]

(2.19)

where

\[ P_i = -l_j p_{ij} \]

(2.20)
Equation 2.19 is the fundamental result of Curle’s work. This work modifies Lighthill’s acoustic analogy to include solid boundaries in the acoustic field. The influence of the solid boundaries is described by the surface integral in Eqn. 2.19 which is exactly equivalent to the sound generated in a medium at rest by a distribution of dipoles with strength \( P_i \) per unit area. From Eqn. 2.25, \( P_i \) is the force per unit area exerted on the fluid by the solid boundaries in the \( x_i \) direction.

### 2.1.1 The Ffowcs Williams-Hawkings Equation

In 1969, Ffowcs Williams and Hawkings published “Sound Generation by Turbulence and Surfaces in Arbitrary Motion” [1] which extended the Lighthill acoustic analogy [7] to moving surfaces. The equation they developed, the Ffowcs Williams-Hawkings Equation or simply FW-H, is an exact reformulation of the continuity and Navier-Stokes equations using generalized functions,

\[
\begin{align*}
\begin{array}{c}
\Box^2 p'(x, t) = \\
= \frac{\partial}{\partial t}\left\{ \rho_0 v + \rho (u_n - v_n) \delta(f) \right\} \\
+ \frac{\partial}{\partial x_i} \left\{ \Delta P_{ij} \mathbf{n}_j + \rho u_i (u_n - v_n) \delta(f) \right\} \\
+ \frac{\delta^2}{\partial x_i \partial x_j} [T_{ij} H(f)]
\end{array}
\end{align*}
\]

(2.21)

The three terms on the right hand side of the FW-H equation are referred to as the monopole, dipole and quadrupole terms, respectively, due to their mathematical structure. The generalized differentiation is implicit in the first two terms due to the presence of the Dirac delta function \( \delta(f) \). Each of the terms in Eqn. 2.21 is a function of \( f \), which is defined as

\[
f = \begin{cases} 
< 0 & \text{Within the data surface} \\
0 & \text{On the data surface} \\
> 0 & \text{Outside the data surface}
\end{cases}
\]

(2.22)

The monopole and dipole source terms are seen to act on the surface \( f = 0 \) as can be recognized by the presence of the Dirac delta function \( \delta(f) \) which for an
arbitrary function \( h(x) \) is defined by

\[
h(x_0) = \int h(x) \delta(x - x_0)dx
\] (2.23)

The quadrupole/volume source term acts in the volume surrounding the surface given the presence of the Heaviside function which is defined by

\[
H(f) = \begin{cases} 
0 & f < 0 \\
1 & f > 0 
\end{cases}
\] (2.24)

An integral form of the FW-H equation solution was developed by Farassat [5] which neglects the quadrupole. Intended primarily for subsonic source motion, Formulation 1A describes the acoustic pressure fluctuation produced by an acoustic data surface (ADS) as a sum of two terms, monopole (thickness) \( p'_T \) and dipole (loading) \( p'_L \):

\[
p'(x, t) = p'_T(x, t) + p'_L(x, t)
\] (2.25)

where the thickness \( p'_T \) and loading \( p'_L \) contributions are calculated from

\[
4\pi p'_T(x, t) = \int_{f=0}^{l} \left[ \frac{\rho_0(\dot{v}_n + \bar{v}_n)}{r(1 - M_r)^2} \right]_{\text{ret}} dS + \int_{f=0}^{l} \left[ \frac{\rho_0\bar{v}_n(rM_r + c(M_r - M^2))}{r^2(1 - M_r)^3} \right]_{\text{ret}} dS
\] (2.26a)

\[
4\pi p'_L(x, t) = \frac{1}{c} \int_{f=0}^{l} \left[ \frac{l_r}{r(1 - M_r)^2} \right]_{\text{ret}} dS + \int_{f=0}^{l} \left[ \frac{l_r - l_M}{r^2(1 - M_r)^2} \right]_{\text{ret}} dS + \frac{1}{c} \int_{f=0}^{l} \left[ \frac{l_r rM_r + c(M_r - M^2)}{r^2(1 - M_r)^3} \right]_{\text{ret}} dS
\] (2.26b)

respectively. PSU-WOPWOP implements Formulation 1A numerically: integrals are evaluated as discrete sums while derivatives and rates-of-change are calculated using difference equations (see Section 3.3.1.1). The terms in Formulation 1A are evaluated at the retarded/source time as denoted by the \( \text{ret} \) sub-
script in each integral. The integrals are obtained by summing the contributions of all the sources/nodes on the acoustic data surface at a particular observer time. Terms which include a factor of $r^{-1}$ are far-field terms since, for large $r$, their strength is relatively higher than those terms that include a factor of $r^{-2}$, which are near-field terms. A Doppler factor $(1 - M_r)^{-1}$ is found in every term of Formulation 1A. This is raised to either the $2^{nd}$ or $3^{rd}$ power depending on the term; terms that include the to the higher power are dominant for cases where $(1 - M_r) \to 0$.

2.1.2 Permeable Surfaces

The quadrupole source term is absent from Formulation 1A for two reasons:

- The volume integral necessary to compute the term is computationally demanding.

- The quadrupole contribution can be captured through manipulation of the terms found in Formulation 1A.

Mathematically, there is no reason why a quadrupole term is needed. The acoustic analogies made by Lighthill and Curle describe the influence of acoustic sources using source distributions equivalent to, but not necessarily identical to, the real sources. The influence of the quadrupole term can be captured by the monopole and dipole terms if the boundaries of the ADS surround the volume in which the quadrupole acts. Unlike the solid boundaries described by Curle, the boundaries of such an ADS are permeable, with a non-zero normal flow velocity on its surface. Such surfaces that do not correspond to physical boundaries, like the one in Fig. 2.1, are called permeable acoustic data surfaces (ADS).

2.1.2.1 Permeable Surface Requirements

The use of permeable ADS eliminates the need to take the volume integrals necessary to compute the quadrupole term, but the requirement that all the acoustic sources be inside the ADS places a larger demand on CFD to compute the flow
Figure 2.1: A cut-away view of a permeable acoustic data surface surrounding a helicopter rotor blade.

field accurately (including acoustic propagation) out to the ADS. Moreover, experience and application of permeable acoustic data surfaces have shown that both high spatial and temporal resolution in the CFD grid are needed to accurately predict noise. The left-hand image in Fig. 2.2 shows a typical CFD mesh used to obtain flow data for use with a permeable ADS. To capture the influence of the quadrupole the flow data must be sufficiently accurate all the way to the permeable surface. This means that the CFD grid needs to have high spatial and temporal resolution in flow regions both near and far away from the physical surface in order to obtain accurate flow solutions. As a result, the CFD computations for these cases can take weeks or even months to run. This need for high spatial and temporal resolution results in large input data files. To be sure, the increasingly prevalent use of permeable surfaces in industry was a leading factor in the decision to develop an algorithm which reduces the memory requirements of existing aeroacoustic prediction codes.
2.2 **PSU-WOPWOP Design**

PSU-WOPWOP is an aeroacoustic prediction code that uses Formulation 1A of the FW-H equation to predict noise. It includes many powerful features, but requires just a short list of information to predict noise: 1) geometry data, which is a time history of the physical location and orientation of each node/source on the ADS, 2) loading data, which is a time history of the load exerted by each node/source on the surrounding fluid, and 3) A file describing the setup of the case including, but not limited to, orientation and motion of a vehicle e.g., a helicopter and its ADS e.g., blades.

2.2.1 **Data Organization**

PSU-WOPWOP uses an object called a “patch” to store surface geometry and loading data for a single surface. The ADS for a rotor blade can be made up of one or multiple patches. One common approach is to use of four separate patches for a blade’s upper surface, lower surface, inboard tip and outboard tip (see Fig. 2.3). This approach is very compatible with the surface grids used in CFD calculations but it also makes analysis much simpler. By avoiding any restrictions in the layout of an ADS, the noise contribution from any one section (such as the tip of a blade) can be analyzed independently of the others and separate patches can be modified without having to rebuild an entire wing.
2.2.2 Data Types

PSU-WOPWOP is designed to handle three different types of loading data: pressure data, loading vectors and flow data. Gauge pressure data $p'$ is provided in units of force/area acting in the surface normal $\hat{n}$ direction such that the resulting loading vector is given by $L = p'\hat{n}$. Loading vectors $L$ are provided in units of force and applied along the vector $L = [L_x, L_y, L_z]$. Flow data is made up of the flow variables $(\rho, \rho u, \rho v, \rho w, p')$ and is used in conjunction with permeable ADS. From this data, the loading vectors can be obtained by

$$L = p'\hat{n} + (u_n - v_n)[\rho u, \rho v, \rho w]$$ \hspace{1cm} (2.27)

with the values of the flow variables taken on the permeable ADS.

2.3 Summary

Background on the acoustic theories and formulations used in the aeroacoustic prediction code PSU-WOPWOP have been presented. The background begins with Lighthill’s acoustic analogy and proceeds through Curle’s extension of the acoustic analogy, the development of the FW-H equation, ending with the development of Formulation 1A. The function and application of permeable surfaces in PSU-WOPWOP have also been described.

The next chapter describes the analysis and development of the new algorithm for reduced memory and real time noise prediction. Details of its deriva-
tion, design and implementation are provided. Also provided are results of time trials which have been used to test the new algorithm for changes in both run time and memory usage.
Chapter 3

Algorithm Development for Reduced Memory Requirements and Real Time Noise Prediction

The following chapter defines the problem of large data requirements for computational aerocoustic prediction. The major topics discussed are:

1. Problem Description: The problem of managing data requirements when predicting noise for large computational domains is detailed. Analysis of the problem and potential solutions are described. Details of the implementation of a new computational algorithm, which provides a solution to the problem, are provided.

2. Algorithm Strategies for Reducing Memory Requirements: Analysis of the memory management problem is presented and the fundamental causes are discussed. Discussion of potential solutions is provided and details of the chosen solution are given.

3. Implementation of a New Computational Algorithm: Implementation of the new computational algorithm is described. The technical challenges and consequences of its implementation are discussed in detail.
3.1 Problem Description

The situation considered here is a case in which the input data necessary to predict the noise from a source, typically a rotor or propeller, is greater than the 2GB allowable memory limit imposed by 32-bit compilers. As a result, if a program attempts to store (allocate) more than 2GB of data in RAM at once, it will crash. Aeroacoustic prediction is very computationally demanding, but as the processing speed and storage capacity of computers have grown in recent years, so too has the size of acoustic domains needed for accurate noise prediction from complex source regions. The memory limitation is usually reached in large cases that use permeable acoustic data surfaces (ADS) to predict noise. Greater detail regarding permeable ADS is given in section 2.1.1, but experience has shown that permeable ADS cases typically require very high spatial and temporal data resolution to accurately predict the noise. The amount of data required to store this information is on the order of the product of the number of spatial grid points and the number of time steps. A new algorithm has been developed which drastically reduces memory requirements. Additionally, this algorithm is ideally suited for use in real-time noise prediction.

3.2 Acoustic Prediction Algorithms

3.2.1 Time in Acoustics

Acousticians are concerned with noise at two instances of time: the time when a sound signal is emitted by a source (source time, $\tau$) and the time when the signal arrives at an observer (observer time, $t$). These two times are fundamentally linked by the propagation time, i.e. the time required for an acoustic signal to propagate at the speed of sound $c$ over the distance $r$ from the source $y(\tau)$ to the observer $x(t)$. This relationship is given by the equation:

$$t = \tau + \frac{|x(t) - y(\tau)|}{c} = \tau + \frac{r}{c}$$  \hspace{1cm} (3.1)
The retarded time $\tau$ can be calculated from a chosen observer time $t$ by simply by rearranging Eqn. 3.1:

$$\tau = t - \frac{r}{c}$$  \hfill (3.2)

### 3.2.2 Solutions to the Retarded Time Formulation

Methods for acoustic prediction have been developed using both Eqns. 3.1 and 3.2 as their foundation. Algorithms which employ Eqn. 3.1, which calculate observer time from source time, are referred to as source-time-dominant algorithms, shown schematically in Fig. 3.1. Algorithms that calculate source time from observer time as in Eqn. 3.2 are typically called retarded-time algorithms, shown schematically in Fig. 3.2. However, solutions to Formulation 1A require that the integrand is computed at the retarded time. Both the source-time-dominant and retarded-time algorithms are approaches to the solution of the retarded time formulation.

Source-time-dominant algorithms begin by choosing a source time of interest. For each grid point at the chosen source time the following terms must be obtained: the observer arrival time and position, the node’s velocity and acceleration at the source time and ultimately the integrand of Formulation 1A. This procedure is performed for every source time of interest. Once completed, there is a final necessary procedure: interpolation of acoustic pressure from every point on the ADS so it can be added at the same observer time.

Retarded-time algorithms begin by choosing an observer time and position of interest. Similar to the source-time-dominant algorithm, the source time and position, velocity, acceleration and integrand must be computed for each node at the given observer time. However, because the observer time has been chosen explicitly, the pressure arrays can be summed directly. This procedure is performed for every observer time of interest.

