

The Pennsylvania State University  
The Graduate School  
College of Engineering

**TRAILING EDGE NOISE PREDICTION USING THE  
NONLINEAR DISTURBANCE EQUATIONS**

A Thesis in  
Aerospace Engineering  
by  
James Paul Erwin

© 2009 James Paul Erwin

Submitted in Partial Fulfillment  
of the Requirements  
for the Degree of

Master of Science

May 2009

The thesis of James Paul Erwin was reviewed and approved\* by the following:

Kenneth S. Brentner  
Professor of Aerospace Engineering  
Thesis Co-Advisor

Philip J. Morris  
Boeing/A.D. Welliver Professor of Aerospace Engineering  
Thesis Co-Advisor

George A. Lesieutre  
Professor and Head, Aerospace Engineering

\*Signatures are on file in the Graduate School.

# Abstract

Direct computation of broadband noise in reasonable computational times is possible if the calculations can be focused in a specific noise making region of interest. Turbulent boundary layer trailing edge (TBL-TE) noise is of special research interest in the wind turbine aeroacoustic community because, in large scale applications, broadband noise can propagate long distances. A new computational aeroacoustic method is proposed that solves the Nonlinear Disturbance Equations (NLDE) in refined time and spatial scales surrounding the trailing edge of airfoil blades. PSU-WOPWOP is Penn State's noise propagation and prediction software, and it uses the NLDE solution to propagate the acoustic solution to observers. A new NLDE flow solver is presented with validation cases. The Blade Systems Design Study (BSDS) flatback airfoil section and an NACA 0012 airfoil are tested. Noise is generated due to unsteady laminar vortex shedding and boundary layer interaction, but no TBL-TE noise has been predicted. Turbulence injection techniques have yet to be introduced that allow for the direct computation of TBL-TE noise. For the NLDE tool to be effective in TBL-TE noise prediction, turbulence must be injected at the inflow of the trailing edge computational domain.

# Table of Contents

<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>x</b>
<b>List of Symbols</b>	<b>xi</b>
<b>Acknowledgments</b>	<b>xiii</b>
<b>Chapter 1</b>	
<b>Introduction</b>	<b>1</b>
1.1 Types of Noise . . . . .	2
1.2 Blade Self-Noise . . . . .	3
1.2.1 Turbulent Boundary Layer - Trailing Edge Noise . . . . .	4
1.3 Wind Turbine Noise . . . . .	5
1.4 Motivation for Research . . . . .	6
<b>Chapter 2</b>	
<b>Trailing Edge Noise Prediction</b>	<b>7</b>
2.1 Previous Noise Prediction Methods . . . . .	7
2.2 New Trailing Edge Noise Prediction Method . . . . .	8
2.3 Issues With TBL-TE Noise Prediction . . . . .	9
2.3.1 Time Accuracy . . . . .	9
2.3.2 Flow Scale Resolution . . . . .	10
2.3.3 Grid Generation . . . . .	11
2.3.4 Low Dispersion and Low Dissipation Numerical Schemes . . . . .	14
2.3.5 Aeroacoustic boundary conditions . . . . .	15
2.4 Development of Nonlinear Disturbance Equations Flow Solver . . . . .	15
<b>Chapter 3</b>	
<b>The Nonlinear Disturbance Equations (NLDE)</b>	<b>16</b>
3.1 NLDE Derivation . . . . .	17

3.2	NLDE Flow Solver . . . . .	21
3.2.1	Runge-Kutta Time Integration . . . . .	21
3.2.2	Time Step Stability . . . . .	22
3.2.3	Dispersion Relation Preserving Finite Differencing . . . . .	23
3.2.4	Grid Metrics . . . . .	24
3.2.5	Boundary Conditions . . . . .	25
3.2.5.1	Radiation Boundary Conditions . . . . .	26
3.2.5.2	No-Slip Adiabatic Wall . . . . .	27
3.2.6	Low Pass Filtering . . . . .	29
3.2.7	Message Passing Interface Parallelization . . . . .	30
3.3	Validation of the NLDE Code . . . . .	31
<b>Chapter 4</b>		
	<b>NLDE Flow Solver Validation</b>	<b>32</b>
4.1	2-D Gaussian Pulse Propagation . . . . .	32
4.2	3-D Gaussian Pulse Propagation . . . . .	34
4.3	2-D Solid Wall Reflection . . . . .	35
4.4	Scattering of a Gaussian Pulse by a Circular Cylinder . . . . .	36
<b>Chapter 5</b>		
	<b>Circular Cylinder and Airfoil Noise Prediction</b>	<b>39</b>
5.1	Circular Cylinder Flow . . . . .	39
5.1.1	Results and Discussion . . . . .	40
5.2	Airfoil Noise Prediction . . . . .	42
5.2.1	NACA 0012 . . . . .	43
5.2.2	BSDS Flatback . . . . .	44
<b>Chapter 6</b>		
	<b>Suggestions for Future Work and Summary</b>	<b>48</b>
6.1	Turbulence Injection . . . . .	48
6.2	Extended NLDE Code Validation . . . . .	49
6.2.1	Circular Cylinder Cases . . . . .	49
6.2.2	Airfoil cases . . . . .	49
6.2.3	Grid Refinement Studies . . . . .	50
6.3	NLDE Code Improvements . . . . .	50
6.3.1	Low Pass Filtering . . . . .	50
6.3.2	C-type Grid Support . . . . .	50
6.3.3	Acoustic Data Surfaces . . . . .	51
6.3.4	Multi Time Stepping . . . . .	51
6.4	Summary and Future Direction . . . . .	51

<b>Bibliography</b>	<b>53</b>
<b>Appendix A</b>	
<b>Acoustic Scattering</b>	<b>57</b>
A.1 Coupling with PSU-WOPWOP . . . . .	58
A.2 Validation Case . . . . .	59
A.3 Acoustic Scattering of Rotorcraft . . . . .	60
A.3.1 BO105 Tail Rotor Scattering . . . . .	60
A.3.2 Quad Tilt Rotor Scattering . . . . .	61
<b>Appendix B</b>	
<b>Source Code for Exact Solutions</b>	<b>65</b>
B.1 Scattering of a Gaussian Pulse by a Circular Cylinder . . . . .	65
B.1.1 MATLAB code . . . . .	66
B.2 Propagation of a Gaussian Pulse with Mach 0.5 Background Flow .	70
B.2.1 MATLAB code . . . . .	70
<b>Appendix C</b>	
<b>NLDE Flow Solver User Manual</b>	<b>73</b>
C.1 Compiling and Running the NLDE Code . . . . .	73
C.1.1 Compiling the NLDE Code on Linux Clusters . . . . .	73
C.1.2 Submitting a Parallel Job . . . . .	74
C.1.3 Preparing a Job to Run . . . . .	75
C.1.4 The Job Namelist File . . . . .	76
C.1.5 PLOT3D File Format . . . . .	82
C.2 Example: 2-D Gauss Pulse Case File . . . . .	83

# List of Figures

- 1.1 Turbulent boundary layer - trailing edge noise generation . . . . . 2
- 1.2 Five blade self-noise generation mechanisms[8]. . . . . 4
  
- 2.1 A schematic of the problem solving process . . . . . 8
- 2.2 A C-type grid is commonly used when the blade has a sharp trailing edge defined by a single point . . . . . 12
- 2.3 An O-type grid is commonly used when the blade has a blunt trailing edge defined by multiple grid points . . . . . 13
- 2.4 Hyperbolic tangent grid stretching along the upper trailing edge blade surface. . . . . 14
  
- 3.1 Decomposition of a flow variable. . . . . 16
- 3.2 NLDE integration procedure (1-D for simplicity). . . . . 20
- 3.3 No-slip adiabatic wall boundary treatment. . . . . 28
- 3.4 Instantaneous velocity vectors with uniform rightward mean flow. Uniform flow violates no slip condition at surface (left). After no-slip condition is applied to perturbation variables, instantaneous flow is corrected (right). . . . . 28
- 3.5 Domain decomposition for MPI parallelization. Master processor (red) and child processors (blue). . . . . 31
  
- 4.1 Initial Gaussian pulse distribution imposed on perturbation density at time  $t = 0$ . Contour of domain (left), Line plot along  $y = 0$  (right). 33
- 4.2 Validation of 2-D Gaussian pulse propagation in a stationary medium. 34
- 4.3 Validation of 2-D Gaussian pulse propagation in  $M = 0.5$  medium (rightward). . . . . 35
- 4.4 Validation of 3-D Gaussian pulse propagation in a stationary medium. 36
- 4.5 Validation of Tam and Dong solid wall reflection case in a stationary medium. . . . . 37
- 4.6 Gaussian pulse scattering problem setup and results. . . . . 38

5.1	Two circular cylinder grids used to model shedding and develop acoustic data surfaces. . . . .	40
5.2	Two observer locations for circular cylinder shedding case (not to scale). . . . .	41
5.3	Contours of velocity magnitude for the coarse and fine grid cylinder	42
5.4	SPL results of cylinder with NLDE - PSU-WOPWOP prediction (a) compared with experimental data in the transverse direction [25] and (b) directivity plots for the predicted shedding frequency and its first harmonic for the fine cylinder . . . . .	42
5.5	NACA 0012 grid for NLDE noise prediction . . . . .	44
5.6	Pressure perturbation solution after 0.025 seconds for the NACA 0012 airfoil at $Re_c \approx 4.5$ million and $0^\circ$ angle of attack (a) for the entire airfoil section and (b)for a zoomed in trailing edge region. . .	44
5.7	Pressure perturbation contours after 0.025 seconds for the NACA 0012 airfoil at $Re_c \approx 4.5$ million and $0^\circ$ angle of attack. . . . .	45
5.8	SPL results of NLDE - PSU-WOPWOP prediction for NACA 0012 airfoil (a) at observer placed at $90^\circ$ (b) directivity plots at shedding frequency $\approx 2300$ Hz. All observers are 5 chords from trailing edge.	45
5.9	The Blade Systems Design Study “flatback” wind turbine blade (root section) . . . . .	46
5.10	BSDS root grid for NLDE noise prediction . . . . .	46
5.11	Pressure perturbation solution after 0.025 seconds for the flatback airfoil. . . . .	47
5.12	SPL results of NLDE - PSU-WOPWOP prediction for BSDS flat-back airfoil (a) at observer placed at $90^\circ$ (b) directivity plots at shedding frequency $\approx 150$ Hz. All observers are 5 chords from trailing edge. . . . .	47
6.1	Turbulence injection, recycling, and rescaling technique proposed for the NLDE code . . . . .	49
A.1	Acoustic scattering problem solving process. . . . .	58
A.2	Setup of the scattering sphere validation case.[15] . . . . .	59
A.3	Validation of a point source scattering problem. The pressure magnitude is normalized by the magnitude of the incident pressure at $\theta = 0$ : (a) the incident field, (b) the total field.[15] . . . . .	60
A.4	Sound pressure level for a point source scattering validation case. The observer grid is located 5 m below the source: (a) the incident field, (b) the total field.[15] . . . . .	61
A.5	BO105 helicopter configuration.[15] . . . . .	62

A.6	Sound pressure level for the tail rotor noise at the second harmonic of tail rotor BPF. The observer grid is located 11 m below the main rotor hub: (a) the incident field, (b) the total field.[15]	63
A.7	The conceptual QTR selected for acoustic scattering investigation: (a) isometric view, (b) top view.[15]	63
A.8	Two QTR configurations: (a) high rotor configuration, (b) low rotor configuration.[15]	64
A.9	Sound pressure level for the QTR noise at the fifth harmonic of rotor BPF. The observer grid is located 10 m below the front rotor hub: (a) the incident field, (b) the total field for high rotor configuration, (c) the total field for low rotor configuration.[15]	64
C.1	Tree structure for example case	76

# List of Tables

3.1	The coefficients of the LDDRK scheme [13]. . . . .	22
3.2	The coefficients of the DRP scheme [28]. . . . .	24
3.3	Low pass filter coefficients for points 4 through $N-3$ . . . . .	29
3.4	Low pass filter coefficients for points 2 and $N-1$ . . . . .	30
3.5	Low pass filter coefficients for points 3 and $N-2$ . . . . .	30
A.1	Operating conditions of the BO105 tail rotor.[15] . . . . .	61
A.2	Operating conditions of the QTR.[15] . . . . .	62

# List of Symbols

$\alpha$	The adjustable low pass filter damping coefficient
BPF	Blade passage frequency
$C$	Speed of sound
CAA	Computational Aeroacoustics
DNS	Direct numerical simulation
FSC	NASA Langley's Fast Scattering Code
FWH	Ffowcs-Williams and Hawkings (equation)
$J$	Jacobian of grid transformation
$k$	Thermal conductivity
LES	Large eddy simulation
$M$	Mach number
$\mu$	Dynamic viscosity
$\nu$	Kinematic viscosity ( $\nu = \mu/\rho$ )
$p$	Total pressure
$\phi$	A scalar field variable
$\phi_\beta$	Shorthand notation for the differential $d\phi/d\beta$ .
$q_i$	Heat flux in the three coordinate directions

$\mathbf{q}$	Conserved vector of flow variables
$\rho$	Density
$R$	Universal gas constant
$Re_c$	Reynolds number based on chord length
$T$	Temperature
$\tau_{ij}$	Shear stress component in the fluid on face $i$ in direction $j$
TBL	Turbulent boundary layer
TE	Trailing edge
$u$	$x$ component of velocity
$v$	$y$ component of velocity
$w$	Half width of a Gaussian pulse, or $z$ component of velocity

# Acknowledgments

I would like to acknowledge Professor of Aerospace Engineering Kenneth S. Brentner for inviting me to work with him at Penn State University, and giving me the opportunity to continue my education. I also want to thank Boeing/A. D. Welliver Professor Dr. Philip Morris for use of his Nonlinear Disturbance Equation formulation and intellectual support and guidance throughout my research.

