ABSTRACT

CARTER, MELISSA BETH. Time–Accurate Implementation of Lighthill’s Acoustic Analogy for Complex 3-D Jet Noise Prediction. (Under the direction of Dr. Jack Edwards.)

A new noise prediction code has been developed for the purpose of more accurate prediction of complex three-dimensional jets that are typical of realistic aircraft engine installations. The new code, Jet3D-UNS (Jet3D – Unsteady Navier Stokes) computes Lighthill’s acoustic analogy time accurate with the Partially Averaged Navier-Stokes and Large Eddy Simulation turbulence models. In contrast to the existing version of Jet3D that predicts noise from Reynolds Averaged Navier-Stokes CFD solutions, some of the advantages of the method used in Jet3D-UNS are that it does not use two-point correlations or calibration constants. Computations have been performed on a round dual flow nozzle with a bypass ratio of five with and without a pylon attached. The pylon adds three dimensional flow features to the jet that change the azimuthal and axial distribution of noise sources in the jet. For these two configurations, predictions were compared to experimental data. Results show that the unsteady solutions do not match the experimental data as well as the RANS solutions and the noise predictions are inferior to the original JET3D predictions. Further work will include integrating this code with other CFD software and other turbulence models.
Time – Accurate Implementation of Lighthill’s Acoustic Analogy for Complex 3-D Jet Noise Prediction

by
Melissa Beth Carter

A dissertation submitted to the Graduate Faculty of North Carolina State University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Aerospace Engineering

Raleigh, North Carolina
2015

APPROVED BY:

Dr. Jack Edwards
Committee Chair

Dr. Ashok Gopalarathnam

Dr. Hassan Hassan

Dr. Arkady Kheyfets
DEDICATION

To Joel, Joshua and Elizabeth Carter, and Susan and John Morehouse and in memory of Dr. Paul Pao. Thank you for always supporting and believing in me.
BIOGRAPHY

Mrs. Melissa B. Carter is an Aerospace Engineer at NASA Langley Research Center. She received her Bachelors of Science with Honors in Aerospace Engineering from the Pennsylvania State University in December 1999 and her Masters of Science in Aerospace Engineering from George Washington University in May 2004. She began work at NASA Langley Research Center as a co-op student in 1998, and was hired on permanently in 2000. Her research has included computational and experimental studies of aerodynamic characteristics of aircraft. In addition to the acoustic work described in this dissertation, she has worked to improve NASA’s in-house sonic boom prediction capability. Currently Mrs. Carter is serving as the inlet distortion subject matter expert for the NASA Environmentally Responsible Aviation Project’s experimental investigation of a Hybrid Wing Body. Mrs. Carter is a Senior Member of the American Institute of Aeronautics and Astronautics (AIAA) and very active in the local section, having served as Section Chair during the 2011-2012 session. To date Mrs. Carter has authored or co-authored 22 published papers in the aerospace field.
ACKNOWLEDGEMENTS

The author would like to thank NASA Langley Research Center for their support of my research and class work. In particular, thanks to Laurence Leavitt, Zachary Applin, and Richard Wahls for their support and encouragement for obtaining my doctorate. I would also like to thank Dr. Russell Thomas for providing and supporting my dissertation research project. Additionally, I would never have been able to complete this work without the help of my ‘Ph.D. dream team’, Dr. Khaled Abdol-Hamid, Dr. Alaa Elmiligui, Dr. Paul Pao, Dr. Craig Hunter, and Dr. Jeremy Pinier. I would also like to thank my coworkers in the Configuration Aerodynamics Branch for their support, friendship, and, when necessary, their commiseration.

Dr. Jack Edwards, thanks for your patience and all the guidance you provided. Thank you also to my committee members for all of their feedback and hard work.

Finally, I would like to thank my family for their encouragement and support. This dissertation has been long coming and their love and support through the years have meant a lot to me.
TABLE OF CONTENTS

LIST OF TABLES .................................................................................................................. vii
LIST OF FIGURES ................................................................................................................. viii
LIST OF SYMBOLS .............................................................................................................. xi
LIST OF ABBREVIATIONS ................................................................................................... xv

Chapter 1: Introduction ........................................................................................................ 1
  1.1 Motivation .................................................................................................................... 1
  1.2 Historical Background ................................................................................................. 1
  1.3 JET3D .......................................................................................................................... 3
  1.4 Organization ............................................................................................................... 5

Chapter 2: Lighthill Stress Tensor Development ................................................................. 8
  2.1 Lighthill Equation Development ............................................................................... 8
  2.2 Green’s Function ....................................................................................................... 9
  2.3 Moving Reference Plane ......................................................................................... 11
  2.4 Post Processing ....................................................................................................... 15
  2.5 Corrections .............................................................................................................. 16

Chapter 3: Computational Fluid Dynamics Software and Turbulence Models .................. 20
  3.1 Computational Fluid Dynamics Program .................................................................. 20
  3.2 PANS Turbulence Model ......................................................................................... 23
  3.3 Hybrid RANS/LES Turbulence Model ..................................................................... 25

Chapter 4: Implementation ................................................................................................. 27
4.1 Implementation of JET3D-UNS within PAB3D .......................................................... 27
4.2 JET3D-UNS Approach to Retarded Time ......................................................................... 30
Chapter 5: Test Cases .................................................................................................................. 32
  5.1 Test Facility and Data Collection Information .................................................................. 32
  5.2 Configuration Information and Test Conditions ................................................................. 33
Chapter 6: Computational Grids .................................................................................................. 36
Chapter 7: Lighthill Stress Tensor Component Study ............................................................. 41
  7.1 $T_{ij}$ Calculations .............................................................................................................. 41
  7.2 $T_{ij}$ Component Study ...................................................................................................... 41
Chapter 8: Data Processing ........................................................................................................ 47
Chapter 9: Computational Results .............................................................................................. 53
Chapter 10: Concluding Remarks and Future Work ................................................................. 69
REFERENCES .............................................................................................................................. 71
APPENDIX .................................................................................................................................. 75
Appendix A – JET3D-UNS OUTLINE .......................................................................................... 76
Appendix B – PREGRID PROCESSOR CODE ............................................................................. 77
Appendix C – USER.CONT JET3D-UNS INPUT ........................................................................ 82
Appendix D – JET3D-UNS POST PROCESSOR ......................................................................... 84
LIST OF TABLES

Table 5-1 Test Conditions.......................................................................................................................... 33
LIST OF FIGURES

Figure 1-1 RANS Results for Dual Flow Jet (Configuration 1) Showing Collapse of Core Flow at Approximately 5 Diameters Downstream of the Jet Exit................................. 6

Figure 1-2 Predicted Results from JET3D Compared to Experimental Data for Dual Flow Jet (Configuration 1) [Adapted from Ref. 1.10]..................................................... 7

Figure 2-1 Moving Axis Transformation [Adapted from Ref. 2.6]................................. 19

Figure 5-1 NASA Langley Low Speed Aeroacoustic Wind Tunnel (dimensions in inches) [from Ref. 5.2] ........................................................................................................ 34

Figure 5-2 Round Core and Fan Nozzle Geometries Tested [from Ref. 5.3]................... 34

Figure 5-3 Configuration 1 Installed on the Jet Engine Simulator (JES) in the NASA Langley Low Speed Aeroacoustic Wind Tunnel [from Ref. 5.3]................................. 35

Figure 6-1 Configuration 1 – Full Grid.............................................................................. 37

Figure 6-2 Configuration 1 – Center Z-Cut of Grid Block Structure ............................. 37

Figure 6-3 Configuration 1 – Center Z-Cut of Grid Cells ............................................. 38

Figure 6-4 Configuration 2 – Full Grid ................................................................................ 38

Figure 6-5 Configuration 2 – Center Z-Cut of Grid Block Structure ............................. 39

Figure 6-6 Configuration 2 – Center Z-Cut of Grid Cells ............................................. 39

Figure 6-7 Configuration 1 – Residual Convergence ..................................................... 39

Figure 6-8 Configuration 2 – Residual Convergence ..................................................... 40

Figure 6-9 Observer Locations Relative to Computational Domain (dimensions in feet) ....... 40
Figure 7-1 Comparison of Lighthill Stress Tensor Components Effect on Pressure ............ 44
Figure 7-2 Comparison of Lighthill Stress Tensor Components Effect on Sound Pressure Levels ........................................................................................................................................ 44
Figure 7-3 Comparison of Pressure Contours at Three Different Time Steps .................. 45
Figure 7-4 Comparison of Density Times Velocity Squared Contours at Three Different Time Steps ........................................................................................................................................ 46
Figure 8-1 Comparison of Configuration 2 Results with Different Time Steps ............... 50
Figure 8-2 Comparison of Configuration 2 Results with Different Data Processing Sizes .... 51
Figure 8-3 Comparison of Configuration 2 Results with Different Numbers of Sub-iterations ........................................................................................................................................ 51
Figure 8-4 Comparison of Configuration 2 Results with Delayed Data Analysis ............. 52
Figure 8-5 Comparison of Configuration 1 Results with Different Grid Sizes ................. 52
Figure 9-1 U-Velocity Cuts from the PANS Solution Downstream of the Configuration 1 Jet ........................................................................................................................................ 56
Figure 9-2 U-Velocity Cuts from the PANS Solution Downstream of the Configuration 2 Jet ........................................................................................................................................ 57
Figure 9-3 Centerline Streamwise Velocity Profile from the PANS Solution ................... 58
Figure 9-4 Normalized total temperature cross sections at X/D=2, 5, 10, 17 for Configuration 1 Jet [Adapted from Ref. 9.1] ........................................................................................................................................ 59
Figure 9-5 Centerline Normalized Unsteady Solution Total Temperature Profile for the Configuration 1 Jet ........................................................................................................................................ 60
Figure 9-6 Centerline Normalized Time-Averaged Solution Total Temperature Profile for the Configuration 1 Jet.................................................................61

Figure 9-7 Predicted Noise Levels for Configuration 1.........................................................62

Figure 9-8 Predicted Noise Levels for Configuration 2.........................................................63

Figure 9-9 Noise Production Histogram Configuration 2.......................................................64

Figure 9-10 Noise Production Histogram Configuration 2 – Low Frequencies .....................64

Figure 9-11 Noise Production Histogram Configuration 2 – High Frequencies .................65

Figure 9-12 Noise Source Plot Configuration 2 JET3D.........................................................65

Figure 9-13 Turbulent Kinetic Energy Plot Configuration 2 JET3D .................................66

Figure 9-14 Turbulent Kinetic Energy Plot Configuration 2 JET3D-UNS .........................67

Figure 9-15 Application of Turbulence Model for Configuration 1 PANS CFD Solution ...68
**LIST OF SYMBOLS**

**English Symbols**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>a</code></td>
<td>absorption coefficient (dB/m)</td>
</tr>
<tr>
<td><code>c</code></td>
<td>speed of sound (m/s) <code>\sqrt{\gamma RT}</code></td>
</tr>
<tr>
<td><code>C_p</code></td>
<td>pressure coefficient <code>((p-p_\infty)/(0.5\rho_\infty u_\infty^2))</code></td>
</tr>
<tr>
<td><code>C_\mu</code></td>
<td>constant (0.09)</td>
</tr>
<tr>
<td><code>C_{\varepsilon_1}</code></td>
<td>constant (1.44)</td>
</tr>
<tr>
<td><code>C_{\varepsilon_2}</code></td>
<td>constant (1.92)</td>
</tr>
<tr>
<td><code>C_h</code></td>
<td>model coefficient</td>
</tr>
<tr>
<td><code>D</code></td>
<td>diameter of jet exit (m)</td>
</tr>
<tr>
<td><code>e</code></td>
<td>specific internal energy (J)</td>
</tr>
<tr>
<td><code>f</code></td>
<td>acoustic frequency (Hz)</td>
</tr>
<tr>
<td><code>f_2</code></td>
<td>blending function</td>
</tr>
<tr>
<td><code>f_d</code></td>
<td>damping function</td>
</tr>
<tr>
<td><code>f_{\varepsilon}</code></td>
<td>ratio of unresolved to total dissipation</td>
</tr>
<tr>
<td><code>f_k</code></td>
<td>ratio of unresolved to total kinetic energy</td>
</tr>
<tr>
<td><code>f_{r,o}</code></td>
<td>relaxation frequency of oxygen</td>
</tr>
<tr>
<td><code>f_{r,N}</code></td>
<td>relaxation frequency of nitrogen</td>
</tr>
<tr>
<td><code>h_r</code></td>
<td>relative humidity (%)</td>
</tr>
</tbody>
</table>
\( k \)  
Turbulent kinetic energy (J)

\( k_u \)  
Kinetic energy sub-filter

\( L_T \)  
Characteristic turbulence length scale

\( M \)  
Mach number

\( n \)  
Normal distance

\( p \)  
Pressure (N/m\(^2\))

\( P \)  
Ambient atmosphere pressure (N/m\(^2\))

\( P_o \)  
Reference pressure (1.013x10\(^5\) N/m\(^2\))

\( P_{ij} \)  
Viscous stress tensor

\( p_{ref} \)  
Reference pressure for SPL calculations (20 \( \mu \)Pa)

\( P_{sat} \)  
Partial pressure of saturated water vapor (N/m\(^2\))

\( q_i \)  
Heat-flux vector

\( R \)  
Gas constant (287 J/kgK)

\( r \)  
Distance between observer and noise source (abs(x-y))