A first look at these two types of algorithms reveals little about their computational efficiency, particularly since a great deal of similarity exists between them. In fact, the computational costs of the two algorithm types are significantly different.
3.2.2.1 Run Time Efficiency

Analysis has shown that the efficiency of each algorithm is heavily dependent upon two factors, the number of points in the observer-time history and the number of necessary coordinate transformations. Brès et al. [11] showed that the source-time-dominant algorithm requires significantly fewer operations (as compared to the retarded time algorithm) when the number of points on the
acoustic data surface or in the observer time history is large, or when there is a large number of coordinate transformations. Fig. 3.3 shows this relationship for a case whose observer-time resolution is three times that of the source time resolution. Fig. 3.4 shows the case; i.e., the source time resolution is five times greater than the observer-time resolution. From Figs. 3.3 and 3.4 it is clear that the source-time-dominant algorithm is much more computationally efficient, requiring significantly fewer floating-point operations, particularly for cases with a larger number of coordinate transformations. This analysis was the basis for the decision to use the source-time-dominant algorithm in PSU-WOPWOP. But the source-time-dominant algorithm is also advantageous when the number of points on the acoustic data surface is high. This is because, in the retarded time formulation, any coordinate transformations must be computed multiple times for each point on the acoustic data surface at each observer time step. In the source-time-dominant approach, the coordinate transformations are performed only once per source time (once for all points on the ADS as compared to several computations for each point on the ADS). While it is true that the observer time and position computation must be performed for every point on the acoustic data surface, these computations are typically very simple and for stationary observers the position does not need to be computed at all. The choice of the algorithm can lead to dramatic changes in computation effort. Similar changes in the numerical algorithm can lead to substantial changes in the memory requirements during the computation.

3.2.2.2 Memory Efficiency

Much of what makes the source-time-dominant algorithm more computationally efficient also gives it the potential to be vastly more memory efficient than the retarded-time algorithm. The greatest burden to memory is storage of the information needed to predict the integrand terms in FW-H at each source time. In PSU-WOPWOP this information is organized in two parts: (1) a geometric description of the acoustic data surface (ADS), which may be a function of time, and (2) a history of the loading information on the ADS. The retarded-time algorithm inherently calls for continual access to a large portion of information about the ADS because the acoustic signal at a single observer time is made
up of contributions from a wide range of source times. This concept can be visualized easily by plotting an iso-surface of observer time (The ADS location corresponding to a single observer time, which is the emission surface) colored by the value source time (Fig. 3.5). In this example the ADS is a rotor blade with a rectangular planform but because the distance to the observer is not the same from root to tip the iso-surface is curved and the acoustic signal at this observer time includes information from multiple source time steps. On the other hand,
because the source-time-dominant algorithm used in PSU-WOPWOP computes the terms of the integrand for source times sequentially it theoretically needs access to ADS information at just one source time step at a time. This subtle difference between the two algorithms allows designers to make the memory requirements very low for a FW-H solver using a source-time-dominant algorithm.

### 3.3 An Algorithm for Reducing Memory Requirements

PSU-WOPWOP is often compiled in a 32-bit implementation as this has been convenient on both Windows and Linux operating systems. It was in this setting that large permeable acoustic data surface cases began to fail in PSU-WOPWOP. The reason for this was that the entire source time data history for each point on the ADS was stored in memory for the entire computation. Hence, large cases could easily exceed the 2 gigabyte (GB) addressing limit of the 32-bit implementation. As most modern computers have 64-bit hardware and 64-bit operating systems, the easiest fix to this problem was to just recompile with a 64-bit com-
pilier and run the problem on a 64-bit operating system. Although this is very quick and easy to do, it does not address the real design problems that lead to a wasteful use of computer memory. In principle, just one time step of data is needed for the noise computation at any time in the source-time-dominant algorithm and the first goal of this work was to reduce the memory use to near the minimum required. The second goal of the new algorithm was to enable real-time computation of noise (for helicopter rotors) with only small modifications. Working with only one time step of data at a time is compatible with a real-time noise prediction because in that case the loading data is streamed to the acoustic code as it is computed in real-time and the acoustic code needs to use it immediately to be able to predict the noise in real time.

3.3.1 Design

There are two options available to reduce the memory requirement when using the source-time-dominant algorithm: 1) work with only one source time step in memory at a time; and 2) only work with part of the data on the ADS. A simultaneous increase of the number of source time steps and the size of the ADS results in a quadratic scaling of the memory requirements. Elimination of the growth of either of these variables reduces the scaling to linear, as shown in Fig. 3.6. Both options would require some reorganization of the computational algorithm in PSU-WOPWOP, and both methods could be used at the same time if necessary. But as previously mentioned, focusing on a reduction in the number of source time steps is compatible with real-time noise prediction. Moreover, if only one, or even just a few, source time steps of data are used in memory at any instance in time, then the number of points on the acoustic data surface would have to be very large indeed for any memory problems to be significant. So the first option, using only one source time step at a time in memory, was chosen.

3.3.1.1 Difference Equations in Discrete Time

Although in principle only one time step of the data is needed at a time, in practice more are needed because numerical finite differencing is used to compute the source time derivatives of the blade motion and to find velocity and acceler-
Figure 3.6: Impact of source time steps and source nodes on memory requirements.

...ation, along with the source time derivatives of the fluid properties. These are needed because Formulation 1A requires these terms as an alternative to performing a numerical time derivative of the integrals themselves a process that can be quite cumbersome and expensive, especially for moving observers. In the PSU-WOPWOP implementation, the source time derivatives are desired to be second order accurate throughout, i.e.

\[
v_j = \left( \frac{\partial y_j}{\partial \tau} \right) = \frac{y_j^{i+1} - y_j^{i-1}}{2\Delta \tau} + O((\Delta \tau)^2)
\]

\[
\dot{v}_j = \left( \frac{\partial^2 y_j}{\partial \tau^2} \right) = \frac{y_j^{i+1} - 2y_j^i + y_j^{i-1}}{\Delta \tau^2} + O((\Delta \tau)^2)
\]

where the subscript \( j \) refers to the spatial point and the superscripts \( (i-1, i, i+1) \) refer to the source time step. Central difference approximations are used throughout, with the exception of the beginning and ending times, which use second order accurate forward and backward differences, respectively.

### 3.3.1.2 Analysis of Terms in Formulation 1A

Velocity is a time derivative of the surface position and Formulation 1A needs both velocity terms and derivatives of the velocity \( v, \dot{v}, M \) and \( \dot{M} \). In addition, Formulation 1A requires the time derivatives of the surface loading vector \( l \) and the time derivative of the outward unit normal vector to surface \( \hat{n} \). PSU-
WOPWOP also predicts the gradient of the acoustic pressure, using formulation G1A [13], which requires source time derivatives of one order higher. While it is possible that these derivatives could be computed analytically, in practice they are routinely computed numerically. Thus the total number of source time steps required in memory depends on the formulation needed and the accuracy of the numerical approximation of the derivatives.

The minimum memory requirements of this algorithm are limited by the needs of these difference equations. To ensure second order accuracy at least three source time points are needed for the normal computation of Formulation 1A and at least four source time points are needed for the computation of the acoustic gradient (Formulation G1A).

### 3.4 Implementation of a New Computational Algorithm

Implementation of the new algorithm required making a few key changes to the computational algorithm used by PSU-WOPWOP. The first step was to change the procedure for managing data. There are three main tasks performed by the code: reading in data, computing solutions to Formulation 1A and outputing the results. Previous versions of the code performed these tasks sequentially; all of the data was read into memory, then solutions were computed, and then the results were output. To reduce memory requirements, changes were made to the read and computation steps. As a result, the new algorithm stores just four time steps of data in memory which greatly reduces the memory footprint of the code.

The procedure begins by reading in data for the first four source time steps, the minimum number required to compute the acoustic pressure gradient. Solutions are then computed for the first two time steps. At this point, a loop is initiated which begins by discarding the data from the last source time, given by the temporal index \((i-2)\) relative to the current time step \((i)\). Data from the next source time \((i+1)\) is then read and terms of the integrand are calculated for the current source time \((i)\). Once the integrand terms have been calculated, a
solution from Formulation 1A can be evaluated for the current source time step. Next, the solution can be interpolated and summed at its respective observer time. This whole procedure then repeats itself until there is no more input data, at which time solutions for the last two timesteps are computed and the program ends. Fig. 3.7 helps to illustrate the structure of the algorithm. Greater detail regarding the algorithm’s implementation may be found in Appendix ??.

It is important to note that the new algorithm differs subtly from the source-time-dominant algorithm described in Fig. 3.1. Whereas the routine in Fig. 3.1 places the ‘interpolation and sum’ step at the very end, the new algorithm performs this step within the loop (on the fly), much like the retarded-time algorithm in Fig. 3.2. This change is discussed in greater detail in Section 3.4.2.

### 3.4.1 Consideration of Data Types

Another consideration in the implementation of this algorithm is that there can be several different types of input data. PSU-WOPWOP allows for input data to be of a constant, periodic, or aperiodic nature in time. Constant data provides the simplest case in which the surface geometry, the loading, or both do not change with time. Periodic data is where either the changes in surface geometry or loading are time dependent, repeating with a known period. In such a case, only one period of data is needed as input into the noise prediction. Finally, aperiodic data is time dependent, but not assumed to be periodic. Neither constant nor periodic surface geometry and loading data present any real memory problems as an aperiodic case might. Additionally, dealing with the periodic case in the new algorithm that keeps only four time steps in memory at any given time is slightly more complex than for the aperiodic case. Therefore, in PSU-WOPWOP, the new algorithm is only implemented for aperiodic cases.

### 3.4.2 Observer Time Interpolation

Changing the order of procedures affects more than memory requirements. Using the source-time-dominant algorithm, the acoustic pressure is calculated for every point on the ADS along with the arrival time when it reaches the observer (observer time). Each source point is a different distance from the observer thus
the arrival times of acoustic signal from each source point will also be different. Moreover, not only does the spacing of arrival times for a single source point \((dt_{\text{local}})\) tend not to be uniform but the arrival times rarely coincide with times in the so-called global observer time array whose time steps are predetermined and (typically) uniformly spaced (see Fig. 3.8). Chaotic as it may seem, the resulting arrays of acoustic pressure at various arrival times can be organized by interpolation of the acoustic pressure time history from each source point onto the global observer time array. Even so, this is one situation in which the retarded-time algorithm has the advantage. By choosing the observer time before calculating the resulting source time all the pressure terms can be summed directly since, by design, they will reach the observer at the same time.
3.4.2.1 A New Interpolation Method in PSU-WOPWOP

In the original implementation of PSU-WOPWOP, the acoustic pressure for all source times from a single source point was stored as an acoustic pressure time history for that point as shown in Fig. 3.9. This design potentially requires large amounts of memory, and it is impractical for real-time noise prediction because interpolation of the pressure histories only occurs at the end of the run.

A different approach, one which requires significantly less memory and is well suited for real-time noise prediction, is to instead perform this time interpolation on the fly after each source time step (Fig. 3.10). Others have implemented similar ideas, which they call binning the data ([14], [15]), but in PSU-WOPWOP a procedure was developed to perform true interpolation with only limited data. In this way, the new interpolation routine works in a manner similar, and more complementary, to the numerical algorithm discussed earlier. Furthermore, it allows for reduced computation time. Interpolation of the pressure arrays on the fly is precisely what is needed for real-time noise prediction. Once it has been confirmed that all of the arrival times from signals emitted at the current source time have exceeded some observer time of interest, and therefore can no longer contribute to the signal at that time, the acoustic signal for that time can be output to the user immediately (in real time). This new procedure requires a restructuring of the order in which the acoustic pressure data
is interpolated but because the wave equation is linear the order in which the interpolation is performed has no impact on the final answer.

### 3.4.3 Parallel Computing

While the computation of noise at a single location is typically very fast, even for big cases where memory can become a problem, it is often the case that more information is needed. In particular, the directivity of the acoustic field can require hundreds or even thousands of single observer computations. For example, noise prediction for landing gear often requires that the acoustic levels and directivity be calculated on multiple measurement planes. Thus it can be necessary to calculate the noise for 10,000 observers or more. Parallel processing can be used to calculate the noise for multiple observers at once, shortening the code’s run time. This sort of parallelism scales linearly, thus the computation time is inversely proportional to the number of processors available during the computation. Large memory computations can also be problematic for parallel computations because typical computer clusters are a combination of multi-core processor servers networked together into a single cluster. The memory on a single node must be shared among the processor cores for that node. If too much memory is required for a single observer noise calculation, problems quickly develop when several processors on the same node also have large memory requirements. Therefore, the algorithm described previously, which reduces memory requirements, offers important benefits for both serial and parallel processing.
3.5 Results

The efficiency of the new memory algorithm was tested in a series of time trials, comparing the computation times of the old and new versions of PSU-WOPWOP over a series of cases using periodic and aperiodic data with 1, 5 and 10 observers for 4 different sets of observer time steps. A description of the time trials setup are given in Section 3.5.1. Results are presented in Section 3.5.2

3.5.1 Setup

Time trials for the new memory algorithm have been run for a series of cases using periodic and aperiodic data, but it was mentioned in Section 3.4.1 that the new memory algorithm is not applied to constant and periodic data. Why, then, should cases with periodic data be of interest? Although the new memory algorithm does indeed have no effect on the run time of cases with constant and periodic data, recall that new observer time interpolation scheme developed to complement the memory algorithm is used regardless of data type. The results of time trials for cases with periodic data will only show the impact of the new interpolation scheme while the results of time trials for cases with aperiodic data will show the impact of both the interpolation scheme and the memory algorithm. Once the impact of the interpolation scheme is established from the results of the periodic data cases, its impact in the aperiodic data cases can be isolated leaving only the effect of the new memory management algorithm.