Words cannot express my appreciation for how much my family and close friends have been with me throughout my entire education. Thank you all, for your never ending concern, care, and motivation through the most challenging times.

This research was supported by Sandia National Laboratories, Purchase Order No. A0342 677302. I would like to thank Dr. Dale Berg and Dr. Matthew Barone who were the Technical Monitors on the project. Also thank you to my colleagues: Leonard Lopes, Christopher Hennes, Seongkyu Lee, and Pavanakumar Mohanamuraly for all of your guidance while I worked closely with you.

# Chapter 1

## Introduction

In the aerospace community, close attention must be paid to the generation of undesirable noise because it has an impact on the public acceptance of some renewable energy sources. Specifically, wind turbines are growing in popularity as an alternative energy source and have the potential to be noisy. The prediction and reduction of noise has proven to be a unique and challenging aspect of engineering design. The goal of the research presented in this thesis is a method to predict trailing edge broadband noise, which is generated by the scattering of turbulence as it passes the trailing edge of airfoil blades. The unsteady turbulent eddies in the flow that passes the trailing edges of airfoils provides one of the most significant sources of the total broadband noise. This is referred to as turbulent boundary layer trailing edge (TBL-TE) noise. The presence of the sharp trailing edge scatters the sound generated by the turbulent eddies very efficiently, especially in the immediate vicinity of the edge. Figure 1.1 is a simple illustration of how a trailing edge generates noise from the energy in the turbulence. Trailing edge noise can consist of substantially higher frequencies than the blade passage frequency (BPF) of helicopter and wind turbine rotors, and has the potential to be annoying because the frequency can be in the range of most sensitive human hearing. The prediction of trailing edge noise for wind turbines has previously been based on semi-empirical methods originally developed for helicopter rotor noise predictions.[8] This semi-empirical method is mostly based on two dimensional measurements from symmetric NACA airfoil sections such as the NACA 0012 airfoil, which is not particularly representative of the type of airfoils used

in large scale wind turbines. In the present research, a first-principles prediction methodology for trailing edge noise is developed, enabling the shape of the blade to be properly reflected in wind turbine noise predictions. To understand where TBL-TE noise fits into the collection of noise generating mechanisms, the following section briefly discusses other types of noise and their sources. Then, a focus on wind turbine noise will emphasize the importance of TBL-TE noise prediction.

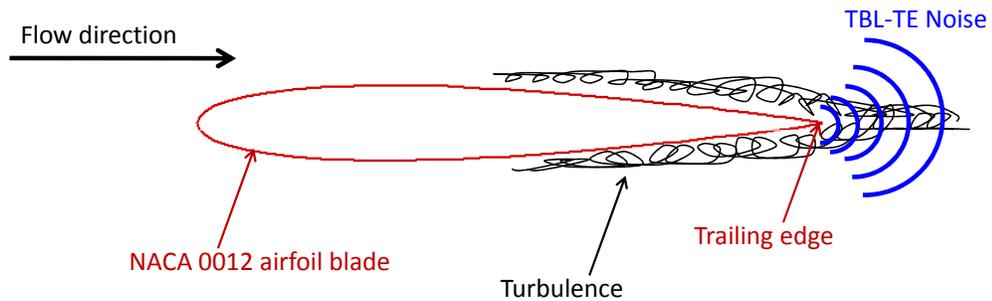


Figure 1.1: Turbulent boundary layer - trailing edge noise generation

## 1.1 Types of Noise

In this section, various noise generation mechanisms will be discussed. The noise generating components of wind turbines are similar to those of aerospace vehicles, and prediction and reduction techniques used in the aerospace industry can be adopted by the wind turbine community.

Power train noise is generated by the components of a wind turbine responsible for transmitting the power from the rotor to the power generator. In typical wind turbines, the power train consists of a main shaft, a gearbox, and a high speed shaft. The purpose of the gearbox is to convert the low speed, high torque output of the rotor to the high speed, low torque input for the power generator. Typical 600 to 750 kW machines can have gear ratios up to 50:1, and require meshing gears of different sizes and speeds[2]. This gear tooth meshing causes noise which can be heard in the proximity of the turbine. Similar to the massive gearbox operating in a wind turbine is the gear reduction required for rotorcraft flight. Gas turbine engine output speed is reduced through many stages of gears which also creates substantial noise inside and out of the cockpit[1]. Transmission

noise can often be associated with spiral bevel gears, which are commonly used in transmissions for improved load-bearing. The gear design is modified to allow for case fluctuations during maneuver, so there is always a small difference between the actual transmission output and the theoretical output from the gear sizes.[16] Off-design operation causes constant operational vibration and noise. Most horizontal axis wind turbines are supported with a tower with a large surface area. The vibrations generated by the power train are transferred through the tower, and propagate noise into the environment. Previous research has been conducted to actively or passively dampen these structural vibrations and the source of vibrations, including, but not limited to, active structural acoustic control [20], active gear-mesh noise control [22], and spiral bevel transmission gears [16]. However, even a wind turbine with a silent power train and no structural vibrations will generate noise due to unsteady aerodynamics.

Aerodynamic noise is generated due to the turbulent flow of air around a body. The fluctuating pressure on wind turbine blades, similar to aircraft wings and helicopter rotors, creates sound waves which can propagate significant distances. A common example of an aeroacoustic noise source is the sound of a helicopter flying overhead. A large component of helicopter noise is due to the dynamic flow environment surrounding the rotor. Although there are many types of aerodynamic noise, the work in this thesis focuses on the prediction of blade self-noise, which is generated by the boundary layer and wake aerodynamics near the surface of the blades.

## 1.2 Blade Self-Noise

Blade self-noise is the noise produced by a blade's interaction with turbulence generated by its own boundary layer and near wake. In 1989, Brooks, Pope, and Marcolini performed experiments to develop semi-empirical blade self-noise prediction models[8]. Five self-noise generation mechanisms were modeled semi-empirically through a characterization of aerodynamic and acoustic wind tunnel tests of airfoil sections.

The dynamic flow environment around an airfoil blade, even with laminar inflow, generates aerodynamic noise. In subsonic flow conditions, Brooks *et al.* pro-

pose five major noise generation mechanisms that are illustrated in Figure 1.2. First, high Reynolds number turbulent boundary layers generated over the surface of airfoils generate noise as they pass the trailing edge. Second, for laminar boundary layers at low Reynolds numbers, instabilities result in vortex shedding and associated noise from the trailing edge. At high angles of attack, the separated flow near the trailing edge causes large scale separation similar to bluff body flow, and low frequency noise known as separation-stall noise. Also, vortex-shedding noise is the noise generated by the pressure fluctuations caused by vorticity oscillations behind a blunt trailing edge. Lastly, tip vortex noise is generated near the tips of lifting blades due to the circulation of finite length blades. The goal of the present research is to accurately predict turbulent boundary layer - trailing edge noise, but the basic method can be applied to all of these self-noise mechanisms.

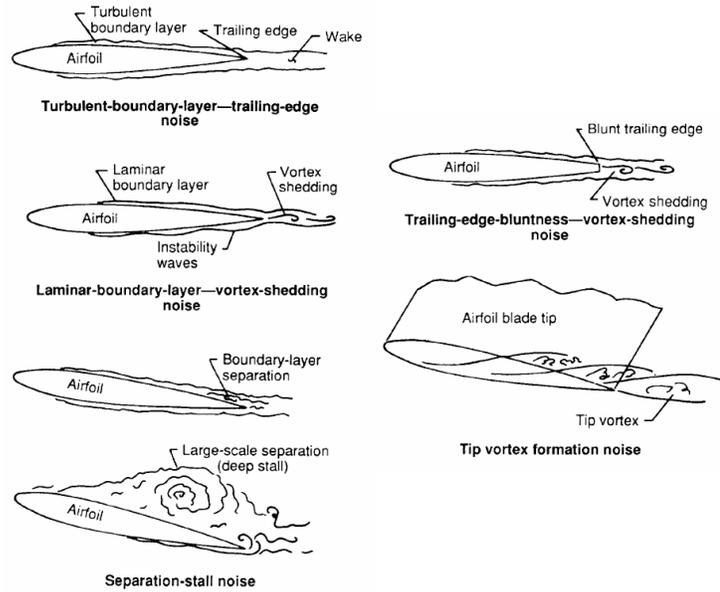


Figure 1.2: Five blade self-noise generation mechanisms[8].

### 1.2.1 Turbulent Boundary Layer - Trailing Edge Noise

Turbulent boundary layer trailing edge noise is caused by the interaction of turbulence passing the trailing edge of airfoil blades. Brooks and Hodgson [7] showed that TBL-TE noise can be predicted accurately if high fidelity turbulent bound-

ary layer pressures are known. However, most previous TBL-TE noise prediction strategies have been based on empirical descriptions of the surface pressure. Schlinker and Amiet [26] argued that due to lack of agreement among test cases, a higher fidelity method of providing surface pressure should be used. The time history of the turbulence must be accounted for in addition to the time averaged characteristics, and the conditions under which turbulence is developed plays an important role in accurate TBL-TE noise prediction. The focus of the present research is to use computational fluid dynamics (CFD) in a first principles based method to predict the TBL and corresponding noise.

TBL-TE noise can be thought of a scattering process, where the turbulent energy in the boundary layer is scattered by the trailing edge and converted into acoustic energy. This generation process should not be confused with the acoustic scattering of preexisting sound waves by solid bodies. Additional work has been completed on bluff body acoustic scattering as part of the author's thesis research and is presented in Appendix A.

### 1.3 Wind Turbine Noise

Large scale wind turbine blades generate complex noise signals that must be understood and managed so that they can be accepted in neighboring communities. Although the amplitude of wind turbine noise is significantly less than aircraft noise, it can still be significant compared to the low environmental noise levels common to rural areas where turbines are installed [37]. Because wind turbine trailing edge noise scales with local blade section velocity to the fifth power[8], turbine noise can increase significantly with increased rotational rates. Low frequency wind turbine noise is a result of the low BPF of the rotor, and can propagate long distances because of low atmospheric attenuation. Although the BPF is sometimes below the threshold of human hearing (the frequency is too low for humans to hear), broadband noise generated by wind turbines consists of frequencies that are audible and can also propagate long distances. Wind turbine broadband noise occurs at substantially lower frequencies than aircraft noise, so it can propagate much further than traditional aircraft broadband noise. An understanding of broadband noise generation is crucial in the prediction of wind turbine noise. Broadband

ingestion noise is generated due the turbine operating in unsteady aerodynamic environments as the revolving blades ingest turbulence generated by the terrain, changing wind direction, and previous blade passages. However, the focus of the present research is strictly limited to the self-noise characteristics of the airfoil blades. After the tools have been developed to predict self-noise, they can easily be extended to broadband ingestion noise prediction.

The noise generated by wind turbine blades is a major design restriction because they cannot be used if they exceed noise level regulations. A turbine may be installed in a variety of geographical locations and operating conditions. The measured experimental data available for wind turbines often insufficiently defines the noise characteristics. Therefore, operators and manufacturers are conservative in their designs to leave a buffer for unknown factors after the turbine is installed. Current wind turbine noise prediction methods are relatively crude and do not provide the guidance needed by the wind turbine industry.

## 1.4 Motivation for Research

A first principles noise prediction method would allow for turbines to operate both efficiently and quietly, but wind turbine noise prediction has unique challenges. This thesis addresses these challenges and presents the development of a higher fidelity approach to the prediction of TBL-TE noise. Chapter 2 is an overview of the current noise prediction methods, and the new broadband noise prediction method will be discussed. In Chapter 3, the details of a new CFD flow solver will be presented which is a first-principles based tool that will exceed the capability of previous empirically based models. Validation cases will be presented in Chapter 4 that demonstrate the accuracy of the new code. In Chapter 5, first attempts at noise prediction of circular cylinder flow and airfoils will be presented. Chapter 6 concludes the thesis with suggestions for future research. Appendix A presents research on bluff body scattering of acoustic signals. Appendix B is a collection of MATLAB code used to generate exact solutions of the validation cases for the new CFD code written, and Appendix C contains the user manual for the code.

# Chapter 2

## Trailing Edge Noise Prediction

Computational aeroacoustics, or CAA, consists of a wide range of methodologies where noise is predicted through a numerical solution of the propagation of sound waves in an inhomogeneous flow field. The advancement of modern day computers has allowed CAA to grow and become a popular choice for noise prediction. The prediction of aeroacoustic noise as part of the preliminary design phase of any aerospace application is necessary for a low noise design.

### 2.1 Previous Noise Prediction Methods

A direct numerical simulation (DNS) solves the compressible Navier-Stokes equations for the acoustic field directly. It attempts to simulate all features of the flow, ranging from the smallest of turbulent flow scales to long range sound propagation. Due to flow scale resolution, DNS is extremely computationally demanding and proves to be limited in use for full scale calculations. Large eddy simulation (LES) only attempts to resolve the larger turbulent scales in the flow and models the finer scales. However, LES is also too computationally demanding for practical use when attempting to solve for the noise of the entire wind turbine.