\( T \)  
Temperature (K)

\( T_{01} \)  
Reference temperature (273.16 K)

\( T_o \)  
Reference temperature (293.15 K)

\( t \)  
Time (s)

\( T_{ij} \)  
Lighthill stress tensor

\( u_{ib}, v_i \)  
Velocity vector (m/s)

\( x \)  
Location of the observer
$x_i$  direction vector

$y$  location of the noise source

Greek Symbols

$\alpha$  constant (0.58)

$\gamma$  ratio of specific heat (1.4 for air)

$\Delta$  LES filter width

$\Delta t$  time step

$\delta_{ij}$  Kronecker delta (=1 when $i=j$, =0 when $i \neq j$)

$\varepsilon$  dissipation per unit mass

$\varepsilon_u$  dissipation sub-filter

$\eta$  correlation separation vector in fixed reference axis

$\theta$  momentum thickness

$\Lambda$  damping parameter

$\lambda$  unresolved characteristic ratio

$\mu$  viscosity (kg/ms)

$\nu_l$  laminar eddy viscosity

$\nu_t$  kinematic eddy viscosity ($m^2/s$)

$\pi$  pi (~3.14159)

$\rho$  density (kg/m$^3$)
\( \sigma_\varepsilon \) constant (1)

\( \sigma_k \) constant (1.4)

\( \tau \) retarded time \( (t-r/c_0) \)

\( \tau_{ij} \) viscous stress tensor

\( \varphi \) observer angle

\( \Omega \) vorticity magnitude

Other Symbols

\( \nabla \) gradient \( (\partial/\partial x_i) \)

Subscripts

\( c \) convection

\( \text{core} \) core flow of the jet

\( e \) jet exit

\( o, \infty \) free-stream

\( t \) turbulent

Superscripts

\( \wedge \) Fourier transform

\( * \) complex conjugate

\( L \) linear

\( \text{NL} \) non-linear
## LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPR</td>
<td>by pass ratio</td>
</tr>
<tr>
<td>DES</td>
<td>Detached Eddy Simulation</td>
</tr>
<tr>
<td>DNS</td>
<td>Direct Numerical Simulation</td>
</tr>
<tr>
<td>JES</td>
<td>Jet Engine Simulator</td>
</tr>
<tr>
<td>LARC</td>
<td>Langley Research Center</td>
</tr>
<tr>
<td>LSAWT</td>
<td>Low Speed Aeroacoustic Wind Tunnel</td>
</tr>
<tr>
<td>LES</td>
<td>Large Eddy Simulation</td>
</tr>
<tr>
<td>NASA</td>
<td>National Aeronautics and Space Administration</td>
</tr>
<tr>
<td>NPR</td>
<td>nozzle pressure ratio</td>
</tr>
<tr>
<td>PANS</td>
<td>Partially Averaged Navier-Stokes</td>
</tr>
<tr>
<td>RANS</td>
<td>Reynolds Averaged Navier-Stokes</td>
</tr>
<tr>
<td>URANS</td>
<td>Unsteady Reynolds Averaged Navier-Stokes</td>
</tr>
<tr>
<td>SMS</td>
<td>space marching scheme</td>
</tr>
<tr>
<td>SPL</td>
<td>sound pressure level</td>
</tr>
</tbody>
</table>
Chapter 1: Introduction

1.1 Motivation

The motivation for the research discussed in this paper came from a change within NASA. The NASA Aeronautics Research Mission Directorate research is focused on solving challenges that still exist within the aeronautics community [1.1]. The challenges include air traffic congestion, safety and environmental impacts. In order to do this, NASA researchers are now focused on the ability to predict, test and validate new techniques and tools to help the aeronautics community.

Consequently, jet noise continues to be a primary concern for today’s aeronautical engineer. Federally mandated regulation of noise during takeoff and landing are limitations that must be met with little to no impact on the aircraft’s efficiency or performance. As these requirements tighten, the need to understand the noise sources and how to mitigate them becomes more important. This led to a push for an improved jet noise prediction capability as part of an overall high-fidelity CFD analysis of a proposed aircraft and consequently the research discussed in this paper.

1.2 Historical Background

A considerable amount of the currently used noise prediction schemes are based on acoustic analogies. An acoustic analogy is where the equations of motion are manipulated to solve for acoustic effects. Lighthill [1.2] and Lilley [1.3] developed the two most commonly used and analyzed analogies. Lighthill’s analogy [1.4], published in 1952, uses the Navier-
Stokes equations and reduces them to a linear wave equation where the jet is represented by a distribution of quadrupole sources. The dominant part of the source term is quadratic in velocity and can be decomposed into a mean plus a fluctuating component [1.5]. Therefore, this component consists of linear and quadratic fluctuating velocity components. Although claimed to be an exact solution, simplifying assumptions concerning the acoustic sources in the jet, which are discussed further in the next section, usually occur while implementing Lighthill’s analogy.

Lilley’s analogy, published in 1958, although similar to Lighthill’s approach, stated that the linear terms of the dominant source term do not radiate any sound and should not be included. In his analogy [1.3], the source term is considerably more complicated, containing the sum of a quadrupole based on the fluctuating velocities and a dipole that is proportional to the temperature fluctuations [1.5]. Due to this source, implementing Lilley’s analogy is extremely difficult and requires significant simplification.

Although there are numerous acoustic analysis tools derived from both of these analogies [1.6], most of them depend on a Reynolds averaged Navier-Stokes (RANS) solution where the characteristic scales of turbulence are usually obtained from a two-equation turbulence model. Other models invoke acoustic analogies for the turbulent mixing noise but still base their flow fundamentals on RANS solutions [1.7]. This research will focus on using Lighthill’s analogy as implemented in NASA Langley Research Center’s JET3D code and solving for jet noise using unsteady Navier-Stokes CFD.
1.3 JET3D

NASA Langley Research Center’s JET3D code, created in 2001, has been an extremely useful tool for noise calculations [1.8]. The code is a post-processor built on the Lighthill Stress Tensor and models the components using two-point, space-time correlations. JET3D requires a converged steady-state RANS CFD solution as input. Although the derivation of the Lighthill Stress Tensor and the corresponding Green’s function solution for the far field are contained in section 2.1, they are shown here in Eqs. 1.1 and 1.2, respectively.

\[ T_{ij} = p\delta_{ij} - c_o^2\rho\delta_{ij} - P_{ij} + \rho u_i u_j \]  

(1.1)

\[ p(\hat{x},t) = \frac{1}{4\pi c_o^2 \int x} \int \hat{r}_j \frac{\partial^2}{\partial t^2} T_{ij}(\hat{y},t-r/c_o) \]

The Lighthill Stress tensor can be broken into four parts. The first two components of the tensor contain the stresses due to fluctuating pressure and density, respectively. The third component consists of viscous stresses due to strain rate and dilatation while component four consists of stresses due to momentum flux. Lighthill neglected both components one and two based on the assumption that those fluctuations are generally small [1.4]. The equation has been further simplified by dropping component three since it is an order of magnitude smaller than component four. Consequently, JET3D, like most Lighthill analogy codes, simplifies Eq. 1.1 to just component 4 (Eq. 1.2).

\[ T_{ij} \approx \rho u_i u_j \]  

(1.2)
As noted by Dr. Hunter [1.9], creator of JET3D, while the simplification of the tensor can be determined by Reynolds stress calculations, errors may exist due to the neglected entropy fluctuations and viscous stresses. Additionally, Lighthill’s work was based on cold jets. The jets used for this study will be hot jets and may affect the accuracy of the assumptions (Chapter 7 will discuss this further).

In order to implement Eq. 1.2, the velocity was separated into a steady state mean flow component and a fluctuating turbulent component while density was assumed to be steady. Therefore in the JET3D solution, the Lighthill stress tensor contains two components: the shear noise correlation due to the noise generated by the interaction of the mean flow and velocity fluctuations and the self noise correlation due to the noise generated from the velocity fluctuations on their own.

Since JET3D uses the solution from a converged steady-state RANS CFD solution, the calculation of the Lighthill tensor requires modeling using two-point space-time correlations for the shear and self-noise described above. The code uses Taylor series expansions to compute the mean flow correlations for velocity and density. The turbulent velocity correlations are computed using local one-point correlation (from a Reynolds stress model) and a combination of Gaussian-type exponential functions and a quadratic function [1.10]. These solutions are then integrated to solve for the shear and self induced acoustic pressure, which is then put through a Fourier transform to obtain the spectral density of the acoustic pressure.
Although JET3D has been extremely useful, there was concern about the accuracy of the results due to the steady state nature of the solution. Figure 1-1 plots the turbulent kinetic energy results from a PAB3D RANS solution of a 2-stream jet engine. Approximately 5 diameters downstream of the jet exit, where the flow should be unsteady, the core flow of the engine collapses. Please note Figure 1-1 was provided through a private communication with Dr. Hunter. Figure 1-2 shows the noise (frequency vs. decibel) both measured experimentally and computed by JET3D at several observer angles (90-degrees is perpendicular to the jet flow). The JET3D results have an artificial hump in the results, which is believed to be due to the collapse of the core flow seen in figure 1.1.

In order to address the artificial hump and answer remaining questions concerning the validity of the correlations used in JET3D, Lighthill’s stress tensor has been added directly into the computational fluid dynamics (CFD) code PAB3D to compute unsteady noise. The new code, JET3D-UNS (Jet3D – Unsteady Navier Stokes) computes the entire Lighthill’s tensor (Eq. 1.1) using time accurate results from either Partially Averaged Navier-Stokes (PANS) or hybrid Large Eddy Simulation (LES) turbulence models. Some of the advantages of this method are that it does not use two-point correlations, or calibration constants.

1.4 Organization

This paper will present the development of the Lighthill stress tensor, the post processing and the standard corrections applied to the data in chapter 2. Chapter 3 summarizes details concerning the CFD code PAB3D and the turbulence models used during
this study. The implementation of JET3D-UNS and its unique approach to retarded time is presented in chapter 4. Details concerning the experimental data are discussed in chapter 5. Computational grid information is presented in chapter 6, while chapter 7 studies the effect of the four components of Lighthill’s stress tensor and compares the difference between how JET3D and JET3D-UNS calculates Lighthill’s stress tensor. Chapter 8 discusses the post processing of the data and the computational results are presented in chapter 9. Finally, chapter 10 contains the concluding remarks and plans for future work. The papers referenced within each chapter are listed at the end of the paper and are organized by chapter.

Figure 1-1 RANS Results for Dual Flow Jet (Configuration 1) Showing Collapse of Core Flow at Approximately 5 Diameters Downstream of the Jet Exit
Figure 1-2 Predicted Results from JET3D Compared to Experimental Data for Dual Flow Jet (Configuration 1) [Adapted from Ref. 1.10]
Chapter 2: Lighthill Stress Tensor Development

2.1 Lighthill Equation Development

The development of Lighthill’s acoustic analogy has been well documented and applied. For this publication, the derivation comes from references 2.1, 2.2, 2.3, 2.4 and 2.5. The derivation of Lighthill’s equation begins with the exact equation of continuity (Eq. 2.1) and the exact equation of momentum (Eq. 2.2), assuming no external forces and an arbitrary continuous medium. $P_{ij}$ is the viscous stress tensor representing the force in the $x_i$ direction acting on a portion of fluid, per unit surface area with inward normal in the $x_j$ direction.

\[
\frac{d\rho}{dt} + \frac{\partial}{\partial x_i}(\rho v_i) = 0 \tag{2.1}
\]

\[
\frac{\partial}{\partial t}(\rho v_i) + \frac{\partial}{\partial x_j}(\rho v_i v_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial P_{ij}}{\partial x_j}
\]

where $P_{ij} = \mu \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial v_k}{\partial x_k} \right)$ \tag{2.2}

By moving the second term to the right hand side and adding $c_o^2 \partial \rho / \partial x_i$ to both sides of Eq. 2.2, Eq. 2.3 is formed:

\[
\frac{\partial}{\partial t}(\rho v_i) + c_o^2 \frac{\partial \rho}{\partial x_i} = -\frac{\partial p}{\partial x_i} + c_o^2 \frac{\partial \rho}{\partial x_i} - \frac{\partial}{\partial x_j}(\rho v_i v_j - P_{ij})
\]

\[
= -\frac{\partial}{\partial x_j}((p - c_o^2 \rho)\delta_{ij} - P_{ij} + \rho v_i v_j) \tag{2.3}
\]

which can be rewritten in terms of the Lighthill stress tensor $T_{ij}$:
\[
\frac{\partial}{\partial t} (\rho v_i) + c_o^2 \frac{\partial \rho}{\partial x_i} = - \frac{\partial T_{ij}}{\partial x_j}
\]

where \( T_{ij} = \left( p - c_o^2 \rho \right) \delta_{ij} - P_{ij} + \rho v_i v_j \) \hspace{1cm} (2.4)

Next take the partial derivative in respect to time of Eq. 2.1 and the partial derivative with respect to \( x \) of Eq. 2.3 while substituting in the definition of the Lighthill stress tensor (Eq. 2.4).

\[
\frac{\partial}{\partial t} (2.1) => \frac{\partial^2 \rho}{\partial t^2} + \frac{\partial^2}{\partial x_i \partial t} (\rho v_i) = 0 \hspace{1cm} (2.5)
\]

\[
\frac{\partial}{\partial x_i} (2.3) => \frac{\partial^2}{\partial t \partial x_i} (\rho v_i) + c_o^2 \frac{\partial^2 \rho}{\partial x_i^2} = - \frac{\partial^2 T_{ij}}{\partial x_j \partial x_i} \hspace{1cm} (2.6)
\]