Time trials were run using 1, 5 and 10 observers for a number of reasons:

With respect to the time trials with periodic data,

- Single and multiple-observer cases were run to ensure that the new interpolation scheme has a consistent impact on run time.

With respect to the time trials with aperiodic data

- A single-observer case was run to distinguish the impact of the interpolation scheme from the memory algorithm.
- Multiple-observer cases were run to assess the memory algorithm’s impact on an increasing number of observers.
3.5.2 Results

3.5.2.1 Run Time Efficiency

Slight improvements to the run time were observed for cases using constant and periodic data (Fig. 3.11); these are due to improvements made to the observer time interpolation scheme. In the case of aperiodic data, the run time increased in a case dependent manner (Fig. 3.12).

In the new version of PSU-WOPWOP, cases using aperiodic data require dramatically less memory as shown previously. This was accomplished by implementing a new algorithm that stores case information at only a few source time steps at a time. However, in reducing memory requirements, both the convenience and benefit of storing all of the source time data in memory was lost. In terms of convenience, this new algorithm requires that data be read from the data file one source time step at a time, rather than being read all at one time as was the previous design. The new algorithm can lead to increased run times for two reasons: (1) numerous calls to read smaller amounts of data from the disk are less efficient than reading large chunks of data at one time. This effect can be seen by comparing the results of Fig. 3.11, which all show a run time reduction of \(\sim 7\%\), and the single observer case in Fig. 3.12, which shows only a 0.7\% reduction in run time. With respect to run time, the improvements made to the observer time interpolation scheme were effectively negated by the decreased data read efficiency. (2) data must now be reread for every observer since it is only stored for four time steps before being discarded. The multiple observer cases in Fig. 3.12 clearly illustrate that what was saved in memory was lost to computation time when solving for multiple observers with serial processing.

But the results of Fig. 3.12 are not universal. Currently, PSU-WOPWOP computes the entire acoustic pressure time history for each observer sequentially (see Figs. 6.1 and 6.2). This design is advantageous in parallel computing because the solution for each observer can be calculated by a different slave processor simultaneously and no post-processing of the data is needed once it returns to the master. What’s more, the increased run times seen in Fig. 3.12 are eliminated when the new algorithm is used by multiple processors in parallel. This result is encouraging, particularly considering the aggressive move
Figure 3.11: Comparison of computation time for periodic data.

Figure 3.12: Comparison of computation time for aperiodic data.
towards multi-core processing seen from the computer industry in the past decade.

3.5.2.2 Memory Efficiency

Preliminary analysis showed very measurable changes to the memory requirements of cases with either constant or periodic data as result of the improvements to the logic used in the new interpolation scheme. The reduction in memory requirements is proportional to the number of observer time steps in the case since the new time interpolation routine only stores the local pressure arrays for 2 time steps rather than for the full length of the case as was the previous design (see Fig. 3.10). As an example, a case with an acoustic data surface with 10,000 nodes described over 10,000 source time steps would need about 400 MB of allocated memory (4 bytes/node x 10,000 nodes/time step x 10,000 time steps) to store the full pressure time history as is necessary in the old code. The new method would only require 80 KB of allocated memory (4 bytes/node x 10,000 nodes/time step x 2 time steps), a 5000-fold reduction, to store the necessary pressure information.

Analysis showed that the new memory algorithm is effective at reducing the memory footprint of aperiodic data dramatically. Recently Lockard [16] attempted to run a large permeable surface case using the unmodified version of PSU-WOPWOP. The 32-bit implementation of PSU-WOPWOP crashed after attempting to allocate more than the 2 GB limit. Running this case with the improved memory algorithm yielded memory requirements of less than 20 MB! More importantly, the case was able to run to completion in a 32-bit compilation of PSU-WOPWOP.

The physical memory requirements for aperiodic data cases have been reduced to what is needed to describe the model at just four time steps. This means that the length of an aperiodic data case (i.e., number of observer time steps) no longer has any impact upon the memory requirements, only upon the run time. By eliminating the length of a case as a factor in the memory requirements the spatial scale of a case must become extremely large before it will present a similar problem. This is an important and necessary result for the purposes of real-time noise prediction. Every program, including one for real-time
noise prediction, has a limited amount of memory to work with. The fact that the length of a case no longer impacts memory requirements means that, so long as the program has enough memory to store the necessary 4 time steps worth of data, it will never crash as a result of a case running too long.

3.6 Summary

In the last 20 years considerable progress has been made in the field of computational aeroacoustics and the scale of aeroacoustic fields predicted by methods based upon the Lighthill acoustic analogy and the FW-H equation has expanded by orders of magnitude. While the basic strategies for solving these equations have been resolved, the ever-increasing demands of the cases to which they are applied require a fresh look at the algorithms used to generate solutions. The new algorithm offers a solution to such computational demands and the results of its implementation provide insight into the direction of future work to improve and expand the capabilities of aeroacoustic prediction codes. The changes necessary to ease computational demands are very complementary to the intriguing goal of real-time noise prediction. Little, if any, modification of the new algorithm is necessary for use in a code that can predict noise in real time. Furthermore, it is the simple matter of using the algorithm with parallel processing to enable real-time noise prediction for multiple observers.
Broadband Noise Prediction

The following defines the problem of semi-empirical broadband noise prediction. The major topics discussed are:

1. Problem Description: Setup of the broadband noise prediction problem is detailed. A discussion of the new capabilities necessary in PSU-WOPWOP for FAA noise certification is given.

2. Analysis and Implementation: Analysis of the prediction methods is presented. Details of their implementation along with a discussion of each method’s limitations with respect to implementation in PSU-WOPWOP are given.

3. Validation of Prediction Methods: Both prediction methods are applied to a series to test cases for the purpose of validation.

4.1 Problem Description

The prediction of broadband noise is of particular importance in FAA civil noise certification. Broadband noise, distinct from discrete noise, is a form of loading noise that originates from several stochastic sources of unsteady loading. Broadband noise is characterized by noise contributions over a wide range of frequencies in the acoustic spectrum. The stochastic or random nature of broadband noise makes prediction by first-principles methods difficult. For this reason,
two semi-empirical methods have been built into an FAA civil noise certification package, newly incorporated into PSU-WOPWOP, which allows for rapid analysis of broadband rotor noise. The two methods implemented have significantly different levels of fidelity which provides an inherent hierarchy for their application. Their implementation provides the groundwork for the future implementation of a broadband noise prediction method built on first-principles.

4.2 Semi-Empirical Broadband Noise Prediction Methods

The two semi-empirical methods implemented in this task vary significantly in their levels of detail. In this manner the two methods form an explicit hierarchy such that they can be used as the first and second levels of analysis before a more computationally-intensive CFD method is developed. The first method is one described by Pegg [2] in 1979, based upon methods by Lowson [17], Hubbard [18], Schlegel [19] and Munch [20]. The second method was proposed by Brooks, Pope and Marcolini [3] in 1989. Considerable effort to classify, understand and predict broadband noise was made in the 10 years between the two publications and it is evident in the markedly more extensive procedure required by the method from Brooks, Pope and Marcolini (BPM).

The analyses presented in Sections 4.2.1.1 and 4.2.2.1 are not original, but rather a summary of the original sources [2] and [3].

4.2.1 The Pegg Method

Pegg’s method is based upon the assumption that broadband noise from a surface in a moving flow can be described by at least three mechanisms which generate random, fluctuating forces. These mechanisms include the surface pressure field which arises from a turbulent boundary layer (Boundary Layer Noise), the force fluctuations which can arise if the surface is moving in an initially-turbulent flow (Inflow Turbulence Noise) and a third, fluctuating force that results from shed vorticity (Vortex Noise). This approach avoids the treatment of noise sources as compact, instead assuming that the plate (rotor blade)
dimension is of the order of the acoustic wavelength. A brief discussion and analysis of these mechanisms is presented in the following section. Further details can be found in [2].

4.2.1.1 Derivation of Pegg’s Method

The noise mechanisms considered by Pegg are all uniquely characterized by the manner in which they arise and their relative contribution to the total noise (see Figure 4.1). Yet approximations of their respective contributions may be obtained from the same initial expression for acoustic power simply by making the proper assumptions. For spherical sound propagation, the relation between acoustic power \( W \) and acoustic pressure \( p' \) is given by

\[
(p')^2 = \frac{\rho c}{4\pi r^2} W
\]  

Figure 4.1: Variation of broadband noise with Mach number and spanwise location on rotor blade.

Therefore, if an expression for the acoustic power of a noise mechanism can be derived, a similar expression for the acoustic pressure can also be obtained.

In the subsonic speed regime, acoustic wavelengths are larger than the turbulent length scale such that sound from a rotor blade with dimensions up to
several times the turbulent scale may be calculated from the fluctuating pressures on the blade. Work to derive expressions for each noise mechanism begins by describing the acoustic power as

\[ W = \frac{1}{12\pi^2 c^3} \int_A \int_{A'} \frac{\partial p(x)}{\partial t} \frac{\partial p(x')}{\partial t} dA' dA \]  

(4.2)

where \( A' \) is the blade surface and \( \frac{\partial p(x')}{\partial t} \) is the rate-of-change of pressure on the surface, both described in retarded time. Equation 4.2 simply describes the total acoustic power as a function of the rate-of-change of pressure on the surface in retarded time. But the presence of a double integral is never comforting. Evaluation of the inner integral of Eqn. 4.2 can be avoided altogether if it is treated as a correlation integral:

\[ \int_{A'} \frac{\partial p(x)}{\partial t} \frac{\partial p(x')}{\partial t} dA' = \frac{\partial p^2}{\partial t} A_c \]  

(4.3)

where \( A_c \) is the correlation area of the pressure. Substitution of Eqn. 4.3 into Eqn. 4.2 yields,

\[ W = \frac{1}{12\pi^2 c^3} \int_{A_b} \frac{\partial p^2}{\partial t} A_c dA_b \]  

(4.4)

where \( A_b \) is a multiple of the blade area \( A \) to account for the blade traveling through homogeneous turbulence. Starting from Eqn. 4.4, Pegg derives expressions for the three noise mechanisms mentioned previously by making different assumptions about the manner in which they are generated.

To begin, inflow turbulence noise may be generated by fluctuations in the surface pressure field which arise from incoming turbulence. Based upon this description, it makes sense to relate surface pressure to some mechanism of a fluctuating nature. Kemp and Sears [21] derived an expression, simplified by Lowson [22], which describes the mean lift per unit span \( L \) on an airfoil with chord \( c \) in a sinusoidal gust of frequency \( \omega \) and magnitude \( u \) as

\[ L = \frac{\pi \rho c U u}{(\pi \omega c U)^{1/2}} \]  

(4.5)

This expression can be extended to three dimensions by noting that the mean
lift in terms of pressure fluctuations is

\[ L = p_{\text{rms}} l_c^{1/2} \]  \hspace{1cm} (4.6)

where \( l_c \) is the correlation length in the chordwise direction. To account for the rate-of-change of pressure on the blade Pegg assumes that the pressure fluctuations can be described by

\[ \left( \frac{\partial p}{\partial t} \right)^2 = \omega^2 p_{\text{rms}}^2 \]  \hspace{1cm} (4.7)

where \( \omega \) is a typical frequency. Substitution of Eqns. 4.5 and 4.6 and 4.7 into Eqn. 4.4 gives

\[ W = \frac{1}{12\pi\rho c^3} A_c (2A) \frac{\omega^2 \pi \rho c U^3 u^2}{l_c} \]  \hspace{1cm} (4.8)

where \( l \) is a typical length. Note that \( 2A \) arises because both sides of a blade radiate. Finally, by assuming \( A_c = l^2 \), \( l_c = l \) and \( A = A_b \), radiated acoustic power from an airfoil due to incident turbulence may be expressed as

\[ W = \frac{1}{6\rho c^3 M^6} \left( \frac{u}{U} \right)^2 A_b \]  \hspace{1cm} (4.9)

Application of Eqn. 4.1 yields the corresponding expression for acoustic pressure

\[ (p')^2 = \frac{\rho^2 c^4}{24\pi r^2} M^6 \left( \frac{u}{U} \right)^2 A_b \]  \hspace{1cm} (4.10)

Beginning with Eqn. 4.4, an expression can also be derived for the acoustic power radiated from boundary layer pressure fluctuations acting on a blade (Boundary Layer Noise). Pegg attempted to relate the correlation area \( A_c \) to the freestream velocity \( U \) using the expression

\[ \omega^2 A_c = K_1 U^2 \]  \hspace{1cm} (4.11)

where \( K_1 \) is a experimentally-derived constant in the range \( 0.1 < K_1 < 1.0 \). Based upon experimental results, Pegg argues that \( p_{\text{rms}}^2 \) is on the order of \( 36 \times 10^{-6} q^2 \), where \( q \) is dynamic pressure \( (\frac{1}{2} \rho u^2) \). Substituting this relationship and Eqn. 4.11 into Eqn. 4.4 gives an expression for the acoustic power generated by
boundary layer noise

\[ W = \frac{9K_1}{12} \rho c^3 M^6 A \times 10^{-6} \quad (4.12) \]

or in acoustic pressure

\[ (p')^2 = \frac{9K_1}{48\pi} \rho^2 c^4 M^6 A \frac{1}{r^2} \times 10^{-6} \quad (4.13) \]

Vortex noise, the third mechanism considered by Pegg, can be categorized into two components. The first component generates noise as a result of vortex shedding at the trailing edge of the blade which creates lift fluctuations. The second component is produced when tip vortices interact with a blade tip. Beginning with Eqn. 4.4, Pegg describes the fluctuating pressure difference as a local lift fluctuation per unit area. By this reasoning \( p \) can be expressed in terms of the local lift coefficient \( c_L \)

\[ p' = \frac{1}{2} \rho U^2 c_L \quad (4.14) \]

where \( U \) is the local mean velocity parallel to the surface. Thus,

\[ \frac{\partial p}{\partial t} = \frac{1}{2} \rho U^2 \frac{\partial c_L}{\partial t} \quad (4.15) \]

and

\[ W = \frac{\rho}{48\pi c^3} \int \frac{U^2}{r} \frac{\partial c_L^2}{\partial t} A_c dA \quad (4.16) \]

As with the previous noise sources, it would be convenient to describe the integral in Eqn. 4.16 in terms of the flow parameters. But the mechanisms which produce lift fluctuations in the absence of external excitation were not very well understood in 1979.