The other approaches to CAA are known as hybrid approaches, which attempt to separate the total noise prediction problem into smaller problems which can be solved more efficiently. For example, a CFD tool is used to calculate the flow field in the noise generating region. Then, an acoustic solver is used to obtain acoustic pressure information and compute how it is radiated into the far field. The

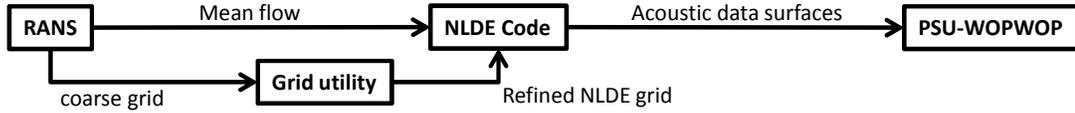


Figure 2.1: A schematic of the problem solving process

acoustic propagation is commonly calculated with an integral type method, such as Lighthill’s analogy, Kirchhoff integration, or the Ffowcs-Williams and Hawkings (FWH) equation. For example, the noise prediction and propagation code PSU-WOPWOP was written at Penn State University and is a FWH solver. The FWH equation is a rearrangement of the traditional Navier-Stokes equations into the form of an inhomogeneous wave equation. It consists of quadrupole terms in the volume surrounding the source, and monopole and dipole terms on the surface of the source. The discrete terms can be associated with specific physical characteristics of the flow, but this is beyond the scope of the thesis. Brentner and Farassat give an in-depth analysis of this CAA method [6]. A new method will now be presented that predicts the high fidelity turbulence in the flow that creates the noise, and also propagates the noise to locations of interest.

## 2.2 New Trailing Edge Noise Prediction Method

A tractable approach is to separate the total CAA problem into component problems and individually apply the most computationally efficient solution methods. The solution procedure is outlined in Figure 2.1. First, a Reynolds Averaged Navier Stokes (RANS) solver is used to calculate the time averaged (mean) flow around the entire turbine blade. Then, the Nonlinear Disturbance Equations (NLDE) will be used to simulate the fine scales in the boundary layer near the airfoil trailing edge. This technique, developed by Morris *et al.* [24] is essentially LES except that it solves for perturbations about an estimate of the mean flow, in a limited region of the total flow. The unsteady surface pressures and/or unsteady flow field calculated by the NLDE are passed to the PSU-WOPWOP noise prediction code, which is used to predict the noise from the trailing edge.

The advantage of the NLDE method over traditional LES for aeroacoustic applications is that it increases the accuracy of the solution in the acoustic field and it can also establish a statistically stationary solution faster. This reduces the total computation time. In a full closure of the flow solver, a turbulence model would be implemented to model the unresolved scales in the boundary layer. The NLDE are only used for the simulation of the flow in the trailing edge region of the airfoil. They predict both fluctuations on the airfoil surface as well as the fine-scale turbulence in the vicinity of the blade trailing edge. The NLDE can be solved in a small fraction of time relative to what would be required if the flow field for the entire blade needed to be calculated. The grids required for the NLDE calculations are finer than the grids needed for the mean flow computations. In conjunction with the permeable surface FWH equation method, these compressible flow simulations enable the associated noise to be predicted.

The acoustic propagation tools used in CAA are only as good as the pressure data provided to them. An accurate FW-H solver such as PSU-WOPWOP will have limited use if the flow data given to it is low fidelity. Among all the different types of noise sources, broadband noise is arguably most difficult to predict because it can originate from the smallest of flow scales, boundary layer turbulence on blade surfaces. TBL-TE noise, discussed in Section 1.2.1, is a result of this type of flow scale. The following sections describe a few challenges associated with CAA and the strategies used to address these challenges.

## **2.3 Issues With TBL-TE Noise Prediction**

In this section, some unique issues associated with CAA will be discussed. A new flow solver has been written that addresses these issues. As each topic is presented, a discussion follows of how the new CAA approach is beneficial.

### **2.3.1 Time Accuracy**

Noise prediction requires a time dependent solution of the propagation of sound waves. CAA solutions are, in general, not steady-state and therefore require explicit time stepping. If a steady-state solution was desired, an implicit scheme

with inherent numerical damping could be used. In general, the solution should be calculated with a constant time step which meets the stability criteria but also minimizes run time. Explicit schemes require that a pressure wave travel less than one grid point per time step. Because of the finite differencing method, if a pressure wave travels further than one cell, the surrounding cells would not be able to properly influence the propagation. Therefore, the size of the grid spacing and local speed of sound strictly limit the size of the time stepping used. For a wave traveling between two grid points separated by a spacing of  $\Delta x$ ,

$$\Delta t < \frac{\Delta x}{C + u} \quad (2.1)$$

where  $C$  is the local speed of sound,  $u$  is the local flow velocity in the wave propagation direction, and  $\Delta t$  is an upper bound on the allowable time stepping in the explicit scheme. The new solver uses explicit five stage, fourth order accurate time marching. The scheme selected is a Low Dissipation and Dispersion Runge-Kutta (LDDRK) method outlined by Hu, Hussaini, and Mantey [13]. Details on the time marching scheme can be found in Section 3.2.1, where a more detailed limit on the time step size is also presented.

### 2.3.2 Flow Scale Resolution

A main challenge with CAA is the resolution of flow scales. The computational power available today is not sufficient to perform a DNS or even LES of the entire flow field. The smallest of turbulent flow scales can only be resolved with fine grid spacing surrounding wall boundaries. In addition, to resolve turbulent flow with DNS, scales must be resolved from the smallest eddies to scales that are on order of the size of the physical domain. However, if the resolution on the body surface is kept throughout the entire domain, the number of grid points grow to unacceptable values. Therefore, the fine grid spacing required to accurately predict turbulence cannot be used to propagate waves to a distant location of interest. As a result, turbulence modeling is often used to simulate the smallest of viscous scales.

One solution to resolving flow scales is to use stretched grids. These types of grids are very fine along solid bodies which require time accurate viscous resolution, and stretch to larger spacing as the grid extends away from the body. The larger

scales away from the body are resolved by the coarser grid without sacrificing accuracy near the body, but this practice often requires substantial time for grid generation.

The new NLDE code only solves for the flowfield in the trailing edge region of the airfoil blade. The grids have fine spatial resolution to simulate the TBL's interaction with the blade trailing edge. The FW-H solver PSU-WOPWOP uses this solution to propagate the acoustic signal to locations far away from the wind turbine. This strategy does not depend on the grid resolution in the domain between the blade CFD solution and the observer positions.

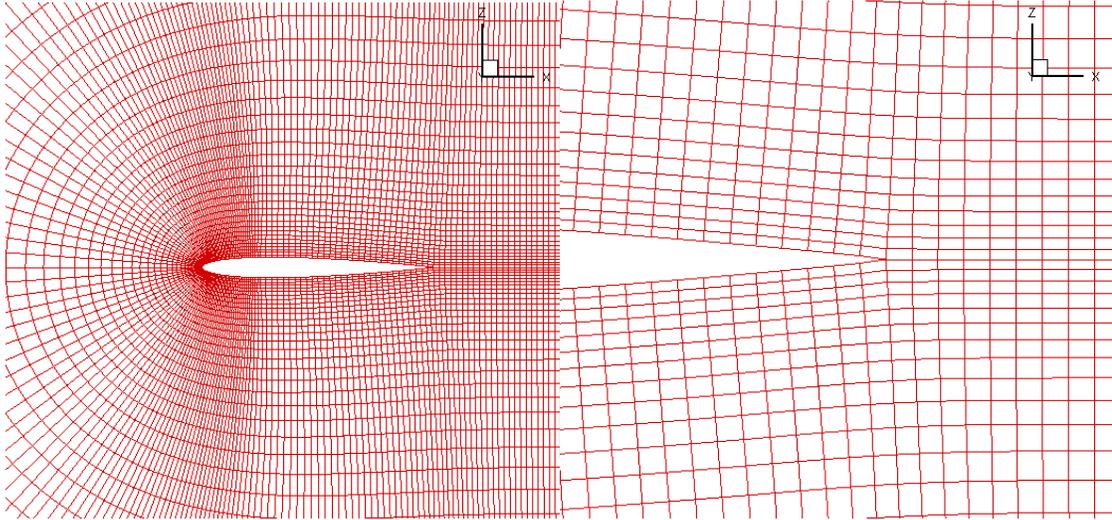
### 2.3.3 Grid Generation

Grid generation remains one of the challenging aspects of CFD. Efficient and robust simulations rely on high fidelity grids built around complex geometries. In the specific case of TBL-TE noise, the flow field needs to be highly resolved in the boundary layer region of the blade surface to resolve time accurate turbulent structures. Although generating grids for simple blade geometries is easier than complex geometries, the basic practices and theory behind grid generation remains the same. In 1988, the Integrated Space Shuttle Launch Vehicle (SSLV) was one of the first applications in complex overset CFD grid generation. The first computations were performed on a simplified model with less than a million grid points. More advanced grids were generated for the SSLV which contained around 16 million points and required approximately two man years for the grid generation alone[9]. Since then, tools have been developed to allow for faster grid generation. In this section, a few key points will be addressed which are required to generate grids for TBL-TE noise prediction.

#### C versus O-type Blade Grids

C and O-type blade grids should be used depending mostly on the shape of the trailing edge of the blade section. If the blade has a sharp trailing edge defined by a single grid point and zero thickness, a C-type grid should be used as seen in Figure 2.2. The grid is C shaped because it wraps around the nose of the blade section and continues into the downstream direction on both top and bottom of the blade.

If the trailing edge has a finite thickness where a single grid point cannot be used to define the trailing edge, an O-type grid should be used as seen in Figure 2.3.



(a) C-grid wrapping around an NACA 0009 blade section (b) Zoomed in view of trailing edge region

Figure 2.2: A C-type grid is commonly used when the blade has a sharp trailing edge defined by a single point

The NLDE code written currently only supports O-type grids. These grids are used for all the test cases including “sharp” airfoil sections with trailing edge thicknesses of 0.1% chord. In the future the code will need to be expanded to support C-type grids for infinitely thin trailing edges.

### OVERGRID and Chimera Grid Tools

The grid generation utilities used in the present research are collectively distributed in a package called Chimera Grid Tools (CGT) developed by NASA. They are written in the Fortran programming language, compiled and executed in the Linux operating environment. Of all the available features CGT offers, a few were used most when developing the code and are summarized here:

- **GRIDED** is a variety of grid editing functions for structured grids. Among these functions are swapping/reversing grid indices, extracting grid sections, translating and scaling grids, and extrapolating grid layers.

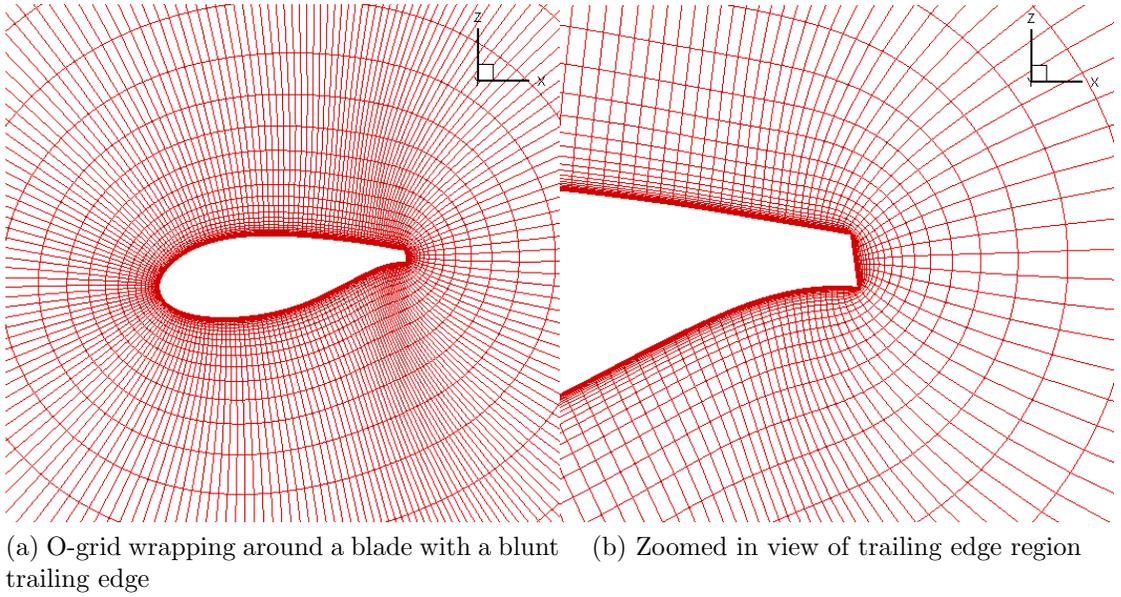


Figure 2.3: An O-type grid is commonly used when the blade has a blunt trailing edge defined by multiple grid points

- **GRIDINF** gives basic grid information about PLOT3D grid files. Some types of information are x,y, and z location of a given point, bounding box dimensions, cell size statistics, total number of grid points, and file type information.
- **HYPGEN** generates hyperbolic type stretched volume grids from a single surface grid.
- **SRAP** can redistribute and project points in a grid based on input parameters such as minimum and maximum spacings, end spacings, stretch ratios, and more.