Subtracting 2.5 from 2.6 and dividing by the speed of sound squared gives Eq. 2.7. In these equations, \( x_i \) is the direction, \( \rho \) is the density, \( v_i \) is the velocity, and \( c_o \) is the speed of sound in uniform medium.

\[
\frac{1}{c_o^2} \frac{\partial^2 \rho}{\partial t^2} - \frac{\partial^2 \rho}{\partial x_i^2} = \frac{1}{c_o^2} \frac{\partial^2 T_{ij}}{\partial x_j \partial x_i} \hspace{1cm} (2.7)
\]

**2.2 Green’s Function**

The next step is to solve for density in Eq. 2.7. In Goldstein’s paper [2.3], he defines Eq. 2.8 as the Green’s function for the case where the region is all of space and the mean flow is zero. This is also known as the free-space Green’s function. Please note the Green’s function is in terms of \( G(y,x,t|x,\tau) \). Goldstein, shows how inserting an incoming wave
equation into Eq. 2.8 and performing the necessary integration with respect to $\tau$ produces Eq. 2.9. Eq. 2.9 can be applied to Eq. 2.7, assuming that any solid boundaries do not significantly affect the sound field and that mean flow is zero, to produce Eq. 2.10.

$$G^0 = \frac{1}{4\pi r} \delta \left( \tau - t + \frac{r}{c_o} \right) \quad (2.8)$$

where $\delta(t - \tau)$ is a delta function

$$4\pi \phi(x,t) = \int_{R=-\infty}^{t} \frac{A(y,\tau)}{r} \quad (2.9)$$

$$4\pi (\rho(x,t) - \rho_o) = \int \frac{1}{rc_o^2} \frac{\partial^2 T_y(y,\tau)}{\partial y_i \partial y_j} d\vec{y}$$

$$\rho(x,t) - \rho_o = \frac{1}{4\pi c_o^2} \int \frac{1}{r} \frac{\partial^2 T_o(y,\tau)}{\partial y_i \partial y_j} d\vec{y} \quad (2.10)$$

where $r = |\vec{x} - \vec{y}|$

$$\tau = t - \frac{r}{c_o}$$

Goldstein further shows that performing partial differentiation with respect to $y_i$ while holding both $t$ and $r$ constant, using the chain rule for partial differentiation and assuming $T_{ij}$ is smooth and decays faster than $1/y$ for large values of $y$, Eq. 2.10 can be simplified to Eq. 2.11.

$$\rho(x,t) - \rho_o = \frac{1}{4\pi c_o^2} \frac{\partial^2}{\partial x_i \partial x_j} \int \frac{T_y(y,t - \frac{r}{c_o})}{r} d\vec{y} \quad (2.11)$$
2.3 Moving Reference Plane

The next part of the equation development focuses on the moving reference plane [2.6]. From figure 2-1, the following relationships are developed. Vectors are denoted with an arrow in the equation, or bold within the text. Turbulence generates sound at $O_y$ in fixed axes or $O_\eta$ in moving axes at time $\tau = t - |\mathbf{x} - \mathbf{y}|/c_0$. The observer is at a distance of $r$ (defined by the coordinates $\mathbf{x}$ and $\mathbf{y}$). The sound from the aircraft propagates downward at the speed of sound ($c_o$) and reaches the observer at $\mathbf{x}$ at time $t$. The aircraft is moving with a velocity of $c_o M_\infty$ (speed of sound times the free stream Mach number). The following equations define the relationship between the fixed axis $\mathbf{y}$ and the moving axis $\eta$ as seen in figure 2-1 (Eq. 2.12) and define $\mathbf{r}$, which is the distance between the noise source and the observer in the fixed axis (Eq. 2.13).

$$\tilde{\mathbf{y}} = \tilde{\eta} + c_o \tilde{M}_\infty \tau \quad \text{where} \quad \tilde{M}_\infty = -M_\infty \hat{\mathbf{t}}$$
$$\tilde{\eta} = \tilde{y} - c_o \tilde{M}_\infty \tau$$

where

$$\eta_1 = y_1 + c_o M_\infty \tau = y_1 + c_o M_\infty t - M_\infty r_1$$
$$\eta_2 = y_2$$
$$\eta_3 = y_3$$

$$\tilde{\mathbf{r}} = \tilde{\mathbf{x}} - \tilde{\mathbf{y}}$$
$$r = |\tilde{\mathbf{x}} - \tilde{\mathbf{y}}| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2} \quad (2.13)$$

$$\frac{\partial r}{\partial x_1} = \frac{1}{2r} \left( x_1 - y_1 \right) = \frac{r_1}{r}$$

If $\varphi(\tilde{\eta}, \tau) = T_{y\tilde{y}}(\tilde{\mathbf{y}}, \tau)$ where Eq. 2.12 is met, then the following transformations can be made.
\[
\frac{\partial \tilde{\psi}}{\partial \tilde{x}} \frac{\partial \tilde{\psi}}{\partial \tilde{\eta}} \frac{\partial \tilde{\psi}}{\partial \tilde{\tau}} + \frac{\partial \tilde{\psi}}{\partial \tilde{\tau}} \frac{\partial \tilde{\psi}}{\partial \tilde{\tilde{x}}}
\]

\[
\frac{\partial \eta_1}{\partial x_1} = -\frac{r}{\rho} M_\infty \quad \text{while} \quad \frac{\partial \eta_2}{\partial x_2} = \frac{\partial \eta_3}{\partial x_3} = 0
\]

\[
\frac{\partial \tilde{\eta}}{\partial \tilde{x}} = \frac{\tilde{r}}{\rho} \tilde{M} \quad \text{where} \quad M_1 = -M_\infty, \quad M_2 = M_3 = 0
\]

\[
\frac{\partial \tilde{\tau}}{\partial \tilde{x}} = -\tilde{r}
\]

\[
\frac{\partial \tilde{\eta}}{\partial \tilde{\tilde{x}}} = c_\circ r
\]

\[
\frac{\partial \tilde{\psi}}{\partial \tilde{\eta}} = \frac{\partial \tilde{\psi}}{\partial \tilde{\eta}_r} \frac{\partial \tilde{\eta}_r}{\partial \tilde{\eta}} + \frac{\partial \tilde{\psi}}{\partial \tilde{\eta}_\eta} \frac{\partial \tilde{\eta}_\eta}{\partial \tilde{\eta}} + \frac{\partial \tilde{\psi}}{\partial \tilde{\eta}_\tau} \frac{\partial \tilde{\eta}_\tau}{\partial \tilde{\eta}}
\]

\[
\left. \frac{\partial \tilde{\eta}}{\partial \tilde{\eta}_r} \right| = 1
\]

\[
\left. \frac{\partial \tilde{\eta}}{\partial \tilde{\eta}_\eta} \right| = -\frac{1}{c_\circ} \frac{\partial [\tilde{x} - \tilde{y}]}{\partial \tilde{\eta}} = -\frac{1}{c_\circ} \frac{\partial \tilde{r}}{\partial \tilde{\eta}}
\]

\[
\left. \frac{\partial \tilde{\psi}}{\partial \tilde{\eta}} \right| = \frac{\partial \tilde{\psi}}{\partial \tilde{\eta}_r} - \frac{\partial \tilde{\psi}}{\partial \tilde{\eta}_\eta} \frac{1}{c_\circ} \frac{\partial \tilde{r}}{\partial \tilde{\eta}}
\]

(2.14)

Ffowces-Williams [2.4] states that to obtain \( \frac{\partial \varphi}{\partial \eta} \) with \( \tau \) held constant, it is necessary to vary \( y \).

\[
\left. \frac{\partial \tilde{\psi}}{\partial \tilde{\eta}} \right| = \frac{\partial \tilde{\psi}}{\partial \tilde{\eta}_r} - \frac{\partial \tilde{\psi}}{\partial \tilde{\eta}_\eta} \frac{1}{c_\circ} \frac{\partial \tilde{r}}{\partial \tilde{\eta}}
\]

If the \( \eta \) and \( y \) in Eq. 2.12 are differentiated, a solution for \( \varphi / \partial \eta \) can be calculated.


\[
\frac{\partial \tilde{y}}{\partial \eta} = 1 - \frac{\dot{M}}{\ddot{r}} \\
\frac{\partial \tilde{\eta}}{\partial \tilde{y}} = 1 - \frac{\dot{M}}{r} \Rightarrow \frac{\partial \tilde{y}}{\partial \eta} = \left(1 - \frac{\dot{M}}{r}\right)^{-1}
\]

\[
1 - \dot{M} \frac{\partial \tilde{r}}{\partial \eta} = \left(1 - \frac{\dot{M}}{r}\right)^{-1} = \left(\frac{r - \dot{M} \tilde{r}}{r - \dot{M} \tilde{r}}\right) = \frac{-\dot{M} \tilde{r}}{r - \dot{M} \tilde{r}}
\]

\[
\frac{\partial \tilde{r}}{\partial \eta} = -\frac{r}{r - \dot{M} \tilde{r}}
\]

(2.17)

Inserting Eq. 2.17 into 2.16, the following relationship is developed.

\[
\frac{\partial \tilde{\varphi}}{\partial \tilde{\eta}} = \frac{\partial \tilde{\varphi}}{\partial \tilde{\eta}} \bigg|_{r} + \frac{1}{c_{o}}\left(\frac{r}{r - \dot{M} \tilde{r}}\right) \frac{\partial \tilde{\varphi}}{\partial \tau_{\eta}}
\]

\[
\frac{\partial \tilde{\varphi}}{\partial \tilde{\eta}} = \frac{\partial \tilde{\varphi}}{\partial \tilde{\eta}} \bigg|_{r} - \frac{1}{c_{o}}\left(\frac{r}{r - \dot{M} \tilde{r}}\right) \frac{\partial \tilde{\varphi}}{\partial \tau_{\eta}}
\]

(2.18)

The following equations use 2.18 as the definition for the space derivative (\(\eta\)) in 2.14 and simplify the results to obtain 2.19.

\[
\frac{\partial \tilde{\varphi}}{\partial \tilde{x}} = \frac{\tilde{r}}{c_{o} \tilde{M}} \left(c_{o} \tilde{M} \frac{\partial \tilde{\varphi}}{\partial \tilde{\eta}} - \frac{1}{c_{o}}\left(\frac{r}{r - \dot{M} \tilde{r}}\right) \frac{\partial \tilde{\varphi}}{\partial \tau_{\eta}}\right) - \frac{\partial \tilde{\varphi}}{\partial \tau_{\eta}}
\]

\[
\frac{\partial \tilde{\varphi}}{\partial \tilde{x}} = \frac{\tilde{r}}{c_{o} r} \left(c_{o} \tilde{M} \frac{\partial \tilde{\varphi}}{\partial \tilde{\eta}} + \left(-\dot{M} \tilde{r}\right) \frac{\partial \tilde{\varphi}}{\partial \tau_{\eta}} + \left(-r + \dot{M} \tilde{r}\right) \frac{\partial \tilde{\varphi}}{\partial \tau_{\eta}}\right)
\]

\[
\frac{\partial \tilde{\varphi}}{\partial \tilde{x}} = \frac{\tilde{r}}{c_{o} r} \left(c_{o} \tilde{M} \frac{\partial \tilde{\varphi}}{\partial \tilde{\eta}} + \left(-\dot{M} \tilde{r} - r \dot{M}\right) \frac{\partial \tilde{\varphi}}{\partial \tau_{\eta}}\right)
\]

\[
\frac{\partial \tilde{\varphi}}{\partial \tilde{x}} = \frac{\tilde{r}}{c_{o} r} \left(c_{o} \tilde{M} \frac{\partial \tilde{\varphi}}{\partial \tilde{\eta}} - \left(\frac{r}{r - \dot{M} \tilde{r}}\right) \frac{\partial \tilde{\varphi}}{\partial \tau_{\eta}}\right)
\]

(2.19)
If $M$ is assumed to be an incompressible vector field, then $c_o \ddot{M} \frac{\partial}{\partial \eta} \ddot{q} = \frac{\partial}{\partial \eta} c_o \dot{M} \ddot{q}$ from the divergence theorem. When Eq. 2.19 is put into 2.11, the first term of 2.19 becomes a surface integral. If turbulence is assumed to be bounded in $y$ and $\eta$ then the first term equals zero and will be dropped from the equation going forward.