It is reasonable to expect the frequency of the lift fluctuations to be about the same as the characteristic frequency of vortex shedding at the trailing edge:

\[ \omega = 2\pi f = \frac{4\pi U}{c} \quad (4.17) \]

where \( c \) is the chord. Then, eqn. 4.16 can be rewritten as

\[ W = \frac{\rho}{48\pi c^3} \int_A U^4 (\omega c_L)^2 A_c dA \quad (4.18) \]
At this point, Pegg makes two important (but not always correct) assumptions. Based upon the idea that lift fluctuations can be related to time fluctuations in the boundary layer thickness at the trailing edge, Pegg argues that the root mean square of the fluctuating lift coefficient should be on the numerical order of the $-1/5$th power of the Reynolds number $Re$ (thus $p_{\text{rms}}^2 \sim Re^{-0.4}$). He also assumes that the correlation area $A_c$ should be governed by the size of the larger eddies at the trailing edge. For a rotating blade with a 10% thickness ratio, that would be on the order of half the section thickness, or $0.05c^2$. From these two assumptions, Eqn. 4.18 can be integrated to yield

$$W = \frac{\pi \rho U^6 Re^{-0.4} A}{120c^3} = \frac{\pi \rho c^3 M^6 Re^{-0.4} A}{120c^3}$$

Assuming a dipole distribution, a typical acoustic pressure level is

$$(p')^2 = \frac{1}{480} \rho^2 c^4 A \frac{M^6 Re^{-0.4}}{r^2}$$

Lastly, an expression for the acoustic power generated from tip radiation is needed. At this point, Pegg begins from Eqn 4.9 and assumes that the correlation lengths are on the order of 5% of the chord and a typical frequency is

$$\omega = 40 \frac{U}{c}$$

Recall that $(p')^2$ is on the order of $36 \times 10^{-6} q^2$. From this relationship, the expression for frequency in Eqn. 4.21 and assuming a correlation area approximately 5% of the blade area, the acoustic power produced by tip radiation can be expressed as follows

$$W = \frac{3}{\pi} \frac{\rho}{c^3} U^6 A_T \times 10^{-6} = \frac{3}{\pi} \rho c^3 M^6 A_T \times 10^{-6}$$

where $A_T$ is the blade tip area over which the vortex interaction occurs (assumed to be 5% of the blade area). Finally, assuming a dipole distribution, the
acoustic pressure can be expressed as

\[(p')^2 = \frac{3}{4\pi^2} \frac{\rho^2 c^4}{r^2} M^6 A_T \times 10^{-6}\] (4.23)

Contributions from the four broadband noise sources are compared in the plots of Fig. 4.1 which show the variation of these noise sources with respect to rotor tip Mach number and rotor radius. Both plots show turbulent inflow noise as dominant at every Mach number and spanwise blade position. The shed vortex noise source also provides a measurable contribution to the noise. The noise contribution from shed vortices remains lower than that from turbulent inflow at every spanwise position, but Fig. 4.1b clearly shows that the shed vortices produce significantly more noise at the tip than turbulent inflow does further inboard. On the other hand, the broadband noise contributions from the boundary layer and tip-vortex interactions were found to be insignificant from this analysis.

4.2.1.2 Input Data

Pegg’s method predicts broadband noise from a rotor by considering just six terms: total blade area of the rotor \(A_b\), rotor thrust \(T\), rotor tip speed \(V_T\) (in hover), average blade lift coefficient \(\bar{C}_L\), the angle between the negative thrust axis and the vector pointing from the hub to the observer \(\theta_1\) and the distance from the source to the observer \(r\).

A namelist was created for use in the PSU-WOPWOP namelist file which accepts any or all of these terms as fixed inputs but the default method of obtaining their values is by computing them within PSU-WOPWOP. Although user input provides the ability to adjust predictions based upon experience, computation of the input terms by PSU-WOPWOP ensures consistency in the computation of terms and avoids potential errors. The total blade area refers to planform area such that \(A_b = N_b c R\) for rectangular blades, where \(N_b\) is the number of blades, \(c\) and \(R\) are the blade chord and radius, respectively. When this value is not explicitly provided in the namelist this value is approximated as

\[A_b \approx \frac{1}{2} \sum_i^n A_i\] (4.24)
where $A_i$ is the cell area of which there are $n$ cells making up a blade. For blades with relatively low camber, which accounts for most propeller and rotor blades, this approximation should be sufficiently accurate.

The rotor thrust $T$ is taken as the component of total thrust $T_{Total}$ along the hub axis such that $T = T_{Total} \cdot \mathbf{r}_{Hub}$. If given in the namelist, this value is assumed to have already been applied along the hub axis. The rotor tip speed $V_{tip}$ is taken to be the magnitude of tip velocity in hover, $\Omega R$, where $\Omega$ is the rotational velocity given rad/sec and $R$ is the blade span given in units of length.

The definition of $\bar{C}_L$ comes from the expression for thrust coefficient where

$$C_T = \frac{1}{2} \int_0^1 \sigma r^2 C_l dr$$

such that the entire blade is assumed to be operating at $C_l = \bar{C}_L$. Thus,

$$C_T = \frac{1}{2} \int_0^1 \sigma r^2 C_L dr = \frac{1}{2} \bar{C}_L \int_0^1 \sigma r^2 dr = \frac{1}{6} \sigma \bar{C}_L$$

or simply

$$\bar{C}_L = 6 \left( \frac{C_T}{\sigma} \right)$$

where

$$C_T = \frac{T}{\rho A b V_{tip}^2} = \frac{T}{\rho (\pi R^2) V_{tip}^2}$$

and rotor solidity $\sigma$ is the ratio of the total blade area to the rotor disk area:

$$\sigma = \frac{A_b}{\pi R^2}$$

Insertion of Eqns. 4.28 and 4.29 into 4.27 yields

$$\bar{C}_L = \frac{6T}{\rho A_b V_{tip}^2}$$

The expression in Eqn. 4.30 is convenient for the present application because it uses terms already needed by the method, namely $T$, $A_b$ and $V_{tip}$.

Pegg uses the angle between the negative hub axis and the vector pointing from the hub axis to the observer as a directivity function to weight the noise
level arriving at the observer. This angle, $\theta_1$, can be found using the definition of the dot product such that

$$\theta_1 = \cos^{-1}\left( \frac{\mathbf{H} \cdot \mathbf{r}_{HO}}{|\mathbf{H}| |\mathbf{r}_{HO}|} \right)$$

(4.31)

where $\mathbf{H}$ is the negative hub axis vector and $\mathbf{r}_{HO}$ is the vector between the hub and the observer. Care must be taken to use $\mathbf{r}_{HO}$ and not $\mathbf{r}_{OH}$ as the answers will differ by $(180^\circ - \theta)$ degrees as shown in Fig. 4.2. Furthermore, the same frame of reference must be used for both vectors. In PSU-WOPWOP, $\mathbf{r}_{HO}$ is determined in the global frame, while $\mathbf{H}$ is specified in the hub-fixed frame. The choice of which reference frame to use does not matter so long as they are same, but routines which transform the local frame into the global frame already exist in PSU-WOPWOP, so the calculation of $\theta_1$ was performed in the global frame.

4.2.1.3 Implementation

The terms used in this method are a function of the rotor blades, though the rotor blades are not considered individually. Thus there is an implicit assumption that the blades are all operating under similar conditions. This assumption is perfectly reasonable for hover but may not be accurate when predicting transient maneuver cases.

The first step to Pegg’s broadband noise prediction method involves the computation of the peak broadband noise frequency using the following equa-
\( f_p = -240 \log_{10}(T) + 2.448V_{tip} + 942 \)  \hspace{1cm} (4.32)

It should be noted that this equation is dimensionally incorrect and thus unit dependent. Eqn. 4.32 should be applied when using metric units, while Eqn. 4.33 should be applied for English units of lb \( f \) for \( T \) and ft/s for \( V_{tip} \).

\( f_p = -240 \log_{10}(T) + 0.746V_{tip} + 786 \)  \hspace{1cm} (4.33)

Next, the 1/3-octave band containing this peak frequency must be identified. The upper and lower limits of a 1/3-octave band are defined by

\[ f_u = f_c * 2^{1/6} \]  \hspace{1cm} (4.34)

\[ f_l = f_c / 2^{1/6} \]  \hspace{1cm} (4.35)

As an example, a peak frequency of \( f_p = 510 \) Hz would fall in the 1/3-octave band with a center frequency of 500 since

\[ 500/2^{1/6} < 510 < 500 * 2^{1/6} \]

The last step is to compute the peak 1/3-octave band acoustic pressure level \( \text{SPL}_{1/3} \) for each 1/3-octave band:

\[ \text{SPL}_{1/3} = 20 \log_{10} \left( \frac{V_{tip}}{c} \right)^3 + 10 \log_{10} \left[ \frac{A_b}{r^2} (\cos^2 \theta_1 + 0.1) \right] + S_{1/3} + f(C_L) + 130 \]  \hspace{1cm} (4.36)

where \( S_{1/3} \) is obtained from Fig. 4.3 such that the peak band center frequency, with \( S_{1/3} = -7.5 \), refers to the octave band containing the peak frequency computed in Eqn. 4.32. Note that \( f(C_L) \) is defined as

\[ f(C_L) = \begin{cases} 
10 \log_{10}(C_L/0.4) & C_L \leq 0.48 \\
0.8 + 80.0 \log_{10}(C_L/0.48) & C_L > 0.48 
\end{cases} \]  \hspace{1cm} (4.37)

In rotor acoustics, a typical range of frequencies (which Pegg assumes) occupies 9 octave bands from 50 Hz to 10 kHz. Pegg’s method applies the spectral
weighting function found in Fig. 4.3 once per octave, corresponding to the 1/3-octave bands which satisfy $f_c = f_{c,peak} \times 2^i$ where $i$ is an integer.

Figure 4.3: Normalized rotor broadband noise empirically determined spectrum shape. [19]

Pegg’s method is quite simple to implement, very computationally efficient and reasonably accurate for low to moderate advance ratios. Although it mod-
els broadband noise as the sum of contribution from three self-noise-generating mechanisms, Pegg’s method generates a prediction for broadband noise as a single decibel level per 1/3-octave band. The Pegg method does not provide any measure of how the various mechanisms contribute independently. And without this feature there is no easy way to analyze, and potentially reduce, the contributions from the different mechanisms.

4.2.2 Method of Brooks, Pope and Marcolini

The method of Brooks, Pope and Marcolini (BPM) was developed to predict self-generated noise of an airfoil encountering smooth airflow. BPM models five self-noise mechanisms due to boundary-layer phenomena (see Fig. 4.4): boundary-layer turbulence passing over the trailing edge, separated-boundary-layer and stalled-airfoil flow, vortex shedding from laminar-boundary-layer instabilities, vortex shedding from blunt trailing edges, and turbulent vortex flow that exists near the tips of lifting blades. This method is based upon noise results from experimental tests of seven NACA airfoil blade sections with chord lengths ranging from 2.5 to 61 cm which were performed at wind tunnel speeds up to Mach 0.21 and angles of attack (AOA) from 0° to 25.2°. Analysis of this method and its terms are presented in the following section with further detail available in [3].

4.2.2.1 Derivation of BPM’s Method

Prior efforts to predict turbulent-boundary-layer–trailing-edge (TBL-TE) noise have shown that accuracy to within 7 dB is possible if the prediction is based, in part, on the airfoil’s boundary layer thickness $\delta$, boundary layer displacement thickness $\delta^*$ and boundary layer momentum thickness $\theta$. The scaling of these terms was determined experimentally for airfoils with both tripped and untripped boundary layers, from zero angle-of-attack up to stall, on both the pressure and suction sides of the blade. Scaling laws were then developed for each of the five self-noise mechanisms by analyzing spectra of the wind tunnel tests described earlier.