### Viscous Wall Spacing

The wall spacing of a grid needs to be able to resolve turbulent structures of certain scales. A nondimensional parameter  $y^+$  can be defined as

$$y^+ = \frac{(u_\tau y)}{\nu} \quad (2.2)$$

where  $u_\tau = \sqrt{\nu(\partial u/\partial y)_{wall}}$  is the friction velocity,  $y$  is the distance between the first grid point away from the wall and the wall surface, and  $\nu$  is the kinematic viscosity. In the case of turbulence resolution, grids should be resolved such that  $y^+ \approx 1.0$ . For example, Lund *et al.* used  $y^+ \approx 1.2$  when generating inflow turbulence for spatially developing boundary layer simulations, while also including turbulence models [18].

**HYPGEN** is used to generate grids with viscous wall spacing. The initial wall spacing can be specified based on the flow Reynolds number while **HYPGEN** stretches the grid with a desired stretch ratio using either hyperbolic tangent or exponential stretching in the wall normal direction. Figure 2.4 is an example of hyperbolic tangent grid stretching in the trailing edge portion of the blade section pictured in Figure 2.3.

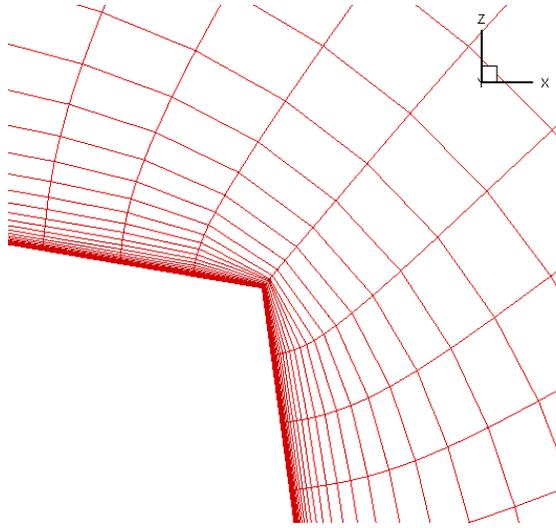


Figure 2.4: Hyperbolic tangent grid stretching along the upper trailing edge blade surface.

### 2.3.4 Low Dispersion and Low Dissipation Numerical Schemes

The propagation characteristics of waves governed by a series of partial differential equations depend on the dispersion relation in frequency and wave number space. The dispersion relation relates wave angular frequency and wave numbers of flow variables [31]. Using Fourier transforms of the governing equations, Tam and

Webb developed a class of finite difference schemes known as Dispersion-Relation-Preserving (DRP) schemes. A fourth order accurate DRP scheme was selected for the NLDE code, which has better dispersion characteristics than a standard sixth order central difference grid. Details of the scheme used in the new NLDE code can be found in Section 3.2.3. Also, a Low-Dispersion and Low-Dissipation scheme was used for time integration and is discussed in Section 3.2.1. Explicit low-pass numerical filtering described in Section 3.2.6 is also used to prevent the growth of high wave number modes inherent to central differencing schemes.

### **2.3.5 Aeroacoustic boundary conditions**

Wave reflections from exterior boundaries should always be minimized in a CFD solution. High fidelity aeroacoustic solutions require the boundaries of the computational domain to allow acoustic waves to leave without any reflection. Farfield radiation conditions are adopted by Tam and Webb [31] which are derived from the asymptotic solutions of the linearized Euler equations (see Section 3.2.5.1). All boundaries away from a solid wall include some form of these radiation conditions, so acoustic waves leave without reflecting back and altering the interior solution.

Solid wall viscous boundary conditions enforce a no-slip condition at the wall. In addition, an adiabatic condition enforces zero pressure and density gradient normal to the wall surface. Aeroacoustic boundary conditions are applied to only the perturbation variables in the solution because the mean flow is assumed to already satisfy the physical boundaries. Biased DRP finite differencing stencils are used to calculate flow derivatives at both the radiation and solid wall boundaries.

## **2.4 Development of Nonlinear Disturbance Equations Flow Solver**

A key component to the new trailing edge noise prediction method is the development of a flow solver capable of solving the NLDE. The rest of the thesis documents the creation, validation, and testing of a new NLDE flow solver. Previous knowledge of the challenges associated with trailing edge noise prediction guided the development process. In the following chapter, the Nonlinear Disturbance Equations and details of the flow solver will be presented.

# Chapter 3

## The Nonlinear Disturbance Equations (NLDE)

The nonlinear disturbance equations (NLDE) are a modified version of the Navier-Stokes equations. The modification involves the splitting of all instantaneous flow variables into a time independent (mean) component and time dependent (perturbation) components. Figure 3.1 shows how all scalar flow variables are decomposed. The left hand side is the instantaneous flow, which is the sum of the mean (subscript zero), a resolved perturbation (single prime), and a modeled sub-grid scale perturbation (double prime). In the present work, no sub-grid scale perturbations are modeled. The only perturbation component of the flow is from the resolved NLDE solution. The resulting equations are not time averaged like the Reynolds Averaged Navier Stokes (RANS) equations, but the time dependent portion of the NLDE is solved with a known mean flow as a constant forcing term. The mean flow in the present work is either obtained by a RANS solver or assumed uniform.

$$\phi = \phi_0 + \phi' + \phi''$$

The diagram illustrates the decomposition of a flow variable  $\phi$  into three components:  $\phi_0$  (mean),  $\phi'$  (resolved perturbation), and  $\phi''$  (modeled sub-grid scale perturbation). Each component is represented by a blue arrow pointing upwards to its corresponding term in the equation. Below the arrows are labels: 'instantaneous (total)' for  $\phi$ , 'mean (given)' for  $\phi_0$ , 'NLDE (resolved perturbation)' for  $\phi'$ , and 'modeled (sub-grid scale perturbation)' for  $\phi''$ .

Figure 3.1: Decomposition of a flow variable.

### 3.1 NLDE Derivation

The traditional compressible Navier-Stokes equations in Cartesian coordinates without body forces or external heat addition can be written in conservation form as

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} + \frac{\partial \mathbf{G}}{\partial z} = 0 \quad (3.1)$$

In the present implementation, the equations are to be integrated on curvilinear grids. Therefore, all of the physical derivatives need to be written in terms of generalized coordinates. Methods presented in Marsden *et al.* [19] are used to transform the equations. The grid metric calculations are covered in detail in Section 3.2.4. Let  $\xi$ ,  $\eta$ , and  $\zeta$  be the coordinates for the uniformly spaced computational space to be used by all the finite differencing schemes. The transformed equations may be written as

$$\begin{aligned} \frac{\partial}{\partial t} \left( \frac{\mathbf{q}}{J} \right) + \frac{\partial}{\partial \xi} \left( \frac{1}{J} [\xi_x \mathbf{E} + \xi_y \mathbf{F} + \xi_z \mathbf{G}] \right) \\ + \frac{\partial}{\partial \eta} \left( \frac{1}{J} [\eta_x \mathbf{E} + \eta_y \mathbf{F} + \eta_z \mathbf{G}] \right) \\ + \frac{\partial}{\partial \zeta} \left( \frac{1}{J} [\zeta_x \mathbf{E} + \zeta_y \mathbf{F} + \zeta_z \mathbf{G}] \right) = 0 \end{aligned} \quad (3.2)$$

where  $J$  is the Jacobian of grid transformation,  $\mathbf{q}$  is the instantaneous flow vector written as  $[\rho, \rho u, \rho v, \rho w, e]^T$ , and the instantaneous flux vectors may be written as

$$\mathbf{E} = \left\{ \begin{array}{c} \rho u \\ \rho u^2 + p - \tau_{xx} \\ \rho uv - \tau_{xy} \\ \rho uw - \tau_{xz} \\ (e + p)u - u\tau_{xx} - v\tau_{xy} - w\tau_{xz} + q_x \end{array} \right\} \quad (3.3)$$

$$\mathbf{F} = \left\{ \begin{array}{c} \rho v \\ \rho uv - \tau_{xy} \\ \rho v^2 + p - \tau_{yy} \\ \rho vw - \tau_{yz} \\ (e + p)v - u\tau_{xy} - v\tau_{yy} - w\tau_{yz} + q_y \end{array} \right\} \quad (3.4)$$

$$\mathbf{G} = \left\{ \begin{array}{c} \rho w \\ \rho u w - \tau_{xz} \\ \rho v w - \tau_{yz} \\ \rho w^2 + p - \tau_{zz} \\ (e + p)w - u\tau_{xz} - v\tau_{yz} - w\tau_{zz} + q_z \end{array} \right\} \quad (3.5)$$

The shear stress terms may be written in generalized coordinates as

$$\begin{aligned} \tau_{xx} &= \frac{2}{3}\mu [2(\xi_x u_\xi + \eta_x u_\eta + \zeta_x u_\zeta) - (\xi_y v_\xi + \eta_y v_\eta + \zeta_y v_\zeta) - (\xi_z w_\xi + \eta_z w_\eta + \zeta_z w_\zeta)] \\ \tau_{yy} &= \frac{2}{3}\mu [2(\xi_y v_\xi + \eta_y v_\eta + \zeta_y v_\zeta) - (\xi_x u_\xi + \eta_x u_\eta + \zeta_x u_\zeta) - (\xi_z w_\xi + \eta_z w_\eta + \zeta_z w_\zeta)] \\ \tau_{zz} &= \frac{2}{3}\mu [2(\xi_z w_\xi + \eta_z w_\eta + \zeta_z w_\zeta) - (\xi_x u_\xi + \eta_x u_\eta + \zeta_x u_\zeta) - (\xi_y v_\xi + \eta_y v_\eta + \zeta_y v_\zeta)] \\ \tau_{xy} = \tau_{yx} &= \mu(\xi_y u_\xi + \eta_y u_\eta + \zeta_y u_\zeta + \xi_x v_\xi + \eta_x v_\eta + \zeta_x v_\zeta) \\ \tau_{yz} = \tau_{zy} &= \mu(\xi_z v_\xi + \eta_z v_\eta + \zeta_z v_\zeta + \xi_y w_\xi + \eta_y w_\eta + \zeta_y w_\zeta) \\ \tau_{xz} = \tau_{zx} &= \mu(\xi_z u_\xi + \eta_z u_\eta + \zeta_z u_\zeta + \xi_x w_\xi + \eta_x w_\eta + \zeta_x w_\zeta) \end{aligned} \quad (3.6)$$

and the heat flux terms are obtained by Fourier's law as

$$\begin{aligned} q_x &= -k(\xi_x T_\xi + \eta_x T_\eta + \zeta_x T_\zeta) \\ q_y &= -k(\xi_y T_\xi + \eta_y T_\eta + \zeta_y T_\zeta) \\ q_z &= -k(\xi_z T_\xi + \eta_z T_\eta + \zeta_z T_\zeta) \end{aligned} \quad (3.7)$$

where  $\mu$  is the dynamic viscosity and  $k$  is the thermal conductivity. They can both be calculated with Sutherland's formulas given by

$$\begin{aligned} \mu &= C_1 \frac{T^{3/2}}{T + C_2} \\ k &= C_3 \frac{T^{3/2}}{T + C_4} \end{aligned} \quad (3.8)$$

where at moderate temperatures, the constants are given as  $C_1 = 1.458 \times 10^{-6} kg/(m \cdot s \cdot \sqrt{K})$ ,  $C_2 = 110.4K$ ,  $C_3 = 2.495 \times 10^{-3} (kg \cdot m)/(s^3 \cdot K^{3/2})$ , and  $C_4 = 194.0K$ .

In the NLDE approach, the instantaneous flow quantities are calculated as the sum of a mean and resolved perturbation component,

$$\begin{aligned}
\rho &= \rho_0 + \rho' \\
u &= u_0 + u' \\
v &= v_0 + v' \\
w &= w_0 + w' \\
e &= e_0 + e' \\
p &= p_0 + p'
\end{aligned} \tag{3.9}$$

where the mean quantities are known from either a RANS solution or an assumed uniform flow, and the perturbation quantities are updated at each Runge-Kutta substep. After decomposition, the instantaneous flow vector may be written as

$$\mathbf{q} = \mathbf{q}_0 + \mathbf{q}' = \begin{Bmatrix} \rho_0 \\ \rho_0 u_0 \\ \rho_0 v_0 \\ \rho_0 w_0 \\ e_0 \end{Bmatrix} + \begin{Bmatrix} \rho' \\ \rho' u_0 + \rho_0 u' + \rho' u' \\ \rho' v_0 + \rho_0 v' + \rho' v' \\ \rho' w_0 + \rho_0 w' + \rho' w' \\ e' \end{Bmatrix} \tag{3.10}$$

and Eq. (3.2) can be written as

$$\begin{aligned}
\frac{\partial}{\partial t} \left( \frac{\mathbf{q}'}{J} \right) &= - \frac{\partial}{\partial \xi} \left( \frac{1}{J} [\xi_x \mathbf{E} + \xi_y \mathbf{F} + \xi_z \mathbf{G}] \right) \\
&\quad - \frac{\partial}{\partial \eta} \left( \frac{1}{J} [\eta_x \mathbf{E} + \eta_y \mathbf{F} + \eta_z \mathbf{G}] \right) \\
&\quad - \frac{\partial}{\partial \zeta} \left( \frac{1}{J} [\zeta_x \mathbf{E} + \zeta_y \mathbf{F} + \zeta_z \mathbf{G}] \right) - \frac{\partial}{\partial t} \left( \frac{\mathbf{q}_0}{J} \right)
\end{aligned} \tag{3.11}$$

If the mean component of the flow is assumed steady, the mean flow time derivative on the right side of Eqn. (3.11) is zero. For large scale trailing edge broadband noise prediction, it will be assumed that the mean flow time scale is much larger as compared to the very small time over which the NLDE are solved for the turbulence, and this term can be neglected. The objective is to solve for the perturbation variables in a time accurate fashion. In this NLDE implementation, the perturbation flow vector  $\mathbf{q}'$  is integrated explicitly in time with a Runge-Kutta method as illustrated in Figure 3.2. At each Runge-Kutta step, the perturbation components in the instantaneous flux vectors are calculated from

$$\begin{aligned}
\rho' &= q'(1) \\
u' &= (q'(2) - \rho' u_0) / \rho \\
v' &= (q'(3) - \rho' v_0) / \rho \\
w' &= (q'(4) - \rho' w_0) / \rho \\
e' &= q'(5)
\end{aligned} \tag{3.12}$$

and the fluctuating pressure, which is used for noise prediction, can be found from

$$\begin{aligned}
p' &= (\gamma - 1) [e' - \rho(u'u_0 + v'v_0 + w'w_0) \\
&\quad - \frac{1}{2}\rho(u'^2 + v'^2 + w'^2) \\
&\quad - \frac{1}{2}\rho'(u_0^2 + v_0^2 + w_0^2)]
\end{aligned} \tag{3.13}$$

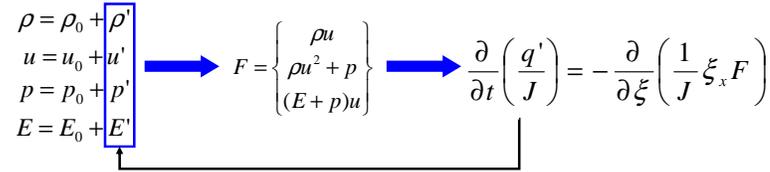


Figure 3.2: NLDE integration procedure (1-D for simplicity).