Applying Eq. 2.20 to 2.11, the second-order space derivative in $y$ is changed to a second-order time derivative.

\[
\rho(\bar{x}, t) - \rho_o = \frac{1}{4\pi c_o^2} \int \frac{\bar{r} r}{c_o^2 r^2 C^2} \frac{\partial^2}{\partial t^2} T_{ij}(\bar{y}, t - r/c_o) d\bar{y}
\]

where $\bar{C} = \left(1 - \bar{M} \frac{\bar{r}}{r}\right) = \left(1 + M^\infty \frac{r}{r}\right)$

Finally, to transform from the $y$ space field to the $\eta$ space field, the following changes occur.

\[
d\eta = \left|\frac{d\eta}{d\bar{y}}\right| d\bar{y}
\]

\[
\frac{d\bar{y}}{d\eta} = 1 - \bar{M} \frac{\bar{r}}{r} = C
\]

\[
d\bar{y} = C d\eta
\]

\[
d\bar{y} = \frac{d\eta}{C}
\]

so $\rho(x, t) - \rho_o = \frac{1}{4\pi c_o^4} \int \frac{\bar{r} r}{r^3 C^3} \frac{\partial^2}{\partial t^2} T_{ij}(\bar{y}, t - r/c_o) d\eta$
terms of density (Eq. 2.23a) where the thermodynamic equation of state of the fluid determines \( \beta \). If the compression-expansion process is assumed to be adiabatic, \( \beta \) is equal to one and Eq. 2.23a can be written as Eq. 2.23b. Finally Eq. 2.22 can be written as Eq. 2.24 for integrating over the volume \((V)\) of the domain in \( \eta \) space field.

\[
p - p_{\infty} = c_o^2 (\rho - \rho_{\infty})^\beta \tag{2.23a}
\]

\[
p - p_{\infty} = c_o^2 (\rho - \rho_{\infty}) \tag{2.23b}
\]

\[
p(x,t) = \frac{1}{4\pi c_o^2} \int \frac{\vec{r} \cdot \vec{r}}{r^3 C_3} \frac{\partial^2}{\partial t^2} \left[ T_{ij}(\vec{y},t-r/c_o) \right] dV \tag{2.24}
\]

In JET3D-UNS, the second derivative of \( T_{ij} \) is calculated using three time steps of \( T_{ij} \) (Eq. 2.4). The variables held in the unsteady turbulence models in PAB3D contain both the steady and the fluctuating components of \( T_{ij} \) which reduces the steps necessary to compute \( T_{ij} \). A simple central differencing scheme is used to obtain the second derivative with respect to time (Eq. 2.25). Chapter 7 will compare this approach with what is used in JET3D.

\[
\frac{\partial^2 T_{ij}}{\partial t^2} = \frac{T_{ij}^{n+1} - 2T_{ij}^n + T_{ij}^{n-1}}{(\Delta t)^2} \quad \text{where } \Delta t = \text{time step} \tag{2.25}
\]

### 2.4 Post Processing

The results from Eq. 2.24 are converted to the auto spectral density function using Eq. 2.26 where \( \hat{p}_n(f) \) is the Fourier transform of \( p(t) \) and the asterisk means the complex conjugate.
\[ G(f) = \frac{2}{T} \langle \hat{p}_n(f) \cdot \hat{p}^*_n(f) \rangle \]  

(2.26)

The fast Fourier transform procedure adapted from *Applied Numerical Methods for Engineers* [2.7] was then used. Finally, to convert the pressure levels to decibels (dB), Eq. 2.27 was used where \( p_{\text{ref}} = 20 \times 10^{-6} \text{ N/m}^2 = 20 \mu\text{Pa} \) for sound in gases.

\[ SPL = 10 \log \left( \frac{G(f)}{p_{\text{ref}}^2} \right) \]  

(2.27)

### 2.5 Corrections

Two types of corrections are applied to the data. Doppler convection factors are applied real-time within Jet3D-UNS to Eq. 2.20. Ffowcs-Williams [2.6] derived two different versions, one to handle the effects of the freestream velocity, the other one to handle the convection velocity. He found the corrections would be applied to the calculation of pressure as shown in Eq. 2.28.

\[ p(\tilde{x},t) = \frac{1}{4\pi c_\infty} \int \frac{\tilde{r} \tilde{r}}{r^3 c_\infty^{3/2} C^{5/2}} \frac{\partial^2}{\partial t^2} [T_0(\tilde{y},t-r/c_\infty)] dy \]  

(2.28)

The two corrections are defined in Eqs. 2.29 and 2.30 where \( \theta \) is the observer angle.

\[ C_\infty = |1 + M_\infty \cos \theta| \]  

(2.29)

\[ C = |1 - M_c \cos \theta| \]  

(2.30)

However, the second term, \( C \), was too simple of an approximation so both Ffowcs-Williams [2.6] and Ribner [2.8] changed the term to:
\[ C = \sqrt{(1 - M_c \cos \theta)^2 + \alpha^2 M_c^2} \]  

(2.31)

In order to compare with the results of JET3D, \( \alpha \) was set to match Hunter’s value of 0.58.

The convection Mach number, \( M_c \), was calculated from a quadratic fit to Davies’ measured convection velocity \( (u_c) \) [2.9] computed by Hunter [2.10] and included below as Eq. 2.32, where \( u_c \) is jet exit velocity, \( u_\infty \) is free-stream velocity, and \( u_1 \) is the local x component of velocity.

\[
\frac{u_c - u_\infty}{u_c - u_\infty} = 0.13 + 1.18 \left( \frac{u_1 - u_\infty}{u_c - u_\infty} \right) - 0.62 \left( \frac{u_c - u_\infty}{u_c - u_\infty} \right)^2
\]  

(2.32)

The second correction is for atmospheric absorption. Since this correction is applied to the sound pressure levels, it could not be done within the main Jet3D-UNS code. Consequently, the correction is made within the post-processor after the fast Fourier transform is completed. The model, developed from experimental data by Shields and Bass [2.11], is shown in Eq. 2.33 while the definitions of the variables used are shown in Eq. 2.34. If this correction is invoked, the sound pressure level, Eq. 2.27, is multiplied by \( 10^{-\frac{a}{10}} \) where \( a \) is defined by Eq. 2.33. The code to solve for this correction is included at the end of Appendix D.
where:

\[ f_{r,o} = \frac{P}{P_0} \left\{ 24 + 4.41e4 * h \left[ (0.05 + h) / (0.391 + h) \right] \right\} \]

\[ f_{r,N} = \frac{P}{P_0} \left( \frac{T}{T_0} \right)^{1/2} \left\{ 9 + 350 * h * \exp \left[ -6.142 \left( \frac{T}{T_0} \right)^{-1/2} \right] \right\} \]

\[ h = h_r * \frac{P_{sat}}{P} \]

\[ \log_{10} \left( \frac{P_{sat}}{P_0} \right) = 10.79586 \left[ 1 - \left( \frac{T_0}{T} \right) \right] - 5.028081 \log_{10} \left( \frac{T}{T_0} \right) \]

\[ + 1.50474e - 4 \left[ 1 - 10^{-8.29692(T_0/T)} \right] + 0.42873e - 3 \left[ 10^{-4.7695\left(1-\left(\frac{T}{T_0}\right)\right)} - 1 \right] - 2.2195983 \]

\[ a(f) = 8.868 \left( \frac{T}{T_0} \right)^{1/2} \left\{ \frac{f^2}{P} \right\} \left( \frac{P}{P_0} \right)^{1/2} \left\{ 1.84e-11 + 2.19e-4 \frac{P}{P_0} * \left( \frac{2239}{T} \right)^2 \exp(-2239 / T) \frac{f_{r,o}^2}{f_{r,o}} + \frac{f^2}{f_{r,o}} \right\} \]

\[ + 8.16e-4 \frac{P}{P_0} * \left( \frac{3352}{T} \right)^2 \exp(-3352 / T) \frac{f_{r,N}^2}{f_{r,N}} + \frac{f^2}{f_{r,N}} \]
Figure 2-1 Moving Axis Transformation [Adapted from Ref. 2.6]
Chapter 3: Computational Fluid Dynamics Software and Turbulence Models

3.1 Computational Fluid Dynamics Program

The NASA Langley Research Center’s computational fluid dynamics (CFD) code PAB3D was used for this study. It is a structured, multi-block, parallel, implicit, finite-volume solver for the three-dimensional Reynolds-Averaged Navier-Stokes (RANS) equations. The code has been updated through the years with a partial summary of its capabilities listed below [3.1]:

- 3-D RANS/Unsteady RANS (URANS)/Space Marching Scheme (SMS) upwind solver:
  - Multiple flow solvers (three factor, two factor, and diagonalized alternating-direction implicit (ADI))
  - Up to third order accurate in space and second order accurate in time

- Two-equation turbulence models with several options: [3.2, 3.3]
  - Compressibility correction for high speed flow
  - Near-wall correction, most commonly used correction is based on Launder and Sharma [3.1]
  - Temperature correction for high temperature jet flow
  - Up to third order accurate in space and second-order accurate in time

- Several algebraic Reynolds stress models [3.4]
- Tripping option for forced transition from laminar to turbulent flow
• Multi-scale-type (hybrid) turbulence models: URANS/LES and Partially Averaged Navier-Stokes (PANS) [3.5]
• Steady and unsteady flow boundary conditions
• Modular multi-block structured-mesh capability with user options selectable for each block:
  — Directional grid sequencing
  — Turbulence model options
  — Scheme choice including limiter and order
  — Viscous terms from simplified to fully coupled
  — Number of iterations ≥0 (effectively turns integration within a particular block on or off)
• Calculations of integrated forces, moments, and flux quantities
• Mixed Roe and van Leer upwind schemes for discretization of inviscid fluxes
• Multi-species and real gas simulation capability (with frozen chemistry) [3.6]

The governing equations for the RANS equations, in conservative form are shown in Eqs. 3.1-3.3. The two-equation (k-ε) turbulence model used to close the RANS equations is shown in Eqs. 3.4 and 3.5.

\[
\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0
\]  (3.1)
\[ \frac{\partial \rho u_i}{\partial t} + \frac{\partial (\rho u_i u_j + p \delta_{ij})}{\partial x_j} = \frac{\partial (\tau_{ij} - \rho u_i u_j)}{\partial x_j} \] (3.2)

\[ \frac{\partial \rho e_0}{\partial t} + \frac{\partial (\rho e_0 u_i + p u_i)}{\partial x_i} = \frac{\partial (\tau_{ij} - \rho u_i u_j)}{\partial x_j} - \rho (q_i + C_p \rho u_i \theta) + \frac{\partial}{\partial x_i} \left[ \rho (v_i + \frac{v_i}{\sigma_e}) \frac{\partial}{\partial x_j} \right] \] (3.3)

\[ \frac{\partial \rho k}{\partial t} + \frac{\partial (\rho u_i k)}{\partial x_i} = -\rho u_i \frac{\partial u_i}{\partial x_i} + \frac{\partial}{\partial x_i} \left[ \rho \left( v_i + \frac{c_k k^2}{\sigma_k e} \right) \frac{\partial}{\partial x_j} \right] - \rho \bar{e} \] (3.4)

\[ \frac{\partial \rho \varepsilon}{\partial t} + \frac{\partial (\rho u_i \varepsilon)}{\partial x_i} = -C_1 \rho u_i u_j \frac{\partial u_j}{\partial x_i} k + \frac{\partial}{\partial x_i} \left[ \rho \left( v_i + \frac{c_\mu k^2}{\sigma_\varepsilon e} \right) \frac{\partial \varepsilon}{\partial x_j} \right] - f_2 \bar{C}_{\varepsilon 2} \rho \varepsilon \left[ \varepsilon - 2v_i \left( \frac{\partial \sqrt{k}}{\partial n} \right)^2 \right] \] (3.5)

where \( C_\mu = 0.09, C_{\varepsilon 1} = 1.44, \sigma_k = 1.4, \sigma_\varepsilon = \sigma_\varepsilon = 1 \), and \( C_{\varepsilon 2} = C_{\varepsilon 2} = 1.92 \)

The turbulent stress components along with the linear contribution to stress are shown in Eq. 3.6 while the RANS turbulent viscosity is defined in Eq. 3.7a. \( f_\mu \) equals 1 unless a near wall correction is implemented (Eq. 3.7b).

\[ \bar{\rho u_i u_j} = \tau_{ij}^L + \tau_{ij}^{NL} \]

\[ \tau_{ij}^L = -2 \rho v_i S_{ij} + \frac{2}{3} \delta_{ij} \rho k \] (3.6)

where \( S_{ij} = \frac{1}{2} \left[ \frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right] - \frac{1}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \)

\[ v_i^{RANS} = f_\mu C_\mu \frac{k^2}{\varepsilon} \]

\[ v_i = v_i^{RANS}, \quad C_\mu = 0.09 \] (3.7a)
\[ f_\mu = \exp \left( \frac{-3.41}{\left(1 + \frac{R_T}{50}\right)^2} \right), \quad R_T = \frac{k^2}{\mu, \varepsilon} \quad (3.7b) \]

\[ f_2 = 1 - 0.3\exp(-R_T^3) \]

The RANS equations assume all of the turbulent energy is modeled through the turbulence transport equation. However, the other implementations such as PANS and LES resolve some of the turbulent energy while modeling the rest. RANS has been shown to over predict the eddy viscosity, which leads to excessive damping of any unsteady motion [3.1]. Turbulent models such as hybrid RANS/LES models [3.7] and the Partially Averaged Navier-Stokes (PANS) models [3.8] have been created to resolve this problem by resolving more of the turbulent energy versus modeling it all like RANS does [3.1]. PAB3D has recently included the PANS and a hybrid-RANS/LES turbulence model, which are discussed below.