For zero angle of attack and tripped boundary layers, the scaled acoustic
Figure 4.4: Self noise mechanisms. [3]
pressure level SPL is normalized by

\[
\text{Scaled SPL}_{1/3} = \text{SPL}_{1/3} - 10 \log_{10} \left( M^5 \frac{\delta^* L}{r_e^2} \right)
\]

(4.38)

where \( L \) is the blade span and \( r_e \) is the retarded observer distance. Eqn. 4.38 provides a baseline with which to formulate scaling laws based on variation in AOA and the extent of boundary layer separation. For each spectrum obtained in their experiments, BPM identified the approximate location of the Strouhal peak, \( St_{\text{peak}} = (f\delta^*/U)_{\text{peak}} \), where \( \delta^* \) is the boundary-layer displacement thickness, and proposed a shape function denoted by \( A \) (see Fig. 4.5a) as representative of the 1/3-octave spectral shape of the TBL-TE noise mechanism. Fig. 4.6 shows the identification for one of the experimental setups. This approach, based upon work of Fink [23], seeks to predict the SPL spectrum by assuming a universal spectral shape for the noise. A useful benefit of such an approach is that it can be applied to the pressure and suction sides of a blade independently simply by dropping the predicted noise level -3 dB. By application of this reduction and the spectral shape function, the scaled SPL may be given by

\[
\text{Scaled SPL}_i = \text{SPL}_i - 10 \log_{10} \left( M^5 \frac{\delta_i^* L}{r_e^2} \right)
\]

\[
= A \left( \frac{St_i}{St_1} \right) + (K_1 - 3)
\]

(4.39)

where \( i = p \) or \( s \) for the pressure or suction side of a blade, and \( K_1 \) is an empirically derived constant. \( St_1 \) is the Strouhal number defined for TBL-TE noise. This equation can be modified for angle-dependent noise SPL\(_{\alpha}\) to represent the separated-boundary-layer noise contribution

\[
\text{Scaled SPL}_{\alpha} = \text{SPL}_{\alpha} - 10 \log_{10} \left( M^5 \frac{\delta_2^* L}{r_e^2} \right)
\]

\[
= B \left( \frac{St_2}{St_2} \right) + K_2
\]

(4.40)

where \( B \) is the spectral shape function for separated flow noise (see Fig. 4.5b), \( St_2 \) is the Strouhal number for separated flow noise and \( K_2 \) is another empiri-
Bringing together Eqns. 4.39 and 4.40 yields

\[ \text{SPL}_{\text{TOT}} = 10 \log_{10} \left( 10^{\text{SPL}_{\alpha}/10} + 10^{\text{SPL}_{s}/10} + 10^{\text{SPL}_{p}/10} \right) \]  

(4.41)

where

\[ \text{SPL}_{\alpha} = 10 \log_{10} \left( \frac{\delta_{\alpha}^* M^5 L D_h}{r_c^2} \right) + B \left( \frac{St_s}{St_2} \right) + K_2 \]  

(4.42)

\[ \text{SPL}_{s} = 10 \log_{10} \left( \frac{\delta_{s}^* M^5 L D_h}{r_c^2} \right) + A \left( \frac{St_s}{St_1} \right) + (K_1 - 3) \]  

(4.43)

\[ \text{SPL}_{p} = 10 \log_{10} \left( \frac{\delta_{p}^* M^5 L D_h}{r_c^2} \right) + A' \left( \frac{St_p}{St_1} \right) + (K_1 - 3) + \Delta K_1 \]  

(4.44)

for angles of attack up to \((\alpha_{\ast})_0\), an angle at which a switch in the definition of \(K_2\) is used to model the jump from attached TBL flow to a stalled flow condition. At angles above \((\alpha_{\ast})_0\),

\[ \text{SPL}_{p} = -\infty \]

and

\[ \text{SPL}_{s} = -\infty \]

and

\[ \text{SPL}_{\alpha} = 10 \log_{10} \left( \frac{\delta_{\alpha}^* M^5 L D_h}{r_c^2} \right) + A' \left( \frac{St_s}{St_2} \right) + K_2 \]  

(4.45)

where \(A'\) is the curve \(A\) but for a value of \(R_c\) three times the actual value. \(D_h\) and \(D_l\) are directivity functions.

Due to the erratic behavior and complexity of the mechanism responsible for laminar-boundary-layer-vortex-shedding (LBL-VS) noise no scaling methods had been developed for it prior to the work by BPM. Still, a similar approach to the one described for TBL-TE noise was used. Scaled 1/3-octave acoustic pressure level spectra were collected for four airfoil sizes, each at four wind-tunnel speeds. At zero AOA with untripped boundary layers, normalization of the SPL with level scaling is given by

\[ \text{Scaled SPL}_{1/3} = \text{SPL}_{1/3} - 10 \log_{10} \left( M^5 \frac{\delta_p L}{r_c^2} \right) \]  

(4.46)
with

\[ St' = \frac{f_\delta p}{U} \]  \hspace{1cm} (4.47)

Figures 4.9 and 4.10 show Scaled SPL$_{1/3}$ levels versus Strouhal number for a wide range of airfoils and operating conditions. These plots show that LBL-VS noise is rather invariant with respect to spectral shape. Based on this observa-
Figure 4.6: Scaled 1/3-octave spectra of tripped BL airfoils at $\alpha_t = 0^\circ (\alpha^* = 0^\circ)$. Symbols indicate approximate spectral peak locations. $\alpha_t$ is the AOA referenced to the tunnel streamwise axis (deg), $\alpha^*$ is the effective AOA corrected for open wind tunnel effects (deg).

A function $G_1$ (see Fig. 4.11) was designed to represent the contribution from LBL-VS noise to the self-noise 1/3-octave spectra. Additionally, a function $G_2$ (Fig. 4.12) specifies the curve shape and $G_3$ (Fig. 4.13) applies angle dependence for the level of the $G_2$ curve by

$$G_3(\alpha^*) = 171.04 - 3.03\alpha^*$$  \hspace{1cm} (4.48)

Altogether, the LBL-VS noise spectrum presented in 1/3-octaves is predicted by

$$\text{SPL}_{\text{LBL-VS}} = 10 \log_{10} \left( \frac{\delta_p M^5 L D_h}{r_e^2} \right) + G_1 \left( \frac{St'}{St'_{\text{peak}}} \right) + G_2 \left[ \frac{R_c}{(R_c)_{0}} \right] + G_3(\alpha^*)$$ \hspace{1cm} (4.49)

The prediction method proposed for tip vortex formation noise was developed by Brooks and Marcolini [24]. Recognizing that 3D models produced both
tip noise and TBL-TE noise while 2D models produce only the latter, they isolated the tip noise, a high-frequency broadband self noise, by comparing aerodynamic and acoustic test results of 2D and 3D airfoil models. Tip noise is generally associated with the turbulence present in the locally separated flow region at the tip of a lifting blade, where tip vortices form. It is assumed that the mechanism of noise production is trailing edge noise due to passage of this turbulence over the edge into the wake. For a noise spectrum presented in a 1/3-octave form, tip vortex formation noise is predicted by

\[
\text{SPL}_{\text{TIP}} = 10 \log_{10} \left( \frac{M^2 M_{\text{max}}^3 l^2 D_h}{r_c^2} \right) - 30.5(\log_{10}(St'') + 0.3)^2 + 126
\]  

(4.50)

where the Strouhal number is defined as

\[
St'' = \frac{fl}{U_{\text{max}}}
\]  

(4.51)

in which \( l \) is the spanwise extent of the viscous core of a vortex at the trailing edge. Note that the second term in Eqn. 4.50 is a parabolic fit about a peak Strouhal number of 0.5. For the rounded tip used in their test, it was determined that the spanwise extent of the separation due to the tip vortex at the trailing edge is

\[
l/c \approx 0.008\alpha_{\text{TIP}}
\]  

(4.52)

With respect to Eqn. 4.52, \( \alpha_{\text{TIP}} \) was redefined as \( \alpha'_{\text{TIP}} \) (Eqn. 4.53) to generalize solutions for arbitrary aspect ratios, blade twist and variations in spanwise flow.

\[
\alpha'_{\text{TIP}} = \left[ \left( \frac{\partial L'/\partial y}{(\partial L'/\partial y)_{\text{ref}}}_{y \rightarrow \text{TIP}} \right) \right] \alpha_{\text{TIP}}
\]  

(4.53)

Further experimentation showed, however, that the definition for \( l \) in Eqn. 4.52 did not hold for airfoils with flat tips. For consistency, the following definition for \( l \) was proposed for flat tips:

\[
l/c = \begin{cases} 
0.023 + 0.0169\alpha'_{\text{TIP}} & (0^\circ \leq \alpha'_{\text{TIP}} \leq 2^\circ) \\
0.0378 + 0.0095\alpha'_{\text{TIP}} & (2^\circ < \alpha'_{\text{TIP}})
\end{cases}
\]  

(4.54)
One final piece of the puzzle remains: a method for predicting trailing-edge-bluntness–vortex shedding (TE-BVS) noise. Wind-tunnel tests were performed on a single 90.96-cm-chord NACA 0012 airfoil with various TE geometries via extensions with blunt or sharp trailing edges, see Fig. 4.14. The results from Fig. 4.16 demonstrate that TBL-TE noise is fairly invariant with regard to changes in edge geometry, so long as the TE is sharp and the boundary layers are effectively the same. Noise spectra for trailing-edge bluntness are presented in Fig. 4.15 for the edge geometries of Figs. 4.14(a) and 4.14(c). The data in Fig. 4.15 was obtained by subtracting the spectral content of the sharp TE spectra from the total spectra, which should represent only the bluntness contribution.

For this part of the analysis, the peak of the spectral humps was chosen as the reference for the level and frequency definition, given by

\[ S_{\text{peak}}' = \frac{f_{\text{peak}} h}{U} \quad (4.55) \]

To scale the signal amplitudes, the peak values of the 1/3-octave spectra of Fig. 4.15 were normalized as

\[ \text{Scaled peak SPL}_{1/3} = \text{Peak SPL}_{1/3} - 10 \log_{10} \left( \frac{M^{5.5} h L}{r^2_e} \right) \quad (4.56) \]

Although a 6th power dependence is typically assumed, BPM found that a Mach dependence to the 5.5 power gave better overall scaling results than either 5 or 6. Fig. 4.7 shows the scaled levels plotted versus the thickness ratio \( h/\delta^* \), where \( h \) is the TE thickness. Note that the scaled levels are uniformly higher for the plates than for the edge extensions with the same thickness ratio. As well, the levels increase with increasing thickness ratio. The curve-fits \( G_4(h/\delta^*, \Psi) \) in Fig. 4.7 are straight lines which level off at \( h/\delta^* = 5 \), an admittedly arbitrary point choice by BPM due to a lack of available noise data. Nevertheless, this choice works well in practice for rotor blades and wings. The function \( G_5(h/\delta^*, \Psi) \) (Fig. 4.8) defines the spectral curve fit with a peak level of 0 dB and a shape defined in terms of \( S_{\text{peak}}''' / S_{\text{peak}}'' \). \( G_5 \) may be obtained for intermediate values of \( \Psi \) by simply interpolating between the limiting cases shown in Figs. 4.8a and
4.8b:

\[
G_5 \left( \frac{h}{\delta_{avg}}, \Psi, \frac{St'''}{St_{peak}} \right) = (G_5)_{\Psi=0^\circ} + 0.0714\Psi \left[ (G_5)_{\Psi=14^\circ} - (G_5)_{\Psi=0^\circ} \right] \tag{4.57}
\]

where

\[
\delta_{avg} = \frac{\delta_p + \delta_s}{2} \tag{4.58}
\]

Figure 4.7: Peak scaled levels for bluntness noise versus thickness ratio \( h/\delta^* \) determined from Fig. 4.15.

Finally, in a 1/3-octave presentation, the TE bluntness noise spectrum is predicted by

\[
SPL_{BLUNT} = 10 \log_{10} \left( \frac{hM^{5.5}L D_h}{r_c^2} \right) + G_4 \left( \frac{h}{\delta_{avg}}, \Psi \right) + G_5 \left( \frac{h}{\delta_{avg}}, \Psi, \frac{St'''}{St_{peak}} \right) \tag{4.59}
\]

In which \( \Psi \) is defined as the angle, in degrees, between the upper and lower surfaces of the airfoil just upstream of the trailing edge. For example, \( \Psi = 0^\circ \) for the edge on a flat plate and \( \Psi = 14^\circ \) for an NACA 0012 airfoil.

As mentioned previously, this prediction method is significantly more involved than the one by Pegg. Its implementation, in particular, requires a thorough analysis of its details in order to understand how the equations to predict
Figure 4.8: Spectral shape functions for TE bluntness noise.

4.2.2.2 Input Data

The method by BPM considers not just each blade separately, but the sections that make up each blade. A blade may be modeled as having one or many
Figure 4.9: Scaled 1/3-octave spectra for untripped boundary-layer airfoils at \( \alpha_t = 0^\circ (\alpha_* = 0^\circ) \). Symbols indicate approximate LBL-VS reference peak locations sections depending on the spanwise variation in flow properties and the blade’s characteristics, specifically the chord \( c \), angle of attack \( \alpha \), the freestream velocity \( U \) and the vector between the source and observer \( r \).