The NLDE were first used to predict high-speed axisymmetric jet noise radiation. Three dimensional validation cases were presented, along with jet noise calculations under different inlet conditions which agreed reasonably well with experimental data[24]. Later, these simulations included subgrid-scale models [23], demonstrating that three dimensional jet noise problems may be solved accurately for large-scale perturbations. Long derived the Perturbed Nonconservative Non-linear Euler (PENNE) equations which are similar to the NLDE and used them to predict acoustic radiation and scattering by bluff bodies[17]. All of the works listed above have been either for inviscid wall bounded flows, or viscous non wall bounded flows. Chyczewski *et al.* used the NLDE to predict wall-bounded turbulence statistics using LES [10].

## 3.2 NLDE Flow Solver

A CFD code was written which solves the NLDE system given in Section 3.1. The solver reads a RANS solution to be used as the mean flow, or a uniform flow can be specified. For code development, a uniform mean flow has been chosen to minimize the number of iterations required to achieve quasi-steady state shedding in circular cylinder test cases[12]. The user manual for the flow solver is located in Appendix C. The following is a brief summary of the code:

- Compressible, 3D structured grid Navier-Stokes solver
- Written in the Fortran 90 language
- MPI (Message Passing Interface) parallel code
- Code structure allows for easy addition and removal of features
- Boundary conditions tailored for CAA
- 4<sup>th</sup> order accurate, 5 stage LDDRK time integration (Low-Dissipation and Dispersion Runge-Kutta)
- 4<sup>th</sup> order accurate DRP finite differencing (Dispersion-Relation-Preserving)
- Explicit low pass filtering

### 3.2.1 Runge-Kutta Time Integration

A 5 stage Low Dispersion and Dissipation Runge-Kutta (LDDRK) method is used for time integration. The scheme is fourth order accurate and the coefficients are optimized to minimize the dissipation and dispersion errors of typical Runge-Kutta time marching schemes [13]. The governing equations are cast into the form

$$\frac{\partial \mathbf{q}'}{\partial t} = R(\mathbf{q}', \mathbf{q}_0) \quad (3.14)$$

where  $\mathbf{q}_0$  remains unchanged through the scheme and  $\mathbf{q}'$  is updated and marched through the scheme. To compute the value of  $\mathbf{q}'^{(n+1)}$  when the current solution  $\mathbf{q}'^{(n)}$  is known, for  $i = 1 \dots 5$ , compute (with  $\bar{\beta}_1 = 0$ )

$$\mathbf{K}_i = \Delta t R(\mathbf{q}'^{(n)} + \bar{\beta}_i \mathbf{K}_{i-1}, \mathbf{q}_0) \quad (3.15)$$

$$\mathbf{q}'^{(n+1)} = \mathbf{q}'^{(n)} + \mathbf{K}_5 \quad (3.16)$$

where  $\mathbf{q}'^{(n)}$  is the perturbation flow vector at time step  $n$ , and  $\bar{\beta}_i$  are the weights of the LDDRK scheme. Using the coefficients given in Table 3.1, the weights are calculated as

$$\begin{aligned} \bar{\beta}_5 &= c_2 \\ \bar{\beta}_4 &= c_3 / \bar{\beta}_5 \\ \bar{\beta}_3 &= c_4 / (\bar{\beta}_5 \bar{\beta}_4) \\ \bar{\beta}_2 &= c_5 / (\bar{\beta}_5 \bar{\beta}_4 \bar{\beta}_3) \end{aligned} \quad (3.17)$$

Table 3.1: The coefficients of the LDDRK scheme [13].

$c_1$	$c_2$	$c_3$	$c_4$	$c_5$
1	1/2	0.166558	0.0395041	0.00781071

### 3.2.2 Time Step Stability

The maximum local time step allowable for compressible solvers is dependent on the local speed of sound and grid spacing. For acoustic solvers, a time accurate solution is necessary and therefore the entire domain must be marched explicitly at the same time step. In addition, the fine grid spacing in the viscous wall region reduces the allowable time step for the entire domain. It is desirable to take the maximum allowable time step without violating the stability of the governing equations. The stability criteria selected for the present solver is based on the Courant-Freidrichs-Lewy (CFL) constraint outlined by Shieh [27] as

$$\Delta t < \frac{CFL}{\lambda_\xi + \lambda_\eta + \lambda_\zeta} \quad (3.18)$$

where CFL is most commonly selected to be around 1.0.  $\lambda_\xi$ ,  $\lambda_\eta$ , and  $\lambda_\zeta$  are the eigenvalues of the generalized coordinate transform defined as

$$\begin{aligned}
\lambda_\xi &= |U| + C\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2} \\
\lambda_\eta &= |V| + C\sqrt{\eta_x^2 + \eta_y^2 + \eta_z^2} \\
\lambda_\zeta &= |W| + C\sqrt{\zeta_x^2 + \zeta_y^2 + \zeta_z^2}
\end{aligned} \tag{3.19}$$

where  $C$  is the local sound speed and  $U$ ,  $V$ , and  $W$  are the contravariant velocities (velocities in the transformed uniform coordinate system) given as

$$\begin{aligned}
U &= \xi_x u + \xi_y v + \xi_z w \\
V &= \eta_x u + \eta_y v + \eta_z w \\
W &= \zeta_x u + \zeta_y v + \zeta_z w
\end{aligned} \tag{3.20}$$

### 3.2.3 Dispersion Relation Preserving Finite Differencing

A differencing scheme with properly selected weighting coefficients can greatly improve the dispersion characteristics of wave propagation. Tam and Webb [31] outlined a method of choosing the coefficients such that the spectral characteristics of discretized system match closely to the continuous system, which will be referred to as Dispersion-Relation-Preserving (DRP).

A method of calculating partial derivatives can be given as

$$\left(\frac{\partial f}{\partial x}\right)_t \simeq \frac{1}{\Delta x} \sum_{j=-k}^k a_j f_{i+j} \tag{3.21}$$

where  $a_j$  are the weighting coefficients of each point in the stencil,  $f_{i+j}$  is the scalar field value at stencil point  $i + j$ ,  $\Delta x$  is assumed constant, and  $k$  governs the size of the stencil. The weighting coefficients are typically calculated based on a Taylor series expansion and collecting the same powers of  $\Delta x$ . However, Tam and Webb suggest that they can be calculated by requiring that the Fourier transform of both sides of Equation 3.21 be equal. The detailed analysis is given by the authors. The coefficients for central difference DRP scheme denoted by  $a_j^{nm}$  can be found in Table 3.2.  $n$  and  $m$  are the number of points to the left and right of the point where the derivative is being calculated ( $j = 0$ ). Forward and backward

stencil coefficients are related noting that  $a_j^{nm} = -a_{-j}^{mn}$ .

Table 3.2: The coefficients of the DRP scheme [28].

$n = 3, m = 3$	$n = 0, m = 6$
$a_{-3} = -0.208431427703$	$a_0 = -2.192280339$
$a_{-2} = 0.166705904415$	$a_1 = 4.748611401$
$a_{-1} = -0.770882380518$	$a_2 = -5.108851914$
$a_0 = 0.0$	$a_3 = 4.461567104$
$a_1 = -a_{-1}$	$a_4 = -2.833498741$
$a_2 = -a_{-2}$	$a_5 = 1.128328861$
$a_3 = -a_{-3}$	$a_6 = -0.203876371$
$n = 1, m = 5$	$n = 2, m = 4$
$a_{-1} = -0.209337622$	$a_{-2} = 0.049041958$
$a_0 = -1.084875676$	$a_{-1} = -0.468840357$
$a_1 = 2.147776050$	$a_0 = -0.474760914$
$a_2 = -1.388928322$	$a_1 = 1.273274737$
$a_3 = 0.768949766$	$a_2 = -0.518484526$
$a_4 = -0.281814650$	$a_3 = 0.166138533$
$a_5 = 0.048230454$	$a_4 = -0.026369431$

### 3.2.4 Grid Metrics

Metrics need to be calculated for each grid point in the computational domain. They are calculated numerically from the corresponding physical domain based on a fourth order Dispersion Relation Preserving scheme [31], the same used during integration. The metrics are calculated from the physical domain at time  $t = 0$  and are stored in the computational domain. At any time step in the integration, the updated metrics are obtained by multiplying the change of base rotation matrix by the metrics matrix:

$$[M]_{new} = [COB] [M]_{old} \quad (3.22)$$

where  $[COB]$  is the standard rotation matrix from the old frame to the new frame, and  $[M]$  is the metrics matrix calculated at  $t = 0$  of the form

$$[M] = \begin{vmatrix} \xi_x & \eta_x & \zeta_x \\ \xi_y & \eta_y & \zeta_y \\ \xi_z & \eta_z & \zeta_z \end{vmatrix} \quad (3.23)$$

A freestream preservation technique used by Thomas and Lombard [33] has been adopted for the metric calculations. Errors in proper metric cancelation arise in the finite difference discretizations of the governing equations. For example, for the x-components of the surface area vectors can be written as

$$\begin{aligned} \xi_x &= y_\eta z_\zeta - y_\zeta z_\eta \\ \eta_x &= y_\zeta z_\xi - y_\xi z_\zeta \\ \zeta_x &= y_\xi z_\eta - y_\eta z_\xi \end{aligned} \quad (3.24)$$

Thomas and Lombard rewrite the expressions prior to discretization in the equivalent “conservative” form as,

$$\begin{aligned} \xi_x &= (y_\eta z)_\zeta - (y_\zeta z)_\eta \\ \eta_x &= (y_\zeta z)_\xi - (y_\xi z)_\zeta \\ \zeta_x &= (y_\xi z)_\eta - (y_\eta z)_\xi \end{aligned} \quad (3.25)$$

Visbal and Gaitonde [35] performed a study using this metric formulation on a highly curved mesh. They found that the alternative method results in dramatic reduction of metric cancelation errors (of order  $10^{-11}$ ) and guarantees freestream preservation to an acceptable level. The grid metrics are calculated numerically at each grid point. The  $x$ ,  $y$ , and  $z$  coordinates are used as scalar variables in the DRP differencing scheme. For example, to calculate  $y_\eta$ , the  $y$  coordinates of the grid would be used in the DRP scheme assuming a constant spacing in the  $\eta$  direction.

### 3.2.5 Boundary Conditions

A detail to consider when solving the NLDE is the application of boundary conditions. If the mean flow is obtained from a RANS solution, it is assumed that the mean flow inherently obeys all wall and farfield boundary conditions. This allows the boundary conditions during the NLDE solution to be applied only to the per-

turbation variables using wall and radiation outflow conditions used specifically for aeroacoustic applications [12].