### 3.2 PANS Turbulence Model

The Partially Averaged Navier-Stokes (PANS) turbulence model was created by Dr. Girimaji to meet the need for a turbulence model that did not require a significant computational effort, such as LES, yet increased the accuracy of the results when compared to RANS [3.9]. PANS solves for the unresolved kinetic energy, \( k_u \), and the unresolved dissipation, \( \varepsilon_u \). While, the \( k \) and \( k_u \) equation are identical, the following coefficients are changed in the \( \varepsilon \) equation (Eq. 3.5) to make the \( \varepsilon_u \) equation [3.8]:
\[
\overline{C_{\epsilon 2}} = \frac{f_k}{f_{\epsilon}} (C_{\epsilon 2} - C_{\epsilon 1}) + C_{\epsilon 1}
\]

\[
\overline{\sigma_k} = \frac{f_k^2}{f_{\epsilon}} \sigma_k \quad \text{and} \quad \overline{\sigma_{\epsilon}} = \frac{f_k^2}{f_{\epsilon}} \sigma_{\epsilon}
\]

where

\[
f_k = \frac{k_u}{k} \quad \text{and} \quad f_{\epsilon} = \frac{\epsilon_u}{\epsilon}
\]

The two coefficients, \( f_k \) and \( f_{\epsilon} \), are the control parameters for PANS such that \( 0 \leq f_k \leq f_{\epsilon} \leq 1 \), where 0 would be a direct numerical simulation (DNS) without any turbulence modeling (not possible at this time), and 1 would be RANS.

Abdol-Hamid and Girimaji created a two-stage approach to estimate the unresolved kinetic energy coefficient, \( f_k \) [3.9]. \( f_{\epsilon} \) is set to a constant value of 1. They assumed that the turbulent viscosity could be related to total kinetic energy, \( k \), as follows:

\[
v_t \approx \frac{k^2}{\epsilon} = \Delta^2 \frac{S}{k_u} \approx \Delta^2 \frac{\epsilon}{k_u}
\]

\[
\epsilon = \frac{k^{3/2}}{L_T} \quad k_u = f_k k
\]

\[
f_k^3 k^3 \approx \Delta^2 \frac{k^3}{L_T^2}, \quad \lambda = \frac{L_T}{\Delta}
\]

\[
f_k \approx C_h \left[ \frac{1}{\lambda} \right]^{7/3}, \quad L_T = \left( \frac{k^{3/2}}{\epsilon} \right)^{RANS}
\]

Here \( L_T \) is the characteristic turbulence length scale and obtained from the RANS solution. \( C_h \) is a model coefficient currently set to 1. The smallest resolved length scale in PANS is
related to $f_k$ and $L_T$. The mesh length-scale $\Delta = \max(\Delta_x, \Delta_y, \Delta_z)$, where the three deltas are the grid cell distances in the x, y, and z directions, respectively.

For this study, the two-stage procedure was used. The grid is first run to convergence using RANS. That solution is used to estimate the $f_k$ parameter for each grid cell using Eq. 3.11 for $f_k$. Each cell is then marked to be run using either PANS or RANS, depending on the size of the grid cell and flow conditions within the cell. The grid is then run unsteady using the PANS model for a higher accuracy, unsteady solution.

### 3.3 Hybrid RANS/LES Turbulence Model

PAB3D’s hybrid RANS/LES turbulence model, created by Nichols and Nelson [3.7], was implemented with Menter’s SST two-equation turbulence model [3.10] and is referred to as a multi-scale model. The method uses Eqs. 3.4 and 3.5. Eq. 3.12 defines the turbulent length scale while Eq. 3.13 defines the LES turbulent kinetic energy. $\Omega$ is the vorticity magnitude.

$$L_T = \max \left( 6 \sqrt{\frac{V^{\text{RANS}}}{\Omega}}, \left( \frac{k^{3/2}}{\varepsilon} \right)^{\text{RANS}} \right)$$  \hspace{1cm} (3.12)

$$k^{\text{LES}} = f_d k$$  \hspace{1cm} (3.13)

The blending function in Eq. 3.13 is shown in Eq. 3.14 with the additional functions being defined in Eqs. 3.15 and 3.16.

$$f_d = \left\{ 1 + \tanh[2\pi(A - 0.5)] \right\} / 2$$  \hspace{1cm} (3.14)
\[ \Lambda = \frac{1}{1 + \left( \frac{L}{\Delta} \right)^{4/3}} = \frac{1}{1 + \lambda^{4/3}} \quad (3.15) \]

\[ \Delta = \max(\Delta_x, \Delta_y, \Delta_z) \quad (3.16) \]

The eddy viscosity (3.17) is calculated using the previously defined functions.

\[
v_i = f_d v_{i}^{\text{RANS}} + (1 - f_d) v_{i}^{\text{LES}} \\
v_{i}^{\text{LES}} = \min(v_{i}^{\text{RANS}}, 0.084 \Delta \sqrt{k_{\text{LES}}})
\]

where \( v_{i}^{\text{LES}} = f_\mu \mu^{*} \frac{k^2}{\varepsilon} \), \( \mu^{*} = 0.09 \)

Since the method is a hybrid method, the transition between RANS and LES is a function of the local grid spacing and local turbulent length scale (predicted by the RANS model). Consequently, the model determines whether the turbulence scales needed to be resolved can be performed on the existing grid and will then transition to LES if appropriate.
Chapter 4: Implementation

4.1 Implementation of JET3D-UNS within PAB3D

Appendix A gives a brief outline of how to run and process files for JET3D-UNS. JET3D-UNS starts with reading in user inputs from the user.cont file that is fundamental to running PAB3D (Appendix C). The JET3D-UNS section checks to see if the program has been selected for use, whether it is a JET3D-UNS restart, what iteration to start doing acoustic measurements, the minimum (dmin) and maximum distance (dmax) from the grid to the noise observers, the non-dimensional time step, and the necessary inputs if the user wants to add the Doppler correction to the data. Finally, the input file includes the number of observers and their locations. From these inputs, several fundamental calculations are made and memory is allocated. The non-dimensional time step is converted to a dimensional time step (seconds) by multiplying it by a conversion factor (PAB3D is computed in SI units) and then divided by the speed of sound (m/s) in order to remove the length scale and end with units of seconds. Next the maximum retarded time is calculated. In order to save computational time, the code subtracts the user defined minimum distance from the calculated distance to the observer. Since the code performs this subtraction consistently throughout the domain, the code assumes the closest noise source will reach the observer immediately. The user-defined minimum distance is subtracted from the user-defined maximum distance, then based on the time step, the maximum retarded time is calculated. This is the number of iterations that the code needs to store, which represents the maximum amount of data required to account for all of the noise sources that might arrive at an
observer at the same point in time. Three arrays are then allocated \((dT_{ij}, dT_{ij}M, P_{rij})\). They are of the size of the number of observers \(\times\) maximum retarded time. The \(dT_{ij}\) array is calculated for each block and is used to store the 2\textsuperscript{nd} derivative of Lighthill’s stress tensor in Eq. 2.24. Since the code is designed to use MPI, the second array \((dT_{ij}M)\) was created as a holding array for passing data to the main node. The \(P_{rij}\) array stores the acoustic pressure calculated from Eq. 2.24 and is passed from iteration to iteration (unlike \(dT_{ij}\), described later).

The fundamental parts of the code are accessed next. After making sure that JET3D-UNS has been turned on by the user and that the iteration is equal to or greater then the user specified starting point, the code cycles through each block. First it calculates the pointer values as specified in other parts of the code. It then passes those values to the acoustic subroutine. There, the code starts storing the fundamental data (unless it is a restart).

JET3D-UNS primarily gets its data from a section in the code called \textit{trackpoint}. Originally this section was implemented so that users could specify a particular cell and obtain fundamental data (density, velocity, energy, etc.) about the flow. The JET3D-UNS code uses trackpoint to transfer Lighthill’s stress tensor \((T_{ij} = (p - c_s^2 \rho)\delta_{ij} - P_{ij} + \rho v_i v_j)\) after it is calculated directly in the unsteady turbulence model. Once the \(T_{ij}\) values are transmitted to Jet3D-UNS, they are saved in an array that holds the three time levels of data necessary to compute the second derivative in time for each component of the tensor (Eq. 2.25). This vector is updated each time step, and limiting the size to just the data that is needed for the immediate computation reduces computational overhead.
The subroutine then sets the $dT_{ij}$ arrays to zero. The code then cycles through each of the cells within the block, stores the *trackpoint* data in the $t_{ij}$temp vector, and calculates the location of the center of the cell. The code next cycles through the user-specified number of observers, and calculates the distance from the cell to each of the observers as well as the volume of the cell. Next the code calculates the Doppler correction that can be applied to the data. If the user specifies that the Doppler correction should not be used, the correction is set equal to 1.0, then the local retarded time ($rdt$), i.e., the time it takes for the noise in the cell to reach the observer, is calculated. As mentioned before, the minimum distance is subtracted from the distance to the observer, then the number of time steps it takes for the sound wave generated at the cell to reach the observer is calculated. The variable $rdt$ is set to an integer since it will be used as a pointer and is calculated as the nearest integer value of the expression (retarded time/time steps). At this point, the user-defined values for $d_{min}$ and $d_{max}$ are checked. If the calculated value for $rdt$ is less then zero, then the code gives an error and closes. In order to cycle through the arrays, the current time step is added to the retarded time, and divided by the maximum retarded time ($rt$), which gives us the pointer $rdtc$. Next the code calculates the constants that form part of the $2^{nd}$ derivative calculation (Eq. 2.24). Finally, the code marches through and calculates the 6 unique components of the $2^{nd}$ derivative of the Lighthill tensor and stores them in the appropriate location within the $dT_{ij}$ array. Once this is done, the cycle is repeated for each observer, then each cell within the block. Once the $dT_{ij}$ vector has been calculated for each cell and each observer, the values are transmitted to the master cell where they are added into the $P_{ij}$ tensor (acoustic pressure...
Eq. 2.24). After these calculations have been done for all blocks, the code checks to see if enough time has passed (retarded time) for all the noise sources to have had their opportunity to reach all of the observers. The code then outputs the iteration number, the number of iterations that JET3D-UNS has been running, and the pressure values at each observer location (Eq. 2.24).

Once the user inputs have been met and JET3D-UNS terminates, the post process takes the pressure versus time data and converts it to decibels versus frequency using a fast Fourier transform. Additionally, if the user chooses, the atmospheric correction is applied (Appendix D).

4.2 JET3D-UNS Approach to Retarded Time

Jet3D-UNS uses a novel approach to retarded time. Retarded time is the time it takes noise to reach the observer. Traditional acoustic models work in observer time. Observer time is the current time where the observer is located and all of the noise sources are at different times in past. Noises are calculated back in time and then reported.

Jet3D-UNS is driven by cell time. The code focuses on when the noise is produced and how long it will take to get to each of the user defined observers. Each noise source is calculated, and then the time to reach the observer is computed. Each observer location has a time line associated with it; the noise produced at the cell is put in the appropriate observer time line (at the time in the future the noise would reach the observer). This time is
designated as the retarded time. The change was necessary in order to compute the noise real-time without consuming large amounts of computer memory.
Chapter 5: Test Cases

5.1 Test Facility and Data Collection Information

The experimental study was conducted at the NASA Langley Research Center Low Speed Aeroacoustic Wing Tunnel (LSAWT) [5.1, 5.2]. It is a continuous flow, in-draft wind tunnel. The test section is covered with fiberglass wedges with a cut-off frequency of 250 Hz. The inner dimension of the test section (from tip to tip of the wedges) is 34 feet long by 17 feet high and 17 feet wide. Additionally, the wind tunnel nozzle and flow collectors were treated to minimize reverberations.

The following three paragraphs relate to information presented in reference 5.2. Installed within LSAWT, the Jet Engine Simulator (JES) can produce two streams to simulate engine nozzle flow [5.2]. The flow is straightened prior to entering the nozzle. The flows, fan and core, can be heated separately by two propane burners. They also have an electric pre-heater that permits low temperature operation and burner stability. The maximum air speed for each flow is 17 lbm/sec. Turbine meters are used to measure fuel flow while pressure and temperature rakes are located upstream of the nozzle contraction.

A 28-microphone array, located 11.54 feet from the model’s centerline, was used to collect the acoustic data. The layout of the test section and microphones are shown in Figure 5-1. Each microphone was 0.25 inches in diameter and had its grid cap removed. Calibration was done both prior and after the test using a piston phone and an electrostatic calibrator.

The experimental data used in this study were obtained with nozzle pylons at 90-degrees relative to the microphone array. The data was processed to the 1/3-octave bands and
corrections were applied for the microphone calibration, background noise, shear layer refraction, Doppler shift and atmospheric absorption [5.2].

5.2 Configuration Information and Test Conditions

Two configurations were used in this study [5.3]. The first configuration, labeled Configuration 1, consisted of a baseline round core and fan nozzle (Figure 5-2a). Figure 5-2b shows the round core and fan nozzle with a pylon and lower fan bifurcator strut, which was the second configuration tested (Configuration 2). The nozzle and pylon designs are not from a specific engine, but are typical of engines with a bypass ratio (BPR) of five. Figure 5-3 shows Configuration 1 installed on the JES in LSAWT.

The data used for comparison, shown in Table 5-1, represents takeoff conditions [5.3]. The data for the two configurations were obtained at the same cycle point. Consequently, the thrust for Configuration 1 was greater than the thrust for Configuration 2 since the pylon subtracts from the fan nozzle stream. As a result, the pylon reduces fan thrust and total thrust by approximately 6.8% and 6%, respectively, when compared to Configuration 1.