Unlike Pegg’s method, the method by BPM calls for a number of variables that PSU-WOPWOP is not currently able to compute. The chord, tip lift-curve-slope \( LCS_{Tip} \), tip angle of attack \( \alpha_{Tip} \), blade section angle of attack \( \alpha \) and trailing-edge flow angle \( \Psi \) must be provided by the user, otherwise default values are used. The variables that can be computed are the blade section length \( L \) and freestream velocity. As before, these terms can be either be provided by the user or computed by PSU-WOPWOP.

BPM’s method uses compact loading, such that the total lift vector at each spanwise location has been integrated over the chord and thus has units of force/length. The blade section lengths can be taken as the spanwise distance between the locations at which the compact load vectors are applied. The freestream velocity is taken as the velocity seen by the source, i.e., in the source local
Figure 4.10: Scaled 1/3-octave spectra for untripped boundary-layer airfoils at various angles of attack. $U=71.3$ m/s. Symbols indicate approximate LBL-VS reference peak locations.

Figure 4.11: One-third-octave spectral shape function $G_1$ for LBL-VS noise.
Figure 4.12: Peak scaled levels for LBL-VS noise versus Reynolds number. Data symbols are values of $\alpha_*$ rounded off to the nearest degree.

Figure 4.13: Normalization of LBL-VS noise peak scaled levels by functions $G_2$. Data from Fig. 4.12
Figure 4.14: Illustration of trailing-edge extensions and plates. Smooth surface transition was provided for all geometries.

reference frame rather than the global reference frame. As in Pegg’s method, the radiation vector \( \mathbf{r} \) is readily available since PSU-WOPWOP uses this term in the retarded-time formulation.

The directivity of the noise is described by the spherical coordinate angles \( \phi \), which is the angle of the radiation vector from the blade-frame x-y plane, and \( \theta \), which is the angle of radiation vector from the blade-frame x-axis (see Fig. 4.17). The blade frame coordinates are defined as follows: \( \hat{y}^b = [0.0, 1.0, 0.0] \), the \( \hat{z}^b \) is taken as the normal to the lifting line (analogous to the surface normal for non-compact surfaces) and \( \hat{x}^b = \hat{y}^b \times \hat{z}^b \). Just as in Pegg’s method the blade frame coordinate system \( [x]^b \) must be converted into the global frame and \( \phi \) and \( \theta \) are calculated by

\[
\phi = \cos^{-1} \left( \frac{\hat{x}^g \times \mathbf{r} \cdot \hat{z}^g}{||\hat{x}^g \times \mathbf{r}|| \cdot ||\hat{z}^g||} \right) \tag{4.60a}
\]
Figure 4.15: TE-bluntness–vortex -shedding noise, data for untripped BL, and data with plate extensions (Fig. 4.14) attached.

\[ \theta = \cos^{-1}\left( \frac{\hat{x} \cdot \mathbf{r}}{|\hat{x}| |\mathbf{r}|} \right) \]  

(4.60b)

### 4.2.2.3 Implementation

While significantly more complex than Pegg’s method, BPM’s method generates noise predictions for each noise mechanisms separately. SPL does not grow
Figure 4.16: Spectral density for TE noise for 60.96-cm-chord airfoil with various degrees of TE bluntness. Tripped BL; $a_t = 0^\circ$; $\Theta_e = 90^\circ$. $\Theta_e$ is the angle from the source streamwise axis $x$ to the observer, see Fig. 4.17. $S(f)$ is the spectrum of self-noise with units $Pa^2/Hz$. The level is referenced to 1-Hz bandwidth. Data is from [25].

Linearly, thus $L_\Sigma \neq \sum_{i=1}^{n} L_i$, where $L_i$ is the SPL contribution from a single noise mechanism and $L_\Sigma$ is the total SPL, but sound intensity $I$ can be added together (Eqn. 4.61). Thus, the total noise level can be easily obtained as the log of the sum of sound intensity (Eqn. 4.62) and the contribution from each mechanism can still be analyzed with ease.

\[
\left( \frac{I_i}{I_{\text{ref}}} \right) = \left( \frac{p_i}{p_{\text{ref}}} \right)^2 = 10^{L_i/10}, \quad i = 1, 2, ..., n \tag{4.61}
\]

\[
L_\Sigma = 10 \log_{10} \left( \sum_{i=1}^{n} \frac{I_i}{I_{\text{ref}}} \right) dB \tag{4.62}
\]

Prediction of each mechanisms separately results in the growth, in terms of size and complexity, of the prediction code. Even so, independent prediction of each noise source makes this a much more powerful prediction method be-
cause the contribution from each noise source can be analyzed and potentially reduced.

The noise levels predicted in BPM’s prediction method are dependent on the Strouhal number which is defined in various forms depending on the specific noise mechanism. This dependence on frequency requires an additional step in the prediction process. A frequency correction factor must be applied to account for the perceived shift in frequency that is produced when the source and observer and are in motion, commonly known as the Doppler effect. Assuming the predicted noise levels are desired at the center 1/3-octave frequencies (as BPM do), the correction factor must be applied to this frequency array to determine the frequencies produced at the source such that they arrive to the observer at the center 1/3-octave frequencies. This correction is given by

\[
f_{\text{source}} = \left( \frac{1 - M_{\text{source}} \cdot \hat{r}}{1 - M_{\text{observer}} \cdot \hat{r}} \right) f_{\text{observer}}
\]

The source Mach number \(M_{\text{source}}\) must be evaluated at the source time \(\tau\) at which the signal is emitted and observer Mach number \(M_{\text{observer}}\) must be eval-
uated at the observer time $t$ at which that signal arrives. It follows that the radiation vector is defined by

$$r = x(t) - y(\tau).$$

(4.64)

4.2.3 Broadband Noise Prediction for FAA Civil Noise Certification

FAA civil noise certification is dependent on the effective perceived noise level (EPNL) of a vehicle’s flight in certain prescribed flight conditions: take-off, level flight (overflight), and descent. EPNL is computed from the tone corrected perceived noise levels (PNLT) which is computed using acoustic pressure data obtained at half-second intervals throughout the flight condition. Both discrete frequency noise and broadband noise contribute to the PNLT and EPNL computations at various stages in the flight profiles. More information regarding FAA civil noise certification and the calculation of EPNL can be found in Chapter 5.

Both of the semi-empirical methods implemented in this project make somewhat implicit assumptions about flight conditions, specifically that the flight condition is fairly steady. A better way to say this might be that these methods are not suited for predicting broadband noise from aggressive, transient maneuvers. Knowing this, it is practical to measure the broadband noise as an average of the predictions made over some small time range. For the purposes of FAA civil noise certification, the obvious choice for that time range is the rotor period. Similar to Eqn. 4.62, the average SPL over the rotor period can be found from an average of the sound intensity such that

$$L_\Sigma = 10 \log_{10} \left( \frac{\sum_{i=1}^{n} \frac{I_i}{T_{ref}}}{n} \right) dB$$

(4.65)

where $n$ is the number of source time predications computed in the rotor period.
Table 4.1: Flight conditions for test case 1 with Pegg’s method

<table>
<thead>
<tr>
<th>Tip Mach Number</th>
<th>0.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotor speed, rps</td>
<td>3.5</td>
</tr>
<tr>
<td>Blade radius, m</td>
<td>9.45</td>
</tr>
<tr>
<td>Observer Angle, $\theta$, deg</td>
<td>85°</td>
</tr>
<tr>
<td>Number of blades</td>
<td>5</td>
</tr>
<tr>
<td>Thrust, N</td>
<td>69420</td>
</tr>
<tr>
<td>Blade area, $m^2$</td>
<td>18.58</td>
</tr>
<tr>
<td>Air density, $kg/m^3$</td>
<td>1.228</td>
</tr>
<tr>
<td>Observer distance, m</td>
<td>61.6</td>
</tr>
<tr>
<td>Mean lift coefficient</td>
<td>0.438</td>
</tr>
</tbody>
</table>

4.3 Results

The two semi-empirical broadband noise methods are validated against the test cases described in their respective papers.

4.3.1 Method by Pegg

Pegg’s report included two test cases in which the details of the method were described. The implementation of this method in PSU-WOPWOP was validated by designing and running two test cases with identical input data and comparing the results to those documented in the report. The input variables for the two test cases are given in Tables 4.1 and 4.2.

4.3.1.1 Results

Figures 4.18a and 4.18b show excellent agreement between the results documented by Pegg and the results produced by the implementation of Pegg’s method in PSU-WOPWOP. Small differences between the results of each case are the result of round-off error in Pegg’s calculations.

The two figures show results at different 1/3-octaves since Pegg’s method uses the 1/3-octave containing the peak frequency to anchor the spectral curve.
Table 4.2: Flight conditions for test case 2 with Pegg’s method

<table>
<thead>
<tr>
<th>Tip Mach Number</th>
<th>0.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotor speed, rps</td>
<td>21.1</td>
</tr>
<tr>
<td>Blade radius, m</td>
<td>1.52</td>
</tr>
<tr>
<td>Observer Angle, $\theta$, deg</td>
<td>10°</td>
</tr>
<tr>
<td>Number of blades</td>
<td>5</td>
</tr>
<tr>
<td>Thrust, N</td>
<td>5206</td>
</tr>
<tr>
<td>Blade area, m$^2$</td>
<td>3.44</td>
</tr>
<tr>
<td>Air density, kg/m$^3$</td>
<td>1.228</td>
</tr>
<tr>
<td>Observer distance, m</td>
<td>62.2</td>
</tr>
<tr>
<td>Mean lift coefficient</td>
<td>0.182</td>
</tr>
</tbody>
</table>

From Eqn. 4.32, case 1 has a peak frequency at

$$f_p = -240 \log_{10}(69420N) + 2.448 \times (3.5\text{rev/sec} \times 2\pi \text{rad/rev} \times 9.45\text{m}) + 942$$

$$= 288.8 \text{ Hz}$$

which falls in the 1/3-octave band with a center frequency of 315 Hz. Thus, the SPL predictions for case 1 are applied at the 1/3-octaves which have center frequencies of $315 \times 2^i$ where $i$ ranges from -2 to 5 given the upper and lower frequency limits of 50 and 10,000 Hz, respectively. Similarly, case 2 has a peak frequency at

$$f_p = -240 \log_{10}(5206N) + 2.448 \times (21.1\text{rev/sec} \times 2\pi \text{rad/rev} \times 1.52\text{m}) + 942$$

$$= 543.3 \text{ Hz}$$

which falls in the 1/3-octave band with a center frequency of 500 Hz. Thus, the SPL predictions for case 2 are applied at the 1/3-octaves which have center frequencies of $500 \times 2^i$ where $i$ ranges from -3 to 4.

### 4.3.2 Method by Brooks, Pope and Marcolini

The preliminary report issued by BPM includes a series of rotor test cases performed at the Duits-Nederlandse Windtunnel (DNW). Noise levels were mea-
(a) Comparison of results for Case 1 of Pegg’s method.

(b) Comparison of results for Case 2 of Pegg’s method.

Figure 4.18: Comparison of results for Pegg’s Method.
Table 4.3: Flight conditions for test cases with BPM’s method

<table>
<thead>
<tr>
<th>Case</th>
<th>$C_T$</th>
<th>Ω (RPM)</th>
<th>$\alpha_{TPP} (\degree)$</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>1050</td>
<td>-20</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.0044</td>
<td>1050</td>
<td>-20</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>0.0044</td>
<td>1050</td>
<td>-20</td>
<td>0.173</td>
</tr>
<tr>
<td>4</td>
<td>0.0044</td>
<td>1050</td>
<td>-2.5</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>0.0066</td>
<td>1050</td>
<td>-10</td>
<td>0.086</td>
</tr>
<tr>
<td>6</td>
<td>0.0044</td>
<td>525</td>
<td>0</td>
<td>0.086</td>
</tr>
<tr>
<td>7</td>
<td>0.0044</td>
<td>525</td>
<td>-20</td>
<td>0.086</td>
</tr>
</tbody>
</table>

sured for numerous flight conditions at multiple observer positions. A representative sample of these cases were used to validate the implementation of BPM’s code in PSU-WOPWOP. The flight conditions used by each case are provided in Table 4.3. Note that for this work BPM reference $\alpha_{TPP}$ to the undisturbed streamline axis.

4.3.2.1 Results

The DNW tests were performed using a 40 percent scale, four bladed, hingeless BO 105 rotor which uses an NACA 23012 airfoil. The validation cases presented here used a Boeing VR-12 airfoil, due primarily to the availability of the data in the rotor aerodynamics code. The profiles of the two airfoils are shown in Fig. 4.19. The two airfoils are sufficiently similar in their performance characteristics so as not to be a significant factor in the validation process. The data used to validate the implementation of BPM’s method in PSU-WOPWOP was obtained from a trim code which used the flight conditions given in Table 4.3 to find a steady-state operating condition. The flight conditions were determined to be steady-state when the thrust matched the weight ($T = W$), where the weight was given as the thrust needed for the required $C_T$ (see Eqn. 4.28).

The results show good general agreement between the predictions published by BPM and those obtained from the PSU-WOPWOP implementation. Exact agreement should not be expected considering that the noise predictions are
for two different airfoils, yet there is sufficient similarity in both the scale and trend of the predicted noise levels to demonstrate the validity of the method’s implementation.