### 3.2.5.1 Radiation Boundary Conditions

These boundary conditions are adopted from Tam and Webb [31]. They are applied to every farfield boundary and allow any combination of pressure, vorticity, and entropy waves to enter or exit the domain. For subsonic cases, all inflow boundaries may be classified as a radiation boundary, where the interior of the computational domain only affects the boundary through pressure perturbations. The radiation conditions of Tam and Webb may be written as

$$\left( \frac{1}{V(r, \theta)} \frac{\partial}{\partial t} + \frac{\partial}{\partial r} + \frac{1}{2r} \right) \begin{Bmatrix} \rho' \\ u' \\ v' \\ w' \\ p' \end{Bmatrix} = 0 \quad (3.26)$$

Here it is understood that if the source of the disturbances are close to the origin (center) of the computational domain, Equation 3.26 can be cast into a form useful to rectangular coordinates. Tam and Webb report that the performance of the boundary conditions is insensitive to the source of the perturbations and origin, as long as the source is not very close to the boundary. With this in mind, the polar derivative can be rewritten as

$$\frac{\partial}{\partial r} = \left( x \frac{\partial}{\partial x} + y \frac{\partial}{\partial y} + z \frac{\partial}{\partial z} \right) / r \quad (3.27)$$

where  $r$  is the distance from the origin to the boundary point defined as  $r = \sqrt{x^2 + y^2 + z^2}$ . The scalar  $V$  is defined as

$$V(r, \theta) = \mathbf{V}_0 \cdot \hat{\mathbf{r}} + \sqrt{C_0^2 - (\mathbf{V}_0 \cdot \hat{\theta})^2 - (\mathbf{V}_0 \cdot \hat{\phi})^2} \quad (3.28)$$

where  $C_0$  is the ambient speed of sound, and  $\hat{\mathbf{r}}$  is the unit vector in the radial direction. The tangential unit normals can be calculated as

$$\begin{aligned}\hat{\theta} &= \begin{Bmatrix} \cos \theta \cos \phi \\ \cos \theta \sin \phi \\ -\sin \theta \end{Bmatrix} \\ \hat{\phi} &= \begin{Bmatrix} -\sin \phi \\ \cos \phi \\ 0 \end{Bmatrix}\end{aligned}\tag{3.29}$$

where  $\theta = \tan^{-1}(y/x)$  and  $\phi = \cos^{-1}(z/r)$ .

Outflow boundaries may experience any combination of pressure, vorticity, and entropy disturbances. The boundary condition for pressure described above remains the same, however the density and velocity equations are now

$$\begin{aligned}\frac{\partial \rho'}{\partial t} + \mathbf{V}_0 \cdot \nabla(\rho') &= \frac{1}{C_0^2} \left( \frac{\partial p'}{\partial t} + \mathbf{V}_0 \cdot \nabla(p') \right) \\ \frac{\partial u'}{\partial t} + \mathbf{V}_0 \cdot \nabla(u') &= -\frac{1}{\rho_0} \frac{\partial p'}{\partial x} \\ \frac{\partial v'}{\partial t} + \mathbf{V}_0 \cdot \nabla(v') &= -\frac{1}{\rho_0} \frac{\partial p'}{\partial y} \\ \frac{\partial w'}{\partial t} + \mathbf{V}_0 \cdot \nabla(w') &= -\frac{1}{\rho_0} \frac{\partial p'}{\partial z}\end{aligned}\tag{3.30}$$

The residuals are calculated from Equations 3.26 and 3.30 and are integrated within the five stage Runge-Kutta timestepping. The spacial derivatives are taken using biased seven point Dispersion Relation Preserving finite differences, which is the same scheme as the domain interior. The mean flow is assumed to always satisfy radiation boundary conditions, so these only need to be applied to the perturbation variables.

### 3.2.5.2 No-Slip Adiabatic Wall

Body surfaces are given no-slip, adiabatic wall boundary conditions. This enforces zero fluid velocity relative to the blade surface velocity, and zero pressure and density gradient in the normal direction to the blade surface. In all cases with a solid wall boundary, the grid has three ghost cell layers beneath the blade surface layer as seen in Figure 3.3. The three ghost layers and the surface layer are specified to be zero fluid velocity relative to the surface velocity. In addition, zero pressure

and density gradient are enforced by explicitly specifying the value at ghost point  $j$  equal to the value of physical point  $8 - j$ .

The mean flow from a RANS solution is assumed to already satisfy the wall boundary conditions, so if using a RANS solution, the treatment described above is satisfactory if applied only to the perturbation variables. However, if the mean flow is assumed uniform, the mean flow velocity violates the no-slip condition as seen in Figure 3.4. In this case, the perturbation velocities must be specified to correct the mean flow such that the instantaneous flow satisfies the no-slip condition.

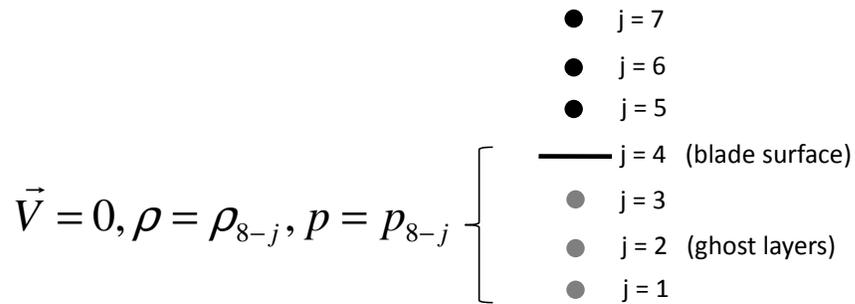


Figure 3.3: No-slip adiabatic wall boundary treatment.

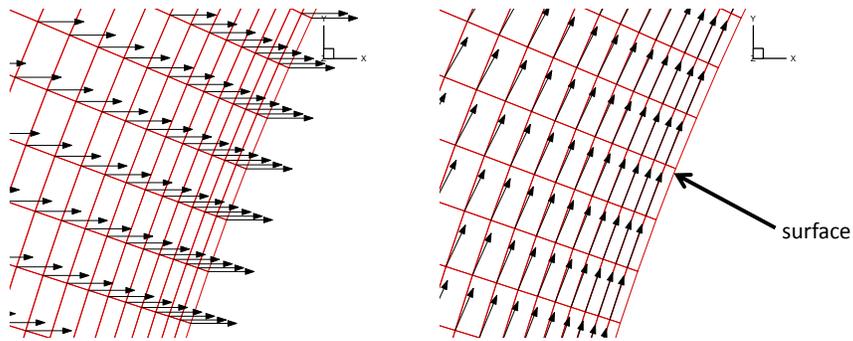


Figure 3.4: Instantaneous velocity vectors with uniform rightward mean flow. Uniform flow violates no slip condition at surface (left). After no-slip condition is applied to perturbation variables, instantaneous flow is corrected (right).

### 3.2.6 Low Pass Filtering

The least possible damping is ideal for aeroacoustic computations when propagating acoustic waves, so low pass filtering (LPF) is used to prevent the growth of high wave number modes instead of artificial viscosity. A LPF technique is adopted from Visbal and Gaitonde [35] that filters the solution at each level through the computational domain. If  $\phi$  is a solution vector for one of the conserved variables along a 1-D line of length  $N$ , then the filtered values  $\hat{\phi}$  are obtained by solving the tridiagonal system

$$\alpha\hat{\phi}_{i-1} + \hat{\phi}_i + \alpha\hat{\phi}_{i+1} = \sum_{n=0}^3 \frac{a_n}{2} (\phi_{i+n} + \phi_{i-n}) \quad (3.31)$$

which provides a sixth order filter on a seven point stencil. The coefficients are derived in terms of  $\alpha$  with Taylor and Fourier series analysis (given below), and acceptable values for the adjustable parameter  $\alpha$  satisfy the inequality  $-0.5 < \alpha \leq 0.5$ . For most cases, values of  $\alpha$  between 0.3 and 0.5 are appropriate. The frequency of application can vary. Visbal and Gaitonde suggest after each Runge-Kutta sub-iteration for implicit algorithms, or once after the final Runge-Kutta stage for explicit algorithms.

Coefficients for points  $n = 4$  through  $N - 3$  are given in Table 3.3. The values

Table 3.3: Low pass filter coefficients for points 4 through  $N-3$ .

$a_0$	$a_1$	$a_2$	$a_3$
$\frac{11}{16} + \frac{5\alpha}{8}$	$\frac{15}{32} + \frac{17\alpha}{16}$	$-\frac{3}{16} + \frac{3\alpha}{8}$	$\frac{1}{32} - \frac{\alpha}{16}$

at the endpoints 1 and  $N$  are not filtered. The boundary points 2, 3,  $N-1$ , and  $N-2$  are given special treatment with one-sided formulas.

$$\alpha\hat{\phi}_{i-1} + \hat{\phi}_i + \alpha\hat{\phi}_{i+1} = \sum_{n=1}^7 a_{n,i}\phi_n, i \in 2, 3 \quad (3.32)$$

$$\alpha\hat{\phi}_{i-1} + \hat{\phi}_i + \alpha\hat{\phi}_{i+1} = \sum_{n=0}^6 a_{N-n,i}\phi_{N-n}, i \in N-2, N-1$$

The right boundary values for  $a$  are found by noting that  $a_{N-n,i} = a_{n+1,N-i+1}$ . The values of  $a$  for points 2,  $N - 1$ , 3, and  $N - 2$  are given in Tables 3.4 and 3.5.

Table 3.4: Low pass filter coefficients for points 2 and  $N-1$ .

$a_{1,2}$	$a_{2,2}$	$a_{3,2}$	$a_{4,2}$	$a_{5,2}$	$a_{6,2}$	$a_{7,2}$
$\frac{1}{64} + \frac{31\alpha}{32}$	$\frac{29}{32} + \frac{3\alpha}{16}$	$\frac{15}{64} + \frac{17\alpha}{32}$	$-\frac{5}{16} + \frac{5\alpha}{8}$	$\frac{15}{64} - \frac{15\alpha}{32}$	$-\frac{3}{32} + \frac{3\alpha}{16}$	$\frac{1}{64} - \frac{\alpha}{32}$

Table 3.5: Low pass filter coefficients for points 3 and  $N-2$ .

$a_{1,3}$	$a_{2,3}$	$a_{3,3}$	$a_{4,3}$	$a_{5,3}$	$a_{6,3}$	$a_{7,3}$
$-\frac{1}{64} + \frac{\alpha}{32}$	$\frac{3}{32} + \frac{13\alpha}{16}$	$\frac{49}{64} + \frac{15\alpha}{32}$	$\frac{5}{16} + \frac{3\alpha}{8}$	$-\frac{15}{64} + \frac{15\alpha}{32}$	$\frac{3}{32} - \frac{3\alpha}{16}$	$-\frac{1}{64} + \frac{\alpha}{32}$

### 3.2.7 Message Passing Interface Parallelization

The NLDE code is parallelized using the Message Passing Interface (MPI). Parallelizing the code means that multiple processors can work on individual portions of the domain simultaneously, reducing the amount of time required for a solution. At every Runge-Kutta substep, adjacent processors share information through send and receive routines built into the MPI libraries.

At the beginning of code execution, the master processor reads the input settings and the entire domain grid. The master then divides the total domain into subdomains for each child processor, maintaining a balanced computational load among the subdomains. The domains are divided into  $i$  and  $j$  sized blocks, and each subdomain solves the entire range of  $k$  indices. The master processor also solves a portion of the domain, but whenever possible, it assigns itself the smallest portion to compensate for its own input and output tasks. Figure 3.5 is an example of how a computational domain can be divided by the NLDE code. In the example, there are 16 available processors to solve a domain of size  $201 \times 201 \times 7$ . Processor 1 is the master processor and processors 2 through 16 are the children. Because the domain is square, each subdomain in this example only solves  $1/16$  of the entire domain, which should reduce the runtime by a factor of 16 if communication time

is negligible. Each processor, therefore, only solves a domain size of approximately  $50 \times 50 \times 7$ . Because of the seven point DRP differencing stencil, each subdomain sends and receives three rows of flow variables to adjacent domains to ensure the use of central differencing at domain interfaces. Each processor exchanges both the primitive flow variables and residual calculation at each Runge-Kutta substep. In a 2-D validation case described in Section 4.1, ideal speedup was achieved when using up to 64 Intel EM64T processors. Thirty-two computing nodes were used, each with dual 3.6GHz processors and 6 GB of RAM.

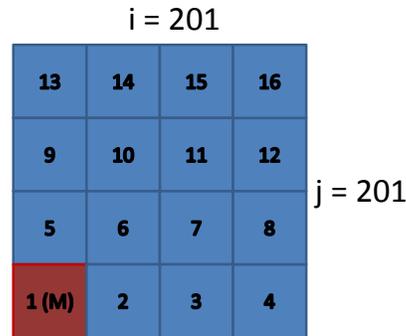


Figure 3.5: Domain decomposition for MPI parallelization. Master processor (red) and child processors (blue).

### 3.3 Validation of the NLDE Code

The NLDE flow solver is a new code which must be validated. The code must accurately predict acoustic wave propagation as well as apply accurate boundary conditions. The following two chapters outline the methods used to validate and test the NLDE flow solver. Chapter 4 validates the linear properties of the code with a series of Gaussian pulse propagation problems. Chapter 5 tests the nonlinear features with circular cylinder shedding and airfoil flow problems.

# Chapter 4

## NLDE Flow Solver Validation

Two and three dimensional validation problems have been solved using the present NLDE solver on a variety of computer clusters. The first test cases are two dimensional acoustic pulses in both stationary and moving media. These cases validate the code's spatial and temporal accuracy. The second set of test cases are outlined by Tam and Dong [29] and are designed to test the solid wall reflective boundary conditions in the code. The third test case is a three dimensional spherical pulse in a stationary medium designed to test the code's three dimensional accuracy. Lastly, the scattering of a Gaussian pulse by a circular cylinder will be calculated. All Gaussian pulses used in the validation are of the form

$$\begin{aligned}\rho' &= A\rho_0 e^{[-\ln(2)(x^2+y^2+z^2)/w^2]} \\ p' &= Bp_0 e^{[-\ln(2)(x^2+y^2+z^2)/w^2]}\end{aligned}\tag{4.1}$$

where  $A$  and  $B$  are scaling constants and  $w$  is defined as the half width of the Gaussian pulse.