<table>
<thead>
<tr>
<th>Table 5-1 Test Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Nozzle</td>
</tr>
<tr>
<td>Fan Nozzle</td>
</tr>
<tr>
<td>Free stream</td>
</tr>
</tbody>
</table>
Figure 5-1 NASA Langley Low Speed Aeroacoustic Wind Tunnel (dimensions in inches) [from Ref. 5.2]

Figure 5-2 Round Core and Fan Nozzle Geometries Tested [from Ref. 5.3]
Figure 5-3 Configuration 1 Installed on the Jet Engine Simulator (JES) in the NASA Langley Low Speed Aeroacoustic Wind Tunnel [from Ref. 5.3]
Chapter 6: Computational Grids

All the cases were run on three-dimensional structured meshes that modeled the entire 360 degrees of the engine and surrounding freestream flow. The PAB3D code allows the grid to be reduced in size for running by skipping every other grid point or more.

Figure 6-1 shows the block structure of the circular computational grid for Configuration 1. Figure 6-2 is a Z-plane cut at the center of the grid while Figure 6-3 shows the actual grid cells. GEOLAB, a lab group at NASA Langley Research Center that specializes in grid generation, produced the grids. The grid was made up of 176 blocks. The coarse, medium, and fine grids had 884,528, 7,089,760, and 28,359,040 grid cells, respectively. All of the grids were run using 60 processors on the Columbia super-computer at NASA Ames Research Center.

Figures 6-4, 6-5 and 6-6 show the block structure for Configuration 2, the Z-plane cut and the actual grid cells, respectively. The grid for Configuration 2 had 96 blocks. The coarse, medium, and fine grids had 356,240, 2,776,320, and 11,105,280 grid cells, respectively.

Figures 6-7 and 6-8 show the convergence of the residual of the L-2 norm for the RANS solutions for Configurations 1 and 2, respectively. The graphs show three bumps, indicating convergence obtained on the coarse, medium, and fine mesh levels. It was essential to ensure that the initial RANS solution was properly converged since both turbulence models use that solution for a starting point and to estimate turbulence length scales.
Figure 6-9 shows the computational domain with respect to the observer locations. The observers are the black triangles along the semi-circle below the jet. They range from an observer angle of 52-degrees to 162-degrees.

Figure 6-1 Configuration 1 – Full Grid

Figure 6-2 Configuration 1 – Center Z-Cut of Grid Block Structure
Figure 6-3 Configuration 1 – Center Z-Cut of Grid Cells

Figure 6-4 Configuration 2 – Full Grid
Figure 6-8 Configuration 2 – Residual Convergence

Figure 6-9 Observer Locations Relative to Computational Domain (dimensions in feet)
Chapter 7: Lighthill Stress Tensor Component Study

7.1 $T_{ij}$ Calculations

In the past, the Lighthill stress tensor, $T_{ij}$ in Eqs. 2.4 and 7.1, was approximated as $T_{ij} \sim \rho v_i v_j$ [7.1 and 7.2]. Within JET3D, a two-point space-time correlation is used to obtain the mean-square pressure [Chapter 3 of 7.2]. Through this correlation [7.3], the mean square pressure only depends on the time delay and the original second derivative of time is converted to a derivative of the time delay.

In JET3D-UNS, all four of the components of $T_{ij}$ are stored within the advanced turbulence model (PANS or hybrid-RANS/LES). Both the fluctuating and the steady state components of the variables are calculated. Consequently, at each time-step and for each cell, $T_{ij}$ is calculated then stored in a 3-cell array so that a second derivative in respect to time can be computed (Eq. 2.25).

7.2 $T_{ij}$ Component Study

In order to determine whether the Lighthill stress tensor should be approximated in this work, as it has been in the past, Configuration 2 was run with the full stress tensor, and then with only a part of the tensor. Eq. 7.1 shows how the equation was broken down for this study.

\[
T_{ij} = (p - c_a^2 \rho)\delta_{ij} - P_{ij} + \rho v_i v_j \quad \text{Comp1, Comp2, Comp3} \tag{7.1}
\]
Component 1 of Eq. 7.1 represents the stresses due to fluctuating pressure and density. Component 2 represents the viscous stresses due to strain rate and dilatation while component 3 represents the stresses due to momentum flux. In a jet, the two terms of component 1 represent entropy fluctuations due to heat conduction. Lighthill dropped this term in the belief that these fluctuations are generally small for cold jets [7.1]. Component 2 was ignored since stresses due to the momentum flux, are significantly greater than the viscous stresses (Comp 3 >> Comp2).

Figure 7-1 shows pressure versus iteration (Eq. 2.24) for each of the three components at a specified location within the flow. The graph shows that component 1 is actually the largest contributor with component 2 being the smallest. The component 1 results could be due to computations being done on a hot jet. Lighthill’s work was based on a cold jet. Component 3, usually the only part that is calculated, does contribute to the solutions and should be used along with component 1. Figure 7-2 compares the frequency (Hz) versus sound pressure level (dB) for each component along with the results from all three components. This figure shows the results from the observer angle of 88-degrees. Component 1 follows closely with the results from components 1-3. However, it is interesting to note that component 3 compares closely with the final results and follows the same trend. This is due to the law of logs where a change of three decibels is a 50% reduction in influence on the final combined results (pressure). However, this graph shows that if only component 3 is available for calculation purposes, it will provide a reasonable result for the
sound pressure level (SPL). This part of the study was conducted using the Configuration 2 with the coarse grid.

Figures 7-3 and 7-4 show pressure (part of component 1) and density times velocity squared (component 3) at three progressive steps in time using the PANS turbulence model on the coarse grid. Please note that these are not consecutive since little change would be noticeable when plotted. Since the noise pressure calculation is obtained by taking the second derivative of the Lighthill stress tensor in respect to time, the results obtained from the program would use these three results (as discussed in section 4.2). All of the figures show that the unsteady flow is evident in the pressure and velocity fields. Although these figures do not provide quantitative results like Figures 7-1 and 7-2, they do confirm visually that these stresses contribute to the overall results.

All of the solutions shown in the rest of the paper use all three components of the stress tensor.
Figure 7-1 Comparison of Lighthill Stress Tensor Components Effect on Pressure

Figure 7-2 Comparison of Lighthill Stress Tensor Components Effect on Sound Pressure Levels
Figure 7-3 Comparison of Pressure Contours at Three Different Time Steps
Figure 7-4 Comparison of Density Times Velocity Squared Contours at Three Different Time Steps
Chapter 8: Data Processing

Processing the data output from PAB3D with JET3D-UNS requires a post-processor that performs a fast Fourier transform on the pressure versus time trace data stored for each observer location. Although data is available to be processed at all frequencies, the accuracy of the results depends on the time step and total run time (including restarts). The lowest frequency in which the results can be trusted is the reciprocal of time (1/time) where time is defined by the number of data points used for the FFT multiplied by the time step. The highest frequency (Nyquist cutoff) is 1/(2*Δt) where Δt is the time step. The time step used with PAB3D is a non-dimensional number. In order to avoid confusion, JET3D-UNS automatically calculates and outputs the actual time step (in seconds) to the jetuns.out output file. For example, most of the runs were conducted using a PAB3D non-dimensional time step of 0.03, which was equal to 2.17x10^{-6} seconds. If 4,096 samples were processed using the FFT, the lowest frequency that the code could predict would be 112 while the highest would be ~230,414. If only 2,048 samples were used, the lowest frequency would be 224, which is higher than that of the first experimental data point. Therefore all of the data was processed with at least 4,096 samples. The code is run much longer than the 4,096 samples in order to obtain multiple data sets. This action is similar to how the experimental data was processed where data is obtained over a certain time period several times, averaged to take out any short term disturbances, and then processed. Likewise, the CFD results are broken into multiple samples of the same size, then averaged and processed. The processing of the data is both the fast Fourier transform to convert from time history to frequency and the
merging of data into data bands. Data bands are predefined frequency regions that are used with the acoustic community for data processing. They are used in the post processor and are user defined inputs.

Studies were conducted to test the effect of time step and number of data points on the final results. All of the studies were conducted on the fine mesh level of Configuration 2 using the PANS turbulence model. The comparison graphs show the results for an observer angle of 88-degrees. Figure 8-1 compares two different non-dimensional time steps (0.03, 0.015) using the medium grid. Since a smaller time step was used but the same number of total points were processed, the results for a non-dimensional time step of 0.015 are not accurate below 224 Hz. Additionally, the data is spread out all the way to ~460,828 Hz providing less data for each frequency band. Consequently the results from running a smaller time step underpredicts the results from using a time step of 0.03 while causing a larger overhead cost (retarded time doubles since it takes twice as many steps for the sound to reach the observer). Figure 8-2 compares sound pressure level results from four different FFT sample sizes with a non-dimensional time step of 0.03 (2,048, 4,096, 8,192, 16,384) obtained on the finest mesh. Note that for the 2,048 sample size results, frequency results below 224 Hz are not reliable so they were left off the chart. This comparison shows that with more data, the sound pressure level increases. This could be due to three reasons, the first of which is that the same total number of data points is used. Therefore, if more data is used for the processing there is less for averaging the samples. Secondly, the lower frequency limit gets smaller with more data points. This pushes more of the data towards the lower-frequency
bands. Finally, since the data is banded, as more data is added together, there is going to be a natural creep upwards in the results. In order to test this theory, data for the four different cases were compared without being banded. The data sets did not show the upward movement and overall the data compared well. For best practice, matching the experimental procedure of obtaining and averaging hundred of data sets would provide the best results. However, it is not practical to replicate this procedure in the code. For the rest of this paper, 4,096 iterations were used since it both meets the upper and lower frequency requirements for matching experimental results and still provides ample data for averaging.

Figure 8-3 compares the results of additional internal time steps. When running unsteady simulations, the code does a number of sub-iterations within each physical time step to converge the unsteady form of the governing equations to a specified tolerance. The standard practice is to use four sub-iterations. However, to ensure that the number of sub-iterations was not affecting the final results, one case was run with eight sub-iterations. Figure 8-3 compares those results using the medium grid and shows minimal effect from doubling the number of sub-iterations. Since increasing the number of sub-iterations dramatically increases computational time, all of the remaining calculations utilized four sub-iterations.

Figure 8-4 compares the effect of obtaining results immediately from the unsteady results versus waiting for the unsteady solution to have reached the far boundary (approximately 5,000 iterations when using a PAB3D non-dimensional time step of 0.03) using the fine grid. The results show that there is minimal difference between the two results.
although the delay results in a slightly shallower region in the 1,000 to 5,000 Hz frequency band. Since the results were not significantly different and since nine data sets were used for averaging in the final results, data was taken as soon as the retarded time (time for noise to reach the observer) had passed.

Finally, although all of the results discussed next are from the fine grid, Figure 8-5 compares the results from the coarse, medium, and fine grids for Configuration 1. The results from the coarse mesh case show that this grid “overpredicts” the sound pressure levels and does not provide a smooth and reasonable noise signature. While the medium and fine mesh cases follow the same general trends, the fine mesh results provide significantly smoother results in the vast majority of the frequency ranges.
Figure 8-2 Comparison of Configuration 2 Results with Different Data Processing Sizes

Figure 8-3 Comparison of Configuration 2 Results with Different Numbers of Sub-iterations
Figure 8-4 Comparison of Configuration 2 Results with Delayed Data Analysis

Figure 8-5 Comparison of Configuration 1 Results with Different Grid Sizes
Chapter 9: Computational Results

Figures 9-1 and 9-2 show axial velocity cuts of the unsteady PANS computational solutions for Configurations 1 and 2, respectively. Both cases show that the well-defined circular definition of the jet flow has collapsed prior to ten jet-diameters downstream of the jet. Figures 9-3a and 9-3b show the centerline cuts for Configurations 1 and 2, respectively. These plots confirm the degradation of the clear path of the jet flow prior to ten jet-diameters downstream of the jet. They also confirm that around five diameters downstream of the jet, the flow becomes unsteady (as hypothesized in the introduction to this paper). Figure 9-4 compares experimental, PAB3D-RANS solution, PAB3D-PANS solution and PAB3D-hybrid-RANS/LES solution for Configuration 1. All of the PAB3D-RANS solutions contain a temperature correction as mentioned in section 3.1. The plots compare normalized total temperature. The PANS and hybrid-RANS/LES solutions have been time averaged. The plots show that the PANS solution tends to mix out the core fluid more rapidly than observed in the experiment or in the RANS solution. Although the hybrid-RANS/LES solutions follows the same trend as the PANS, the hybrid-RANS/LES solution maintains a better defined core flow compared to PANS.

Figures 9-5 compares centerline cuts for the PANS, RANS and hybrid-RANS/LES CFD while Figure 9-6 compares the RANS with the PANS and hybrid-RANS/LES time-averaged CFD solutions for Configuration 1. Again, the PANS solution jet is mixing much faster than the RANS cases, while the hybrid-RANS/LES solution is mixing at a slower rate than the PANS solution. The effect of this drop is seen in the noise predictions shown in
Figure 9-7. The 52, 88 and 121-degree observer position cases show that the inability of the code to match the experimental data is greatly pronounced for frequencies under 10,000 Hz. However, JET3D-UNS does a better job with the 152-degree case. Figure 9-8, which shows the results for Configuration 2, confirms the code’s inability to predict the noise levels as well as or better than JET3D at most observer angles. For both Figures 9-7 and 9-8, the data for experimental and JET3D was provided by Dr. Craig Hunter.