The predictions from PSU-WOPWOP typically are within +/-3 dB of those made by BPM. The most significant disimilarities are seen in cases 1, 2 and 3 at frequencies between 2-5 kHz in which BPM predicts spikes in the noise levels that do not appear in the PSU-WOPWOP results. Further work is likely needed to identify the mechanism(s) responsible for these differences.
Figure 4.20: Comparison of results for case 1 of BPM’s method. Comparison is between closed symbols.
Figure 4.21: Comparison of results for case 2 of BPM’s method. Comparison is between open symbols.
Figure 4.22: Comparison of results for case 3 of BPM’s method. Comparison is between closed symbols.
Figure 4.23: Comparison of results for case 4 of BPM’s method. Comparison is between open symbols.
Figure 4.24: Comparison of results for case 5 of BPM’s method. Comparison is between open symbols.

(a) BPM results

(b) PSU-WOPWOP results
Figure 4.25: Comparison of results for case 6 of BPM’s method. Comparison is between open symbols.
Figure 4.26: Comparison of results for case 7 of BPM’s method. Comparison is between closed symbols.
4.4 Summary

The two methods have been shown to predict broadband noise levels in good agreement with the original reports. Further work is needed to rigorously test their implementation and determine their limitations, but the work presented in this chapter demonstrates they apparently have been implemented correctly.

The semi-empirical broadband noise prediction methods presented in this chapter are well suited for preliminary analysis of the broadband noise produced by most rotors presently used in industry. These methods allow for efficient and accurate noise predictions valid for FAA civil noise certification. Much of the work needed for their implementation in PSU-WOPWOP forms the foundation for future work involving broadband noise prediction from first principles. The addition of a first-principles broadband noise prediction method will complete the assembly of a multi-tier array of prediction methods each providing their own benefits and fidelity level.
Chapter 5

FAA Noise Certification Package

The following defines the problem of an automated FAA noise certification package. The major topics discussed are:

1. Design and Implementation of the FAA Noise Certification Package: Discussion is presented on the design of a new noise certification package which includes the broadband methods discussed in Chapter 4 and additional capabilities discussed in the following chapter. Revisions, limitations and details of the final product are discussed.

2. Testing and Demonstration of the FAA Noise Certification Package: Cases are designed to test the capabilities of the new package. Results are discussed and detailed.

5.1 Problem Description

The two semi-empirical methods discussed in Chapter 4 were built into PSU-WOPWOP as part of an FAA noise certification package. Significant automation of the code was included in this package through the use of pre-set and assumed case conditions to allow for rapid data analysis by eliminating the previous need to set these conditions for every case. In addition, the pre-set conditions adhere to the requirements for FAA noise certification, thereby helping to ensure that the results of a case are valid for FAA noise certification.
5.2 FAA Noise Certification

All commercial rotorcraft in the United States must be certified by the FAA before they may be used for the transportation of persons or property. Full certification requires approval of an vehicle’s electrical, engine, flight and, most important for this thesis, noise performance. The noise performance of rotorcraft is evaluated in three different steady-operating flight conditions: takeoff, overflight (flyover), and approach (see Fig. 5.1). A vehicle may be certified by the FAA if its effective perceived noise level (EPNL), measured in EPNdB, is below the maximum allowable level set for each flight condition. Details regarding the FAA certification and EPNL calculation processes can be found in [26].

5.2.1 Flight Paths

The first step in the automation of FAA noise certification was to design a collection of flight paths corresponding to the flight conditions required for certification (see Fig. 5.1). However, the exact operating conditions for these flight paths is case-dependent, thus it became necessary to design a way for users of PSU-WOPWOP to input and change the settings for a flight path. As soon as user input became a part of the flight path design, it became obvious that the concept of flight paths needed to be more generalized. While the flight paths necessary for FAA noise certification can all be approximated using a constant velocity vector, the flight paths built in PSU-WOPWOP would need to allow for varying flight conditions throughout a case.
5.2.1.1 Design

Prior to this research, PSU-WOPWOP allowed for motion to be input as constant, periodic or aperiodic. Constant motion may approximate the conditions necessary for FAA noise certification but obviously does not allow for flight conditions to vary, making it unsuitable for more realistic analysis. Periodic motion is simply inappropriate in this application. Aperiodic motion, motion that cannot be described as occurring with periodicity, will certainly work but it was designed for motion of significantly higher resolution than was necessary for this project. Thus, flight paths act as a middle-ground between constant and aperiodic motion, allowing for flight conditions to vary in a case, thereby approximating aperiodic motion, yet requiring significantly less user-input to accomplish this than is necessary for aperiodic motion (see Fig 5.2).

To accomplish this, flight paths were broken into segments where the ends of the segments are known as “waypoints”. Flight paths have a minimum of two waypoints but no maximum. Each waypoint is defined by a source time, position and velocity. Beginning from the first waypoint at the given source time, the vehicle travels in a straight line to the next waypoint position at the velocity defined at the previous waypoint. The first waypoint requires the input of a source time to anchor the flight path in time, but the source time’s of the other waypoints are calculated within PSU-WOPWOP since \( t = \frac{x}{v} \). A vehicle’s velocity \( v \) in a segment is determined by applying the user-defined
flight velocity $v_{user}$ along the unit vector $\hat{x}$ pointing from the previous to the upcoming waypoint:

$$v_{new} = |v_{user}| \cdot \hat{x}$$

(5.1)

$v_{new}$ automatically corrects any problems in the motion if $v_{user}$ is not in the correct direction to reach the next waypoint.

### 5.2.2 Computation of EPNL

The basic element in the FAA civil noise certification criteria is the effective perceived noise level, EPNL, which is a single number evaluator of the subjective effects of aircraft noise on humans. EPNL consists of instantaneous perceived noise level, PNL, which is corrected for spectral irregularities and duration. The correction, called the “tone correction factor”, is applied only for the maximum tone at each increment of time. PNL and the tone corrected PNL, PNLT, are measured in half-second segments of observer time such that

$$PNLT(k) = PNL(k) + C(k) \quad k = 1, 2, 3, ..., n$$

(5.2)

where $C(k)$ is the tone correction factor and $k$ is a single segment of observer time.

The procedure for calculating EPNL consists of the following five steps, adapted from [26]:

- the 24 one-third octave bands of sound pressure level ranging from 44.7 Hz to 11.22 kHz are converted to perceived noisiness by means of a noy table. Noy, $N$, is the perceived noisiness at any instant of time that occurs in a specified frequency range. The noy values are combined and then converted to instantaneous perceived noise levels, PNL($k$) (see Fig. 5.3);

- $C(k)$ is calculated for each spectrum to account for the subjective response to the presence of spectral irregularities;

- the tone correction factor is added to the perceived noise level to obtain tone corrected perceived noise levels, PNLT($k$), at each half-second increment of time (see Eqn. 5.2).
The instantaneous values of tone corrected perceived noise level are derived and the maximum value, PNLTLM, is determined;

- a duration correction factor, \( D \), is computed by integrating the tone corrected perceived noise level versus time;

- effective perceived noise level, EPNL, is determined by the algebraic sum of the maximum tone corrected perceived noise level and the duration correction factor:

\[
EPNL = PNLTLM + D
\]  

(5.3)

A crucial requirement in the calculation of EPNL is that the limits of the time integration correspond to time segments in which the PNLT level is within 10 dB of the maximum PNLT (PNLTLM) (see Fig. 5.4).

![Figure 5.3: Perceived noise level as a function of total perceived noisiness [26].](image-url)
5.2.2.1 Automatic Extension of Observer Time and Flight Paths

When building a case to predict EPNL there is no way to know with certainty whether the PNLTM-10 dB requirements will be met. Previous versions of PSU-WOPWOP simply depended on the user to check to see if the PNLTM-10 dB criteria was met. Thus a case could run to completion and users would still have to analyze the results to verify that the PNLTM-10 dB requirements were met, thereby ensuring validity of the EPNL prediction. To help avoid such situations, a procedure was developed to extend the observer time range of a case (and hopefully meet the -10 dB requirements). An example of this procedure is as follows: a case intended for use in FAA civil noise certification is built and run. After the PNLT is computed for each of the desired time segments, but before computation of the EPNL, the PNLTM is found and the PNLT range is scanned to determine if the PNLTM-10 dB requirements were met. If either requirement was not met, the observer time range is extended accordingly and the case is rerun for a ‘new’ observer over the added time segment. At the end of the
run, the time histories of acoustic pressure for the 'old' and 'new' observers are combined (see Fig. 5.5) and the procedure starts over with determining whether PNLT-10 dB was met. If the -10 dB requirement is missed at the beginning of the observer time range a 'new' observer time range \([t'_{\text{min}} : t'_{\text{max}}]\) is defined with respect to the 'old' observer time range \([t_{\text{min}} : t_{\text{max}}]\) as

\[
t'_{\text{min}} = t_{\text{min}} - dt
\]

\[
t'_{\text{max}} = t_{\text{min}} - \Delta t
\]

Similarly, if the -10 dB requirement is missed at the end of the observer time range a 'new' observer time range is defined as

\[
t'_{\text{min}} = t_{\text{max}} + dt
\]

\[
t'_{\text{max}} = t_{\text{max}} + \Delta t
\]

where \(dt\) is the observer time step used in the case and \(\Delta t\) is typically defined as an integer multiple of 0.5 seconds (one PNLT segment). By defining a 'new' observer time range in this manner, 'old' and 'new' time ranges can be added together to form a continuous time history that may be continually expanded. Note that since this procedure may be repeated multiple times the term 'old' is used in place of 'original'.

### 5.2.2.2 Implication for Flight Paths

The capability to extend the observer time range means that there also needs to be a way to extend flight paths to ensure that a case does not break down or produce erroneous results if a new observer time range exceeds the limits of the original flight path. In such a case, the velocity vectors at the first and last waypoints are used to extend the flight path as far as is needed given the current observer time range (see Fig. 5.6). In this way, a flight path need only be defined once but can be used for as long as is necessary to capture PNLTM-10 dB and thus the EPNL of a case.
Figure 5.5: Example of observer time-range extension.

Figure 5.6: Example of flight path extension.

5.2.3 FAA Certification Namelist

Cases designed for FAA civil noise certification are usually quite similar in their setup, particularly with regard to observer location, observer time range and the desired output from a run. Since each case in PSU-WOPWOP requires its own namelist file, the inclusion of the option to let users input a second namelist which holds all the terms they infrequently or never change would help remove
much of the monotony inherent in building up a case. Figure 5.7 illustrates just how much information in a typical case is often the same. In addition to speeding up the case setup, a second namelist helps to avoid errors by removing the variables which do not (or should not) change between cases.

5.3 Results

A series of cases were run to demonstrate the features of the FAA civil noise certification package including flight paths, observer time-range extension, flight path extension and broadband noise prediction.
5.3.1 Setup

To test the new FAA civil noise certification package three cases were setup; one for each of the flight profiles required for FAA civil noise certification (see Fig. 5.1). Each case uses its respective flight path profile and was intentionally designed to fail (at least) the first round of checks regarding the PNLT-10 dB requirement. The time range of the each flight path was set to the limits of the initial observer time range meaning that failure of the PNLT-10 dB requirement (and hence expansion of the observer time range) also tested the expandability of the flight profiles. Thus this compact design allowed for the testing of all three major features present in the FAA civil noise certification package.

Note that the loading data used for this case was generated from a simple trim code using blade-element theory. The results are not representative of an actual flight test, but rather just demonstrate the capabilities of the new features.

5.3.2 Results

The results of these tests are given as time histories of each case’s PNLT. Each case predicts PNLT for three observers as required for FAA civil noise certification. Figure 5.8 shows the results of the test cases when the initial observer time range was limited to 25 seconds and observer time expansion was not allowed. By not allowing expansion of the observer time the PNLT-10 dB requirements were not met for any of the cases. There is no way to meet the requirements when none of the cases even show a established peak in the noise levels. Yet the way in which the observer time expansion was designed allows for situations like this one. So long as a peak has not been found the PNLT-10 dB requirement is not met, thus the right hand side of the observer time range will be expanded until a peak is found, after which the expansion will continue until PNLT-10 dB is achieved.

The same cases were then tested while allowing observer time expansion, the results of which are shown in Fig. 5.9. As a result of the observer time expansion, every case met the PNLT-10 dB requirements. In some cases the PNLT levels dropped well below the PNLT-10 dB mark, a result of the size of the observer timestep used when expanding the time range. For these cases the
step size was set to 2.5 seconds, but PSU-WOPWOP does not know in advance whether this time range will be sufficient or more than is necessary to capture PNLTM-10 dB. As a result, some cases used a wider range of observer time than was necessary to capture PNLTM-10 dB but this can be avoided simply by reducing the step size of the observer time expansion.
Figure 5.8: Expanded view of the initial demonstration of the FAA civil noise certification package without observer time expansion plotted on the same scale as the full-length demonstration.
Figure 5.9: Demonstration of the FAA civil noise certification package with observer time expansion.
5.4 Summary

The FAA civil noise certification package was designed to make acoustic prediction as convenient and automated as possible. Use of the two semi-empirical broadband noise methods discussed in Chapter 4 makes broadband noise prediction extremely simple while eliminating the hassle and computational expense of using complex CFD models. The methods have been built into the FAA civil noise certification to be used in EPNL prediction. Flight paths have been incorporated which can be designed and modified rapidly. As well, the new capability to automatically extend the observer time range and flight path (if used) ensures that the PNLTM-10 dB requirement is met and the predicted EPNL will be valid regardless of whether the ‘original’ case achieved this requirement or not. The addition of an FAA Certification namelist lets users avoid the inconvenience of resetting the same variable definitions for every case, but also helps to ensure consistency between cases by keeping those terms separate and thus safe from modification.
Conclusion

The motivation behind this research has been the improvement of acoustic prediction through expansion of the range of predictable noise sources and the development of more efficient computational methods for general prediction.