### 4.1 2-D Gaussian Pulse Propagation

An exact linearized Euler solution of the propagation of a Gaussian pulse is presented by Tam and Webb [31]. Although this case does not test the nonlinearity of the code, it does test for spatial and temporal accuracy. A Gaussian pulse of the form  $A = 0.01$ ,  $B = 0.01$ , and  $w = 3$  is imposed as an initial condition on the

perturbation solution at time  $t = 0$ . Figure 4.1 is a contour and line plot of the initial Gaussian pulse, which is homogeneous in the  $z$  direction because the case is two dimensional. The wavefront of the acoustic pulse expands radially outward. The case was run with both stationary and  $M = 0.5$  background flow velocity in the positive  $x$  direction, with constant density of  $\rho_0 = 1.2 \text{ kg/m}^3$ , pressure of  $p_0 = 101 \text{ kPa}$ , and sound speed of  $C_0 = 343.269 \text{ m/s}$ .

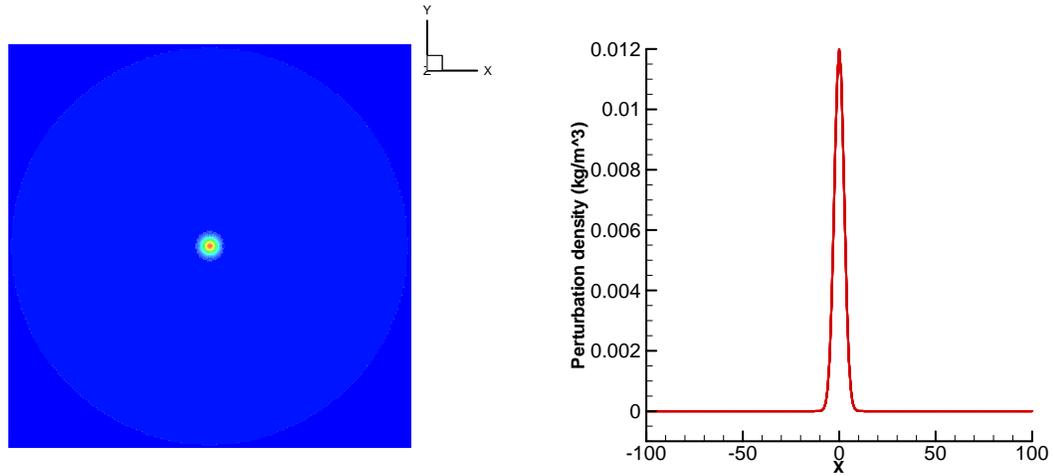


Figure 4.1: Initial Gaussian pulse distribution imposed on perturbation density at time  $t = 0$ . Contour of domain (left), Line plot along  $y = 0$  (right).

The grid dimensions are  $201 \times 201 \times 7$ , with bounds of  $-100 \leq x \leq 100$ ,  $-100 \leq y \leq 100$ . The grid is seven layers deep to allow for two dimensional cases to be performed with the three dimensional NLDE code. A CFL number of 0.5 was used. Radiation boundary conditions were applied to all boundaries. Figure 4.2 compares the NLDE solution with the exact solution at four different times for stationary background flow. Both the contour and amplitudes of the wavefronts agree quite well, and the pulse exits the domain with negligible reflections, demonstrating the effectiveness of the Tam and Webb radiation boundary conditions. Figure 4.3 compares the NLDE solution with the exact solution with  $M = 0.5$  background flow. Again, the magnitude and location of the wavefronts agree. The rightward wave travels at the sound speed plus the background flow velocity, and the leftward wave travels at the sound speed minus the background flow velocity.

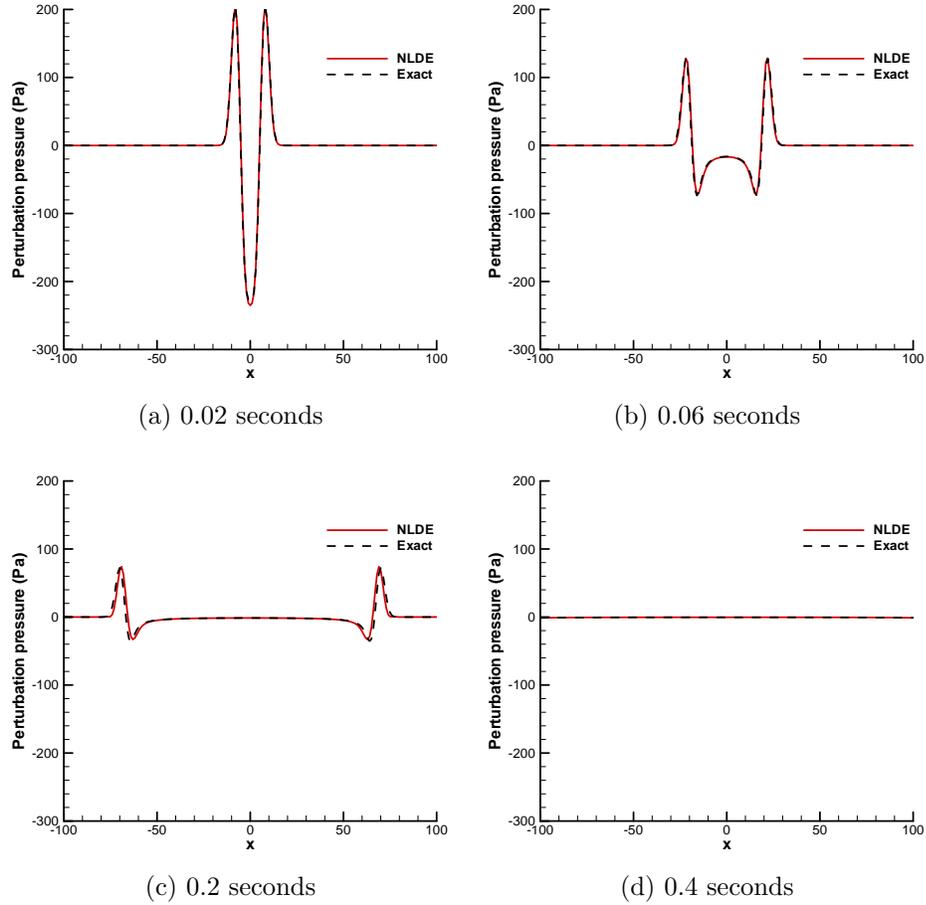


Figure 4.2: Validation of 2-D Gaussian pulse propagation in a stationary medium.

## 4.2 3-D Gaussian Pulse Propagation

A three dimensional Gaussian pulse validates the three dimensional accuracy of the code. The grid dimensions are  $61 \times 61 \times 61$ , with bounds of  $-30 \leq x \leq 30$ ,  $-30 \leq y \leq 30$ ,  $-30 \leq z \leq 30$ . Again the pulse is of the form  $A = 0.01$ ,  $B = 0.01$ , and  $w = 3$ . Two hundred timesteps were taken with a CFL number of 0.5. Figure 4.4 shows a comparison of the NLDE results with the exact solution with zero background flow. Both wave magnitude and speed agree very well with the exact solution. The pulse exits the domain with negligible reflections, demonstrating the effectiveness of the Tam and Webb radiation boundary conditions in three dimensions.

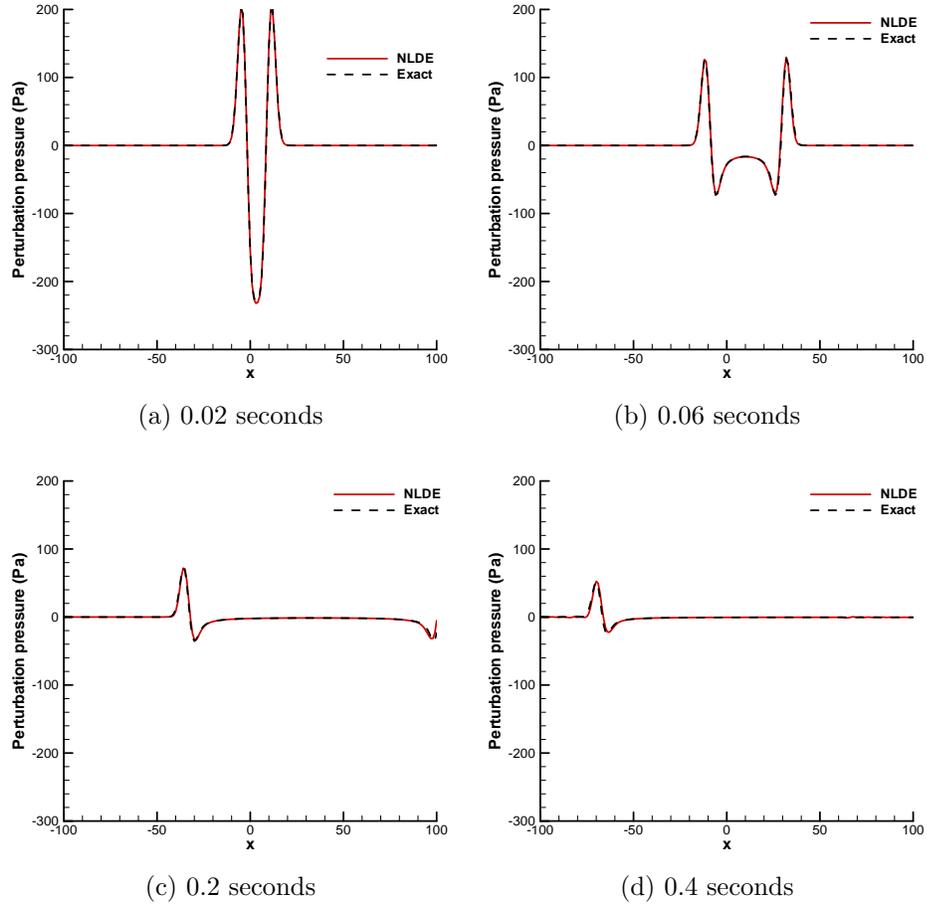


Figure 4.3: Validation of 2-D Gaussian pulse propagation in  $M = 0.5$  medium (rightward).

### 4.3 2-D Solid Wall Reflection

The same pulse was used by Tam and Dong [29] to validate solid wall reflective boundary conditions. The grid dimensions are  $151 \times 76 \times 7$ , and the grid bounds are  $-75 \leq x \leq 75$ , and  $0 \leq y \leq 75$ . The bottom of the domain ( $y = 0$ ) is specified to be an inviscid, adiabatic solid wall, while all others are specified to be radiation boundaries. The solution was run for 500 timesteps with a CFL of 0.5. Figure 4.5 is the NLDE result compared directly with the Tam and Dong result in a stationary medium. The NLDE solution compares well with Tam and Dong's solution, who validated with the analytical solution. Additional validation was performed in a medium with a background flow of  $M = 0.5$  in the positive  $x$  direction, and similar agreement was found.

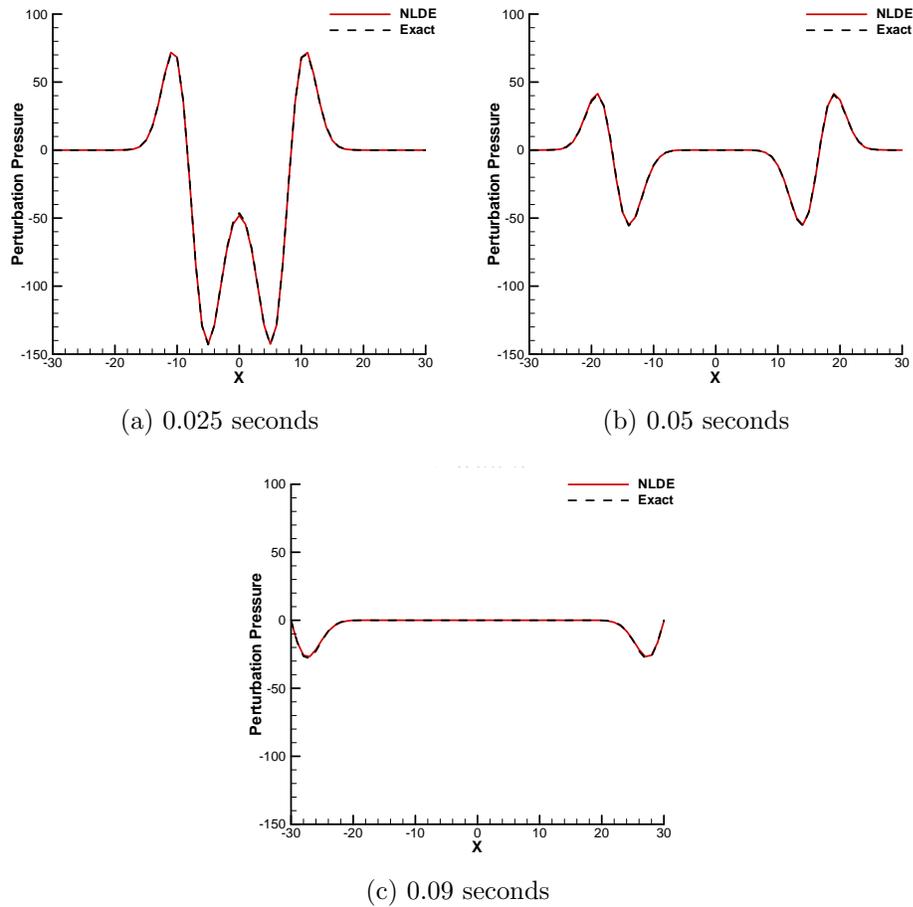
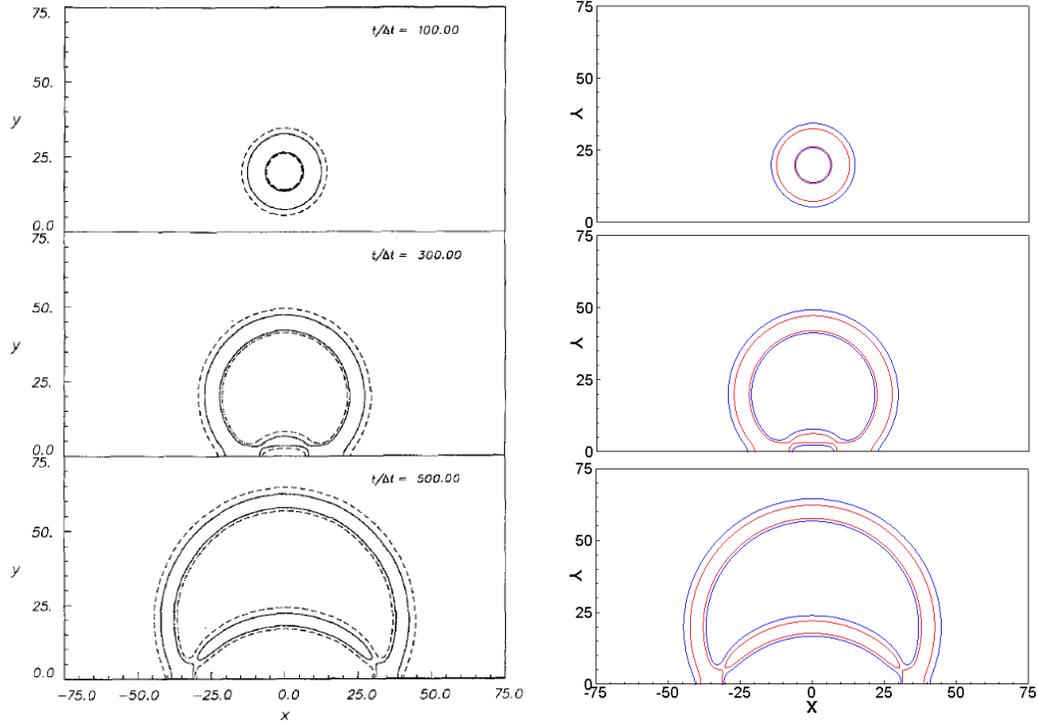