Figure 9-9 shows a noise production histogram for configuration 2 produced by JET3D. Figures 9-10 and 9-11 show subsets of the data with the X/D=5 location clearly marked. Please note Figures 9-9 through 9-13 were provided through private communication with Dr. Hunter. The main area of noise production for each frequency is the peak of the curve. This is due to the rules of logarithms, where a three dB reduction leads to a 50% reduction in influence on the final combined results. These three figures show that the region of higher-frequency noise production is closer to the engine exit while the low frequency noise is produced much further downstream where the PANS and hybrid-RANS/LES jet flows falter compared to the RANS solution (Figures 9-5 and 9-6). Figures 9-5 and 9-6 also show that the hybrid-RANS/LES turbulence model predicts the jet flow longer and therefore, does a better job with the lower frequency predictions (Figure 9-7a and 9-7b).

Figure 9-12 uses results from JET3D to graphically show the noise sources throughout the flow. This is a great visualization, which shows that the noise is produced from the shear layers caused by the interaction of the jet flow with the free stream. Unfortunately, since all of the noise data is not stored for each iteration due to memory
constraints, these noise graphs (Figures 9-9 through 9-12) cannot be made with the JET3D-UNS data. However, in Figures 9-13 and 9-14, the total kinetic energy is plotted for the RANS (input for JET3D) and the PANS (input for JET3D-UNS) results. These show that the PANS turbulence model does not capture the kinetic energy around the shear layers as well as the RANS solution. When compared to Figure 9-12, the shear layers are areas of high noise production.

As discussed in sections 3.2 and 3.3, the PANS and hybrid-RANS/LES turbulence models were implemented in a hybrid method, i.e., not all of the cells within a grid are run using the more advanced turbulence model. Figure 9-15 plots the file used by PAB3D to determine which cells use RANS (1.0) and which use PANS (less than 1.0) for Configuration 1. The freestream flow and boundary layers are computed using RANS while the jet flow and shear layer interactions are mainly processed using PANS.
Figure 9-1 U -Velocity Cuts from the PANS Solution Downstream of the Configuration 1 Jet
Figure 9-2 U-Velocity Cuts from the PANS Solution Downstream of the Configuration 2 Jet
a) Configuration 1

b) Configuration 2

Figure 9-3 Centerline Streamwise Velocity Profile from the PANS Solution
Figure 9-4 Normalized total temperature cross sections at X/D=2, 5, 10, 17 for Configuration 1 Jet [Adapted from Ref. 9.1]
Figure 9-5 Centerline Normalized Unsteady Solution Total Temperature Profile for the Configuration 1 Jet
Figure 9-6 Centerline Normalized Time-Averaged Solution Total Temperature Profile for the Configuration 1 Jet
Figure 9-7 Predicted Noise Levels for Configuration 1
Figure 9-8 Predicted Noise Levels for Configuration 2
Figure 9-9 Noise Production Histogram Configuration 2

Figure 9-10 Noise Production Histogram Configuration 2 – Low Frequencies
Figure 9-11 Noise Production Histogram Configuration 2 – High Frequencies

Figure 9-12 Noise Source Plot Configuration 2 JET3D
Figure 9-13 Turbulent Kinetic Energy Plot Configuration 2 JET3D
Figure 9-14 Turbulent Kinetic Energy Plot Configuration 2 JET3D-UNS
Figure 9-15 Application of Turbulence Model for Configuration 1 PANS CFD Solution
Chapter 10: Concluding Remarks and Future Work

The primary conclusion of this investigation is that JET3D-UNS produces results inferior to the results from the baseline JET3D RANS results. Although the “bump” discussed in the introduction is gone, a valley now exists in its place. Although it is disheartening that the results from the unsteady solutions did not obtain the desired improvement in prediction capability, there are several avenues to investigate before this method is ruled out. Figures 9-9 through 9-11, along with the plots of the CFD flow results, show that the shear flow closer to the engine exit produced better matching data for the higher frequency ranges. For both Configuration 1 and Configuration 2, JET3D-UNS does a good job of predicting the noise levels for frequencies above 10,000Hz.

When comparing axial velocity and normalized temperature, it is easy to see that the unsteady case does not adequately predict the jet structure far downstream of its inception point. The RANS case does a better job of matching the experimental data. The noise production plots from JET3D shows that the jet structure far downstream of its inception point is important to the noise prediction of the low frequencies. Additionally, the noise source plots confirm that the shear layer is important to the noise prediction and further explains the under predictions of JET3D-UNS.

As shown, two different configurations with three different grid refinements (coarse [884,528 mesh points], medium [7,089,760 mesh points], and fine [28,359,040 mesh points]), along with numerous processing results, produced matching noise predictions.
As discussed previously, a majority of the implementations of the Lighthill Stress tensor simplify the equation by dropping terms in the tensor (Eq. 2.4). Early on in the study, runs were conducted with only one component turned on at a time since all four of the components of the tensor’s data were readily available within the turbulence model. The results showed that two of the components of the Lighthill tensor contributed to the pressure fluctuation.

As for future work, the NASA Subsonic Fixed Wing project recently awarded contracts to numerous universities to further investigate and develop unsteady turbulence modeling for CFD. As turbulence modeling continues to progress, JET3D-UNS will be tested with the new turbulence models as they become available within PAB3D. In the mean time, now that the fundamentals of the calculations and their implementation is working, the goal is to port JET3D-UNS into another CFD code at the NASA Langley Research Center that has been recently updated with newer turbulence models in addition to a different implementation of the hybrid-RANS/LES model. Since both turbulence models used within PAB3D are hybrid in their application, it would be beneficial to run a different version of hybrid-RANS/LES to see if the implementation is affecting the results. The hope would be that the unsteady solution from a different turbulence model would be able to model the jet core properly.
Chapter 1


Chapter 2


2.4 Ffowcs Williams, J.E. “Some Thoughts on the Effect of Aircraft Motion and Eddy Convection on the Noise from Air Jets.” University of Southampton USAA report 155, 1960.


Chapter 3


Chapter 5

5.1 “Langley Research Center’s The Low Speed Aeroacoustic Wind Tunnel (LSAWT) Fact Sheet.” NP-2011-09-415-LaRC, 2011.


Chapter 7


Chapter 9

APPENDIX A – JET3D-UNS OUTLINE

The below listing is a brief Jet3D-UNS outline:

Grid Preprocessor
Computes minimum and maximum distances between grid and observers
End Running Grid Preprocessor

Run PAB3D with JET3D-UNS built in
Reads JET3D-UNS inputs – check to see if JetUNS option is turned on
Define Constants
Calculate Max Retarded Time
Allocate Tensors Based on Need
Open Files for Output
Write Header for Output Files
Broadcast Constants and Tensors to other Processors
Iteration Loop
   Continue only if iteration # is equal or greater then user inputted starting iteration #
   Block Loop
      Cell Loop
         Calculate time step
         First Two Iterations:
            Fill in Tij tensor (3 time steps needed for 2nd derivative)
         Third Iteration on:
            Get current Tij tensor data
            Compute constants: distances, Doppler correction constants
            Calculate how long it would take for noise to reach observer
            Calculate and store cell’s Prij Component in temporary tensor
            Send data to Master Processor and add to Master Prij tensor
         End Cell Loop
      Calculate Current Time Step
      Output Current Time Step Data
      Reset Current Time Step Tensor Location
   End Block Loop
End Iteration Loop
End Running PAB3D with JET3D-UNS built in

Post Processor
User Inputs Data Set Size and Averaging Information
User Input Atmospheric Correction Option
Pij versus Time data converted to Decibels versus Frequency
End Running Post Processor
Below is the Fortran file for the grid preprocessor that calculates maximum and minimum distance to the observer so that maximum retarded time can be calculated properly.
program pregrid

c Grid Information Code
惚 Melissa Carter - Ph.D. & PAA work
惚 NASA Langley Research Center

c Include parameters and define variables
c This version is pre-processor for JET3D-UNS inside PAB3D
惚 Reads information out of user.cont instead of separate file
惚 Updated to new version of code - version 8

integer n,nobs,nblx,i,j,k,ii,icm,jcm,kcm,i,j,k,l,m,nr
integer ijk,l,ijkl,iijkl,ijklkl,ijklkl,l,m,nr,nm,km,nc,zt
real*8 ysc(3),yv(3,4),dmax,dmin,d,dt,rj
character*16 unf,for
character*40 gridfile,tester,tester1,tester2,tester3,skip
real*4,dimension(:,1),allocatable :: yN
real*8,dimension(1,:),allocatable :: obs,oby,obz
integer,dimension(1,:),allocatable :: idm,jdm,kdm
for='formatted'
unf='unformatted'

99 format (a)
open (unit=55, file='tpab3d.cont',status='old')
write(6,*),'Reading in Grid File Name'
read(55,*),gridfile

**** Read in Grid Units - rj from tpab3d.cont file *****

tester1=' Grid description'
tester2=' Grid description -- written with GridIron'
tester3='# created by GridIron 1.5.1 -- written with GridIron'

do j=1,10000
read(55,99,end=112) tester3
if(tester3.eq.tester.or.tester3.eq.tester1.or.
tester3.eq.tester2) goto 113
enddo

112 write(6,*), 'Could not find Grid Description Information'
write(6,*), 'at the end of tpab3d.cont file.'
write(6,*), 'You need to add the following to tpab3d.cont'
write(6,*), ' Grid description -- written with GridIron' or"
write(6,*), ' Grid description'" STOP

113 write(6,*), 'Found Grid Description'
read(55,*)
read(55,*) rj,dt
close(55)
c***** Read Control File *****
  open (unit=56, file='user.cont', status='old')
  tester='Begin JET3D-UNS Cont'
  do j=1,10000
     read(56,99,end=114) testerr1
     if(testerr1.eq.'testerr1') goto 115
  enddo
114  write(6,*)'Could not find JET3D-UNS Information'
  write(6,*)' at the end of user.cont file.'
  write(6,*)' You need to add the following to user.cont'
  write(6,*)'"Begin JET3D-UNS Cont"'
  stop
115  write(6,*)'Found JET3D-UNS Information'
  read(56,*) rt
  if (rt.ne.1) then
    write(6,*)'JET3D-UNS not turned on, must be set to 1'
    stop
  endif
  read (56,*)
  read (56,*)
  read (56,*)
  read (56,*)
  read (56,*)
  read (56,*) nobs
  write (6,*)'Total Number of Observers: ',nobs
  allocate(obx(nobs), oby(nobs), obz(nobs))
  read (56,*)
  do n=1,nobs
     read (56,*) skip,skip,obx(n),oby(n),obz(n)
  enddo
  write (6,*)
  write (6,*)'Done reading Control file'
  close(56)

c*** Assume observer locations in meters already (Suggested by Craig)

  c***** Read Grid File *****
  open (unit=14, file='gridfile', convert='big_endian',
       form='unf', status='old')
  read (14) nblok
  write (6,*)'There are ', nblok, ' blocks in the grid.'
  allocate(idm(nblok), jdm(nblok), kdm(nblok))
  read (14) (idm(n), jdm(n), kdm(n), n=1, nblok)
  write (6,*)'Done reading in block sizes'
  ijk=0
dmax=0
dmin=10000000
  do n=1,nblok
if (idm(n)*jdm(n)*kdm(n).gt.ijkm) then
  ijkm=idm(n)*jdm(n)*kdm(n)
endif
enddo

allocate(yN(3,ijkm))

***** Do loop through each block *****

do n=1,nblk
  inm=idm(n)
  jnm=jdm(n)
  knm=kdm(n)
  icm=idm(n)-1
  jcm=jdm(n)-1
  kcm=kdm(n)-1
  ijk=inm*jnm*knm

  write (6,*,'(Reading in block', n

  read (14, yN(1,ijk), ijk=1, ijkm),
  & (yN(2,ijk), ijk=1, ijkm),
  & (yN(3,ijk), ijk=1, ijkm)
  do ijk=1,ijkm
    yN(1,ijk)=yN(1,ijk)*rt
    yN(2,ijk)=yN(2,ijk)*rt
    yN(3,ijk)=yN(3,ijk)*rt
  enddo

***** Use Craig's method for reading in cell indices *****
***** k,j,i order dictated by Tecplot plotting *****

  do k=1, kcm
    do j=1, jcm
      do i=1, icm

        ijk=i+(i-1)+(j-1)*inm+(k-1)*inm*jnm
        lijk=i+(i-1)+(j-1)*inm+(k-1)*inm*jnm
        lijk=1+(i-1)+(j-1)*inm+(k-1)*inm*jnm
        lijk=1+(i-1)+(j-1)*inm+(k-1)*inm*jnm
        lijk=1+(i-1)+(j-1)*inm+(k-1)*inm*jnm
        lijk=1+(i-1)+(j-1)*inm+(k-1)*inm*jnm

        c ***** Calculate Cell Centers and Distance to Observers *****

        do ii=1,3
          ycc(ii)=(yN(ii,li,ijk)+yN(ii,li,ijk))/2
          & yN(ii,li,ijk)+yN(ii,li,ijk)
          & yN(ii,li,ijk)+yN(ii,li,ijk)
          & yN(ii,li,ijk)+yN(ii,li,ijk))
        enddo
d=gt(ycc(1)-obx(ii))**2+(ycc(2)-oby(ii))**2
  & +(ycc(3)-obz(ii))**2
  dmax=max(dmax,d)
  dmin=min(dmin,d)
enddo

80
enddo
enddo
enddo
enddo
close(14)

write (6,*) 'Put this in user.cont in JETID-UNS cont section:'
write (6,*) ' Minimum Distance (dmin) is ',dmin
write (6,*) ' Maximum Distance (dmax) is ',dmax
end
APPENDIX C – USER.CONT JET3D-UNS INPUT

Below is the control section necessary to run JET3D-UNS. It is added to the user.cont file.