6.0.1 Reduced-Memory Algorithm

The ongoing effort to accurately predict noise using permeable ADS over large computational domains offers numerous technical challenges. A need for high data resolution in such cases can quickly lead to enormous memory requirements if the data is not properly managed.

Acoustic prediction using permeable surfaces avoids the volume integral present in FW-H by capturing quadrupole sources within an acoustic data surface, which is fictitious (does not physically exist), that should include all the real acoustic sources. Theoretically, the FW-H method has the potential to predict noise exactly, but the predictive accuracy of this approach is limited by two factors: the spatial and temporal resolution of the input data and the accuracy of the input data, usually obtained from CFD. Both of these factors introduce potential errors. Experience with the application of permeable ADS has shown that data must be of high spatial and temporal resolution for accurate noise prediction, which is the reason for the enormous data requirements necessary in permeable ADS cases. But even with high resolution the input data must be of sufficient accuracy if an acoustic prediction code is to have any chance
at accurately predicting the noise. In fact, the CFD solution must accurately include even the very small (compared to the aerodynamic flow) acoustic pressure fluctuations at the ADS. To do this, high-order accurate discretization and high spatial and temporal resolutions are required.

The work described in this thesis did not cover the large computation times required to get the CFD data. Instead, it dealt with relieving the heavy burden of large data requirements needed for noise prediction using permeable ADS. This was accomplished efficiently by designing an algorithm which mimics noise prediction in real time. Treating the input data as a continuous stream, rather than a static block of data, this new algorithm provides dramatic reductions to memory requirements by storing only the data necessary to predict noise at a single time. An example of the benefits of this work is a case provided by Lockard [16] of NASA Langley which required only 20 MB of allocated memory using the new algorithm, compared to the more than 2 GB of memory required in previous versions of PSU-WOPWOP, which do not incorporate the new memory-saving algorithm.

6.0.2 Broadband Noise Prediction

Semi-empirical prediction methods suffer from neither of the problems mentioned earlier. Rather than requiring enormous amounts of data, semi-empirical methods need relatively small amounts of information about a case. Indeed, Pegg’s method (discussed in Section 4.2.1) calls for just six pieces of information, none of which require lengthy CFD analysis to obtain. Of course, unlike permeable surfaces, semi-empirical methods are limited in the breadth of their predictive capabilities. But there is practical benefit to such methods: as a first-round analysis semi-empirical methods offer reasonably accurate prediction with relatively little effort and minimal computation time. The two methods implemented in this project incorporate data from a wide range of test cases and are well suited for the majority of rotors currently used in industry.

The capability to let users provide input data directly or let PSU-WOPWOP compute it is powerful in two ways: Allowing users to provide input data directly makes the methods adjustable and can be used to increase the predictive
accuracy of a given method based upon experience. Using PSU-WOPWOP to compute input terms avoids potential user errors and maintains consistency in the computation of terms.

6.0.3 FAA Civil Noise Certification Package

Ultimately, it was a goal of this research to include these broadband prediction methods in a comprehensive package with the intention of generating FAA-certified noise predictions. This package includes a number of extremely useful new features including default namelists, easy-to-assemble flight paths and automatic observer time expansion and broadband noise prediction.

6.1 Summary of Contributions

6.1.1 Memory Management

A new algorithm was designed which dramatically reduces memory requirements for aeroacoustic prediction. Memory reductions were achieved by designing the algorithm to manage data as though it was working with data being streamed in real time: working with a small amount of data at a time before discarding old information and reading in new information. As such, this algorithm is ideally suited for use in a real-time noise prediction program.

6.1.2 Broadband Noise Prediction

Two semi-empirical broadband noise prediction methods were implemented in PSU-WOPWOP to provide preliminary analysis of broadband noise produced by rotors and propellers with minimal computational effort. These methods are well suited for the majority of rotor and propeller blades used in the industry.

6.1.3 FAA Noise Certification

A new package was built into PSU-WOPWOP to automate the noise prediction process for FAA civil noise certification. Included in this package is a number
of new features including default namelists, easy-to-assemble flight paths, automatic observer time expansion and broadband noise prediction. This package features a comprehensive collection of the tools needed to make noise prediction for FAA civil noise certification as easy as possible while providing simple methods for case modification and a great deal of flexibility in case setup.

6.2 Recommendations for Future Work

A great deal of work went into the research discussed in this paper, yet more can always be done. The following sections provide a brief discussion of future work that could improve the state of the current methods and algorithms.

6.2.1 Noise Prediction for Multiple Observers

PSU-WOPWOP currently computes pressure solutions for one observer at a time (see Fig. 3.9). This approach is ideal for computing with multiple processors in parallel since each processor can compute solutions for a different observer and the results require no post-processing. An alternative approach, and one perhaps better suited for work with multiple observers in serial processing, would be to make the iteration through observers the inner-most loop as shown in Fig. 3.10. While this design, in conjunction with the new algorithm, would help to reduce computation time in serial processing, it would make obtaining a solution from parallel processing a bit more complex. The pressure arrays returning from each slave would need to be reassembled and then interpolated in order to obtain the correct pressure histories. Ultimately, it may be necessary for code designers to include the capability to use either of these algorithms if users are to see reductions in run time and memory use regardless of how a case is setup.

6.2.2 Real Time Noise Prediction

The combination of the new reduced-memory algorithm and the new observer time interpolation routine makes it possible to mimic real-time noise prediction.
Figure 6.1: Routine currently used by PSU-WOPWOP to calculate pressure solutions.

Figure 6.2: Alternative routine for calculating pressure solutions.

But it is not true real-time noise prediction because the information being read into PSU-WOPWOP comes from a static file that already has the entire geometry and loading history for a case written and saved prior to the noise prediction. The results are not output in real time, either. But working from a static file provides a number of opportunities for keeping code simple. The current design is free to access the data files at any time, but a code designed to predict noise in real-time needs a management system which can both store data if it arrives before the program is ready to use it and pause the program if the data is delayed in its delivery (after the program is first ready to use it). Although such data management was not the focus of the research, it is a straightforward task to incorporate it now that the routines necessary to actually predict noise in real time have been developed.

6.2.3 Broadband Noise Prediction from First Principles

The semi-empirical broadband noise prediction methods discussed in Chapter 4 are well suited to predict broadband noise for a wide range of airfoils. Yet
both of these methods are limited in their application because they explicitly assume steady flow conditions and do not extend their prediction capability to novel blade shapes. While they are both a quick and efficient means of predicting broadband noise, these constraints limited their value when designing new blade sections with reduced noise signatures. For this application a much more rigorous approach is necessary. A powerful new approach is one which relies heavily on CFD to solve for both large and small scale turbulence responsible for generating broadband noise. Although it is much more computationally demanding, such an approach has the capability to accurately predict the noise generated by any conceivable shape assuming the correct inflow conditions are known. The permeable FW-H approach would almost certainly be used for such first-principles broadband noise prediction.
Appendix

Reducing Memory Requirements

In order to effectively reduce an algorithm’s memory requirements, its fundamental limits must be determined. In the case of Formulation 1A those limitations are set by the finite difference equations necessary to approximate terms required to evaluate the integrand, specifically the velocity, acceleration and loading derivative. Analysis has shown that these terms can be approximated with sufficient accuracy using 2\textsuperscript{nd} order-accurate equations. These equations require no more than 4 terms to approximate a given term. The new memory algorithm was designed around these equations, using their requirements to dictate its structure.

Computing derivatives for the first timestep requires the use of forward-difference equations such that

\[
v_j = \left( \frac{\partial y_j}{\partial \tau} \right) = \frac{-y_j^{i+2} + 4y_j^{i+1} - 3y_j^i}{2\Delta \tau} + O((\Delta \tau)^2) \quad (6.1a)
\]

\[
\dot{v}_j = \left( \frac{\partial^2 y_j}{\partial \tau^2} \right) = \frac{-y_j^{i+3} + 4y_j^{i+2} - 5y_j^{i+1} + 2y_j^i}{\Delta \tau^2} + O((\Delta \tau)^2) \quad (6.1b)
\]

Derivatives for the last timestep are obtained using similar backward-difference equations

\[
v_j = \left( \frac{\partial y_j}{\partial \tau} \right) = \frac{-3y_j^i + 4y_j^{i+1} - y_j^{i+2}}{2\Delta \tau} + O((\Delta \tau)^2) \quad (6.2a)
\]

\[
\dot{v}_j = \left( \frac{\partial^2 y_j}{\partial \tau^2} \right) = \frac{2y_j^i - 5y_j^{i-1} + 4y_j^{i-2} - y_j^{i-3}}{\Delta \tau^2} + O((\Delta \tau)^2) \quad (6.2b)
\]
Every other timestep in a case uses central-differencing (see Eqns. 3.3) to compute derivatives.

Obtaining the 2nd-order derivative (acceleration) for the first source timestep requires the code to have access to data from the first four source timesteps. Thus, the algorithm must begin by reading in that data. Once completed, the necessary derivatives can be obtained and solutions to Formulation 1A can be computed for the first source time. Without reading more data there is also enough information to calculate derivatives and Formulation 1A solutions for the second source timestep since \( \tau_1 \) requires data from timesteps \([1,2,3,4]\) and \( \tau_2 \) requires data from timesteps \([2,3,4]\). But solutions for the third source timestep require data from timestep \([5]\) which must be read in. Beginning with the third timestep, a routine is implemented which swaps out data from the oldest timestep for new data, limiting the memory storage to what is needed for four source timesteps (see Fig. 3.7). This routine uses memory pointers to perform the swapping, adjusting the memory location referenced by each pointer rather than physically moving the data in memory which is a much more intensive and time-consuming process. Finally, once data from the last timestep has been read into memory, the derivatives for the last timestep are obtained using Eqn. 6.2.

**Observer Time Interpolation**

PSU-WOPWOP uses a source-time-dominant algorithm to solve Formulation 1A. This type of algorithm calculates the observer time at which a signal arrives from a corresponding source time based on the equation \( t = r/c + \tau \). This approach requires that the predicted pressure history be interpolated over a global observer time array in order to organize the data in a coherent format. Previous implementations of the source-time-dominant algorithm stored the pressure data from each node and each observer time until the end of a case before performing this interpolation. Yet the memory required to store this information grows proportionally with the size of a case. Reductions in the memory requirements of the observer time interpolation routine may be obtained using an approach similar to the one designed for the acoustic prediction algorithm:
interpolating data on the fly as it becomes available.

The new procedure is as follows: First, the pressure from every source point at one source time is stored in memory. After the pressure results from a second source time become available the arrival times of the two signals from each point are compared to determine whether they fall over any of the time steps in the observer time array. If they do, then the acoustic pressures are interpolated and added to the current solution. If they do not, then the interpolation can be skipped altogether. Once the data from every point has been analyzed the oldest pressure data is discarded and replaced with the newest pressure data. This routine continues through the last source time in the case. The end result of this procedure is identical to the old method but only requires data to be stored for 2 source time steps rather than the entire source time range.

Initial implementation of this method was very memory efficient but took considerably longer to run than the original method. In order to find the observer times over which to interpolate the data, the new method was stepping through the observer time array in its entirety for each iteration of source time, severely increasing the run time. The problem was solved by determining the range of necessary observer times before interpolation begins. Assuming that the timesteps in the global observer time array are equally spaced, the range of observer times can be described as integers multiples of this timestep, indicating their locations in the array. This range can be determined by using the following equations to bracket the time range influenced by the current pressure data:

\[
N_{\text{min}} = \text{floor}\left(\frac{(t_1 - t_{\text{min}})}{\Delta t}\right) + 1 \quad (6.3a)
\]

\[
N_{\text{max}} = \text{ceiling}\left(\frac{(t_2 - t_{\text{min}})}{\Delta t}\right) + 1 \quad (6.3b)
\]

where \(t_{\text{min}}\) and \(\Delta t\) are the minimum observer time and the step size in the global observer time array, respectively. \(t_1\) is the observer time associated with the previous source time and \(t_2\) is the observer time associated with the current source time, such that \(t_2 > t_1\). Finally, \(N_{\text{min}}\) and \(N_{\text{max}}\) are the first and last observer time steps which can be influenced by the range of current observer times, specifically \([t_1 : t_2]\). The range of observer times over which to interpolate is \([t(N_{\text{min}}) : t(N_{\text{max}})]\). Determining the range of necessary global observer times
prior to interpolation speeds up this routine dramatically since the number of required interpolations \((N_{\text{max}} - N_{\text{min}} + 1)\) is typically a small fraction of the full observer time array.
Bibliography


[16] Lockard, D.P., personal communication


[26] ICAO, 999 University Street, Montréal, Quebec, Canada, *Environmental Protection, Volume 1 Aircraft Noise, 6th edition*, 2011