'Begin JET3D-UNS Cont'
 1  *** 1 JET3D on 0 JET3D off
 0  *** 1 Restart, 0 Fresh run
*** Mint, dmin, dmax, dt ***
113001 8.82925 16.079258 0.015
*****
 1  *** 1 for Doppler corrections, 0 for none
457.2 0.28  *** Exit velocity of Jet (m/s), freestream Mach number
*** Number of observers ***
28
**** Locations of the observers ***
 1 52  -6.905 0.000 -10.024  ****number, obs name, x, y, z location (m)
 1 55  -6.351 0.000 -10.440
 1 58  -5.727 0.000 -10.856
 1 62  -5.032 0.000 -11.261
 1 65  -4.268 0.000 -11.642
 1 69  -3.437 0.000 -11.987
 1 74  -2.540 0.000 -12.286
 1 78  -1.581 0.000 -12.527
 1 83  -0.56 0.000 -12.699
 1 88  0.510 0.000 -12.790
 1 93  1.629 0.000 -12.789
 1 98  2.788 0.000 -12.684
 1 103 3.978 0.000 -12.464
 1 109 5.191 0.000 -12.116
 1 115 6.410 0.000 -11.628
 1 121 7.622 0.000 -10.990
 1 124 8.219 0.000 -10.611
 1 127 8.808 0.000 -10.189
 1 131 9.383 0.000 -9.724
 1 134 9.942 0.000 -9.216
 1 137 10.480 0.000 -8.665
 1 141 10.993 0.000 -8.072
 1 144 11.475 0.000 -7.439
 1 148 11.919 0.000 -6.774
 1 152 12.319 0.000 -6.087
| 1  | 155 | 12.669 | 0.000 | -5.390 |
| 2  | 158 | 12.962 | 0.000 | -4.706 |
| 3  | 162 | 13.201 | 0.000 | -4.051 |

'End JET3D-UNS Cont'
APPENDIX D – JET3D-UNS POST PROCESSOR

Below is the post-processor for JET3D-UNS that converts pressure versus time to decibels versus frequency.
program jetunfft
C===============================================================================================================
c Fast Fourier Transfer for Acoustic Data from JET3D-UNS Results
c Melissa Carter - Ph.D. & PAA work
c NASA Langley Research Center
C===============================================================================================================
c2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901
c===============================================================================================================
c Bands Data as Mike Marcolin suggested.
c Has Jeremy Pinier Correction
c Does multiple sets of data
c Does multiple observer locations
c Does atmospheric absorption correction
C===============================================================================================================
integer d, n, i, p, k, m, r, temp, temp2, mult, nband, nobs, abscor
character*10 skip2
character*30 title, title2
real*8 theta, pl, skip, dt, fmax, r1, r2, r3, rstar
real*8 tinf,phin, rh, cx, cy, cz, air, aircor, rr, centf
complex*16 u, w, ji, t
complex*16,dimension(:,:),allocatable :: z
real*8,dimension(:,:),allocatable :: band, bmin, bmax, fk
real*8,dimension(:,:),allocatable :: obsx, obsy, obsz
real*8,dimension(:,:),allocatable :: a, hold, so1n, value, decib
integer,dimension(:,:),allocatable :: name

pi=3.14159265358979
j=(0,1)

open(unit=35, file='user.cont', status='old')
open(unit=56, file='jetuns.inpt', status='old')
open(unit=55, file='jetuns.out', status='old')

read(55,*), skip2, skip, skip2, dt
read(55,*), nobs, n, mult
read(55,*), atmcor

if (atmcor.eq.1.or.atmcor.eq.2) then
    read(55,*), tinf,phin, rh
    read(55,*), cx, cy, cz
    tinf=tinf/1.8
    phin=phin*6.895e+3
else
    read(55,*)
    read(55,*)
    read(55,*)

85
read(55,*)  
endif

allocate (name(nobs), obsx(nobs), obsy(nobs), obsz(nobs))

99 format (a)
title2='"Begin JET3D-UNS Cont"'
rewind(35)
do i=1,10000
  read(35,99,end=1000) title
  if(title.eq.title2) goto 1001
endo

1000 write(6,*) 'Could not find JET3D-UNS Information in user.cont'
  stop

1001 continue
  read(35,*)
  read(35,*)
  read(35,*)
  read(35,*)
  read(35,*)
  read(35,*)
  read(35,*)
  do i=1,nobs
    do j=1,2
      read(35,*) skip, name(i), obsx(i), obsy(i), obsz(i)
    enddo

if (mult.eq.0) mult=1
  i=2
  do while (i.le.n)
    k=1
    i=i*2
  enddo

  if (n.ne.k) then
    write(6,*) 'n is the number of points used for each FFT and must be a multiple of 2. Will use the k value'
    write(6,*) 'n, k', n, k
    n=k
  endif

write(6,*), 'Initial number of data points', n, 'Number of data sets', mult
temp=n*mult
allocate(z(nobs,k),a(nobs,k),soln(nobs,k),hold(nobs,temp))
read(55,*)
write(6,*), 'Total Number of Data Points',temp

  do i=1,temp

read(55,*), skip, skip, (hold(d,1),d=1,nobs)
endo

do i=1,n
  do d=1,nobs
    s1n(d,i)=0
  enddo
endo
c Done Reading in Data, now reading in octave band information
read(56,*),
read(56,*),
read(56,*), nbands
read(56,*),
allocate(band(nbands),value(nobs,nbands),bmin(nbands),bmax(nbands),
  decib(nobs,nbands))

do i=1,nbands
  read(56,*), skip, band(i), bmin(i), bmax(i)
endo
open(unit=35, file='fft.out', status='unknown')

do p=1,mult
  temp1=(p-1)*n+1
  temp2=temp1+n-1
  temp3=1
  do i=temp1, temp3
    do d=1, nobs
      z(d,temp2)=hold(d,i)
    enddo
    temp2=temp2+1
  enddo

do d=1, nobs
  theta=-2*pi/n
  r=n/2
  i=1
  do while (i.lt.n)
    w=cos(i*theta)+j*sin(i*theta)
    k=1
    do while (k.le.n)
      u=1
      do m=0, r-1
        t=z(d,k+m)-z(d,k+m+r)
        z(d,k+m)=z(d,k+m)+z(d,k+m+r)
        z(d,k+m+r)=t*u
        u=u*w
      enddo
      k=k+2*r
    enddo
  endwhile
  i=i+1
enddo
i=i*2
r=r/2
enddo
do i=1,n
r=1-1
k=0
m=1
do while (m.lt.n)
k=2*k+MOD(r,2)
r=r/2
m=2*m
enddo
if (k.ge.i) then
t=z(d,i)
z(d,i)=z(d,k+1)
z(d,k+1)=t
endif
enddo
do i=1,n
a(d,i)=2*(z(d,i)*CONJG(z(d,i)))/(n/dt)
if (i.eq.1.or.i.eq.n/2+1) then
a(d,i)=a(d,i)/2
endif
enddo
do i=1,n
soln(d,i)=soln(d,i)+a(d,i)
enddo
dndo
do i=1,n
r=obsx(d)-cx
r2=obsy(d)-cy
r3=obsz(d)-cz
rr=sqrt(r1*r1+r2*r2+r3*r3)
rstar=(-0.28*r1+sqrt(0.28*0.28*r1*r1+(1-0.28*0.28)*rr*rr))/(1-0.28*0.28)
r1=r1-0.28*rstar
rr=sqrt(r1*r1+r2*r2+r3*r3)
soln(d,i)=soln(d,i)/mult
soln(d,i)=soln(d,i)/mult/((1+0.28*r1/rr)**3)
if (i.eq.1) write (*,*) d, 1/((1+0.28*r1/rr)**3)
enddo
dndo
r=n/2+1
allocate(fk(r+1))
fmax=1/(2*dt)
write(*,*) fmax
do i=1,r+1
   fk(i)=(i-1)*(fmax/n)
   write(*,*) i,fk(i),soln(1,i)
enddo

do i=1,nband
do d=1,nobs
   value(d,i)=0.0
enddo
enddo

do d=1, nobs
do i=2, r
   m=1
   if (m.gt.nband) goto 222
   if (fk(i-1).ge.bmin(m)) then
      if (fk(i-1).ge.bmax(m)) then
         m=m+1
         goto 333
      else
         value(d,m)=value(d,m)+(bmax(m)-fk(i-1))
         /fk(2)*soln(d,i)
         m=m+1
         goto 333
      endif
   elseif (fk(i).le.bmax(m)) then
      value(d,m)=value(d,m)+soln(d,i)
   endif
   endif
endif
endif
endif
endif
   endif
endif
enddo

222 enddo
enddo

if (atmcor.eq.1.or.atmcor.eq.2) then
do d=1,nobs
   rr=sqrt((obsx(d)-cx)**2+(obsy(d)-cy)**2+(obsz(d)-cz)**2)
do i=1,nband
   centf=band(i)
}
if (atmcor.eq.1) then
    call airab(pinf,tinf,rh,cenf,air)
else if (atmcor.eq.2) then
    call absorb(pinf,tinf,rh,cenf,air)
endif
aircor=10.**(air*rr/10.)
value(d,i)=value(d,i)*aircor
endif
do d=1, nobs
do i=1, nbnd
    if (value(d,i).gt.0) then
        decib(d,i)=10*log10(value(d,i)/0.00002/0.00002)
    else
        decib(d,i)=0.0
    endif
enddo
enddo
write(35,*), 'Band Information'
do d=1, nobs
    write(35,*), 'Observer', name(d)
do i=1, nbnd
    write(35,*), band(i), value(d,i), decib(d,i)
enddo
write(6,*), 'exited writing loop'
close(56)
close(55)
close(35)
end

 subroutine airab(p,t,rh,f,air)
c*** Subroutine to compute atmospheric absorption based on ***
c*** airab subroutine by Shields & Bass NASA CR-2760 pg 200 ***
real*8 p, t, rh, f, air, p0, t0, t01, Ps, h, fro, frn, alpha
p0=101300.0
 t0=293.15
 t01=273.16
Ps=10.79586*(1.0-t01/t)-5.02808*log10(t/t01)+1.50474E-4*5
   .(1.0-decibel**(-8.29692*((t/t01)-1.0)))+0.42873E-3
   .*(decibel**4.76955*(1.0-(t01/t)))-1.0)-2.2195983
Ps=10**Ps
h=Ps*(p0/p)*rh
fro=(p/p0)*2.40+h*4.41E+4*(0.05+h)/(0.391+h)
frn=(p/p0)*sqrt(t0/t)*(9.0+350.0*h
   .  *exp(-6.142*((t0/t)**(1./3.)-1.)))
alpha=1.84E-11+2.1913E-4*(t0/t)*p0*(2239.1/t)**2*
   .  exp(-2239.1/t)/(fo*(f**2./frn))
alpha=alpha+8.1619e-4*(t0/t)*(p/p0)*(3352.0/t)**2*
   .  exp(-3352.0/t)/(frn+(f**2./frn))
alpha=alpha*sqrt(t/t0)*(p0/p)**f**2
air=alpha*8.6860
return
end

subroutine absorb(p,t,rh,f,air)
c*** Calculates atmospheric absorption losses for air according to
  ANSI Standard 51.26-1978. The absorption loss ALPHA (dB/meter)
  is computed for a given frequency (Hz) as a function of
  pressure (Pa.), temperature (degK), and relative humidity(%)
  This is what is used in the NASA LaRC tunnel and is from
  Dr. Craig Hunter's JET3D acoustics noise prediction program

real*8 p, t, rh, f, air, t01, t0, p0, cof, pr
real*8 ahm, fro, fn2, cof1a, cof1b, cof1c, cof1d

  t01=273.16
  t0=293.15
  p0=101325.0

  c*** Calculate absolute humidity (ahm)
  cof=10.79586*(1.0-(t01/t))-5.02880*log10(t/t01)
  cof=cof+(1.50474e-4)*(1.0-10.**(-8.29692*((t/t01)-1.0)))
  cof=cof+(0.42873e-3)*(-1.0+10.**((4.76955*(1.0-(t01/t))))
  pr=10.0**(cof)
  ahm=rh**pr/(p/p0)

  c*** Calculate oxygen (fro) and nitrogen (fn2) relaxation freqs.
  fro=(p/p0)**((24.0+44100.0*ahm*)((0.05+ahm)/(0.391+ahm)))
  cof=9.0+350.0*ahm*exp(-6.142*((t/t0)**(1.0/3.0))-1.))
  fn2=cof*(p/p0)*sqrt(t/t0)
  cof1a=0.1068*exp(-3352.0/t)
  cof1b=0.01278*exp(-2239.1/t)
  cof1c=(t/t0)**-2.5
  cof1d=(1.84E-11)*(p0/p)**sqrt(t/t0)

  c*** Calculate absorption coefficient (air)
  cof=cof1a/(fn2+(f*f/fn2))+cof1b/(fro+(f*f/fro))
  cof=cof*cof1c
  air=8.6859*f**2*(cof+cof1d)

RETURN
END