Figure 4.4: Validation of 3-D Gaussian pulse propagation in a stationary medium.

## 4.4 Scattering of a Gaussian Pulse by a Circular Cylinder

This validation case was taken from the Second Computational Aeroacoustics Workshop on Benchmark Problems [30] (Problem 2, Category 1). The case is selected to validate the propagation of acoustic waves and the implementation of solid wall and farfield boundary conditions. The case is illustrated in Figure 4.6a. The center of the two dimensional Gaussian pulse is located four cylinder diameters to the right of the cylinder center. For simplicity, the cylinder diameter was selected to be one. The observer points A, B, and C are located five diameters from the cylinder center. The angular positions are  $\theta = 90, 135, \text{ and } 180^\circ$ , respectively.



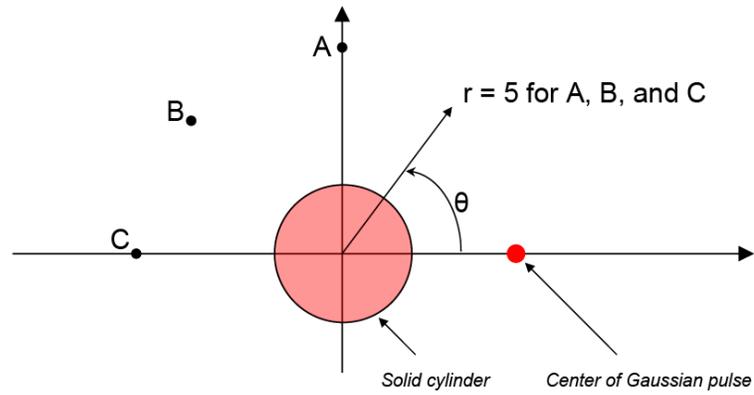
(a) Tam and Dong solution contours in a stationary medium [29]

(b) NLDE equivalent contours

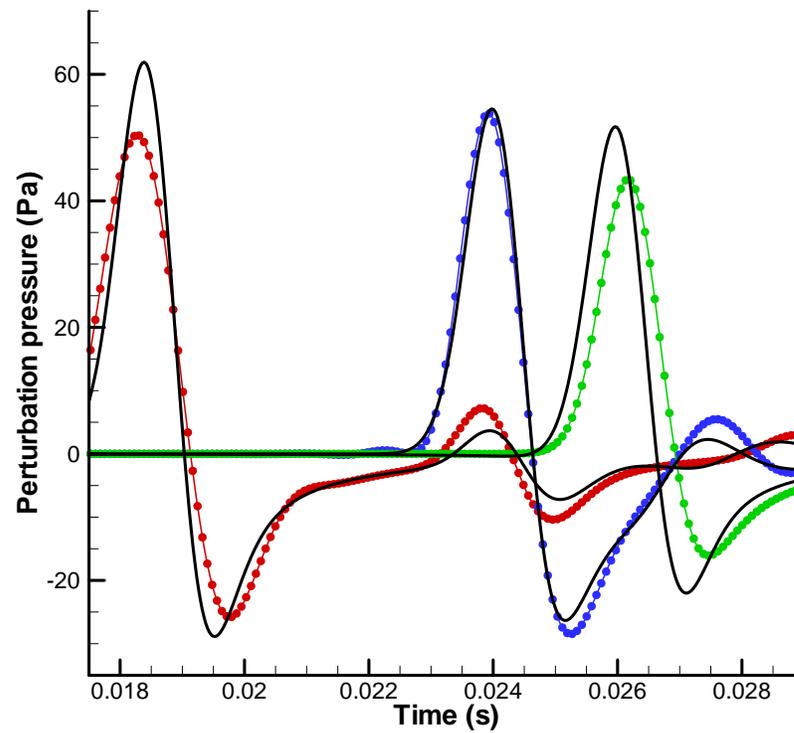
Figure 4.5: Validation of Tam and Dong solid wall reflection case in a stationary medium.

The initial Gaussian distribution is of the form  $A = 0.01$ ,  $B = 0.01$ , and  $w = 0.2$ , and is centered four units to the right of the cylinder.

There is zero background flow velocity for this problem. The cylinder surface is an adiabatic, inviscid wall, and the farfield radiation conditions of Tam and Webb are used. The grid contains only 200 and 250 uniformly spaced points in the radial and circumferential direction, respectively. The farfield boundary is located eight cylinder diameters from the cylinder surface. The analytical solution is provided by Kurbatskii in the workshop publication[30]. Figure 4.6b is a comparison of the NLDE results with the exact solution. The coarse grid that was used did not capture the exact magnitude of the reflected pulse at all locations, but the results show that the proper wave propagation speed was captured and that the curved solid wall boundary conditions are effective. It is recommended that further cases are run to study the effect of grid refinement on the solution.



(a) Scattering benchmark problem setup



(b) Comparison of exact solution with NLDE solution. Red - Point A; Blue - Point B; Green - Point C; Black - exact

Figure 4.6: Gaussian pulse scattering problem setup and results.

# Chapter 5

## Circular Cylinder and Airfoil Noise Prediction

### 5.1 Circular Cylinder Flow

The Gaussian pulse test cases do not test the nonlinear capability of the NLDE approach. Two and three dimensional circular cylinder flow cases were developed in order to evaluate the code's ability to model periodic shedding. In the following circular cylinder flow cases, uniform mean flow is used. To initialize the NLDE simulation, the instantaneous flow velocity at the cylinder surface is set to a no slip condition by setting the perturbation velocities to be equal and opposite to the mean flow velocity. The pressure and density of the mean flow is uniform, so the adiabatic condition is enforced by setting zero pressure and density gradient on the perturbation pressure and density only.

Two 2-D circular cylinder test cases were chosen to perform bluff body shedding similar to the work of Hansen *et al.* [12], and also to test the output acoustic data surfaces in the NLDE code for noise prediction. Both cylinders operate at  $Re_d \approx 90,000$  and  $M = 0.2$  mean flow. The mean density is  $1.2 \text{ kg/m}^3$  and the pressure is 101 kPa. The grids are O-grids and extend eight diameters into the farfield where radiation boundary conditions are applied. Grid one is the coarser of the two grids (near the cylinder surface) with 100 and 150 uniformly spaced points in the circumferential and radial direction. The wall grid spacing in the radial direction is 5% of the diameter. The second grid contains 301 points in the circumferential

direction and 65 points in the radial direction. Hyperbolic tangent grid spacing is used at the cylinder surface and the wall grid spacing is ten times more fine (0.5% of the diameter). Figure 5.1 shows the grid in proximity to the cylinder surface for both cylinder test cases.

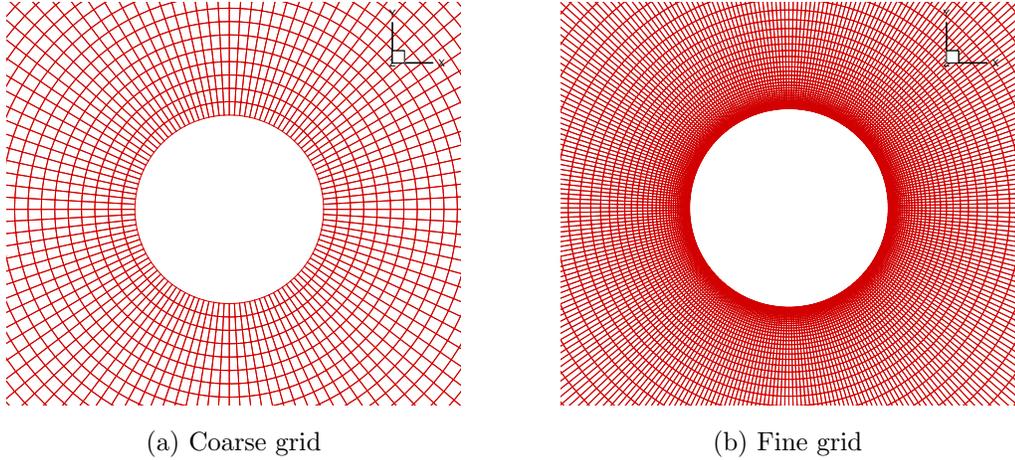


Figure 5.1: Two circular cylinder grids used to model shedding and develop acoustic data surfaces.

Permeable acoustic data surfaces (ADS) are used to extract pressure perturbations generated by the shedding flow behind the cylinders. The surfaces are located one diameter away from the cylinder wall in both cases and encircle the cylinder. A collection of PSU-WOPWOP observers were placed around both cylinders to show the directivity of the noise. They are distributed in a circle around the center of the cylinder with a radius of 128 cylinder diameters. Figure 5.2 shows the observer locations for both cylinder cases.

### 5.1.1 Results and Discussion

The coarse and fine resolution case were both run until quasi-steady shedding was observed. Laminar vortex shedding was achieved in both resolutions as seen in Figure 5.3. The no slip boundary condition applied to the perturbation variables in the NLDE triggers the unsteadiness, and the NLDE corrects the mean flow so that the instantaneous flow is the total physical solution. The coarse grid is actually more resolved than the fine grid in the farfield, due to the fine grid

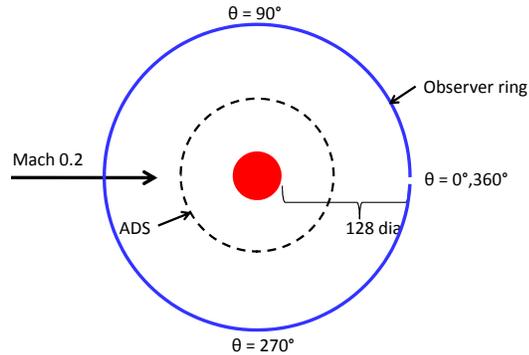


Figure 5.2: Two observer locations for circular cylinder shedding case (not to scale).

having significant grid stretching away from the cylinder surface. Consequently, the vortical structure behind the coarse cylinder grid is less damped than the fine cylinder grid. Although the differences between the two grids have not been examined quantitatively, a qualitative similarity in vortex shedding is observed.

The experimental work of Revell *et al.* [25] is used to assess the accuracy of the NLDE and PSU-WOPWOP acoustic predictions. The experiment consisted of a cylinder 26.3 diameters long, operating at  $Re_d = 89,000$  and  $M = 0.2$ . The observer was located in the  $90^\circ$  (cross flow) direction located 128 diameters from the surface. Although the current NLDE cases are 2-D, the overall directivity and SPL are expected to generally agree with the experimental work. All SPL results are referenced to  $20 \mu\text{Pa}$ .

In Figure 5.4a, the SPL results for the  $90^\circ$  observer are compared with the experimental work for both the fine and coarse grid. The 2-D NLDE solution greatly over predicts the shedding frequency for the coarse grid, but the SPL for both the shedding frequency and its first harmonic agree well with experimental data. In addition, the shedding frequency and its first harmonic for the fine grid agree fairly well with the experimental data. The directivity plot of SPL for the fine grid is shown in Figure 5.4b for the shedding frequency and its first harmonic. In the transverse direction, the largest SPL is found at the shedding frequency while in the wake direction, the dominant tone is from the drag dipole at two times shedding frequency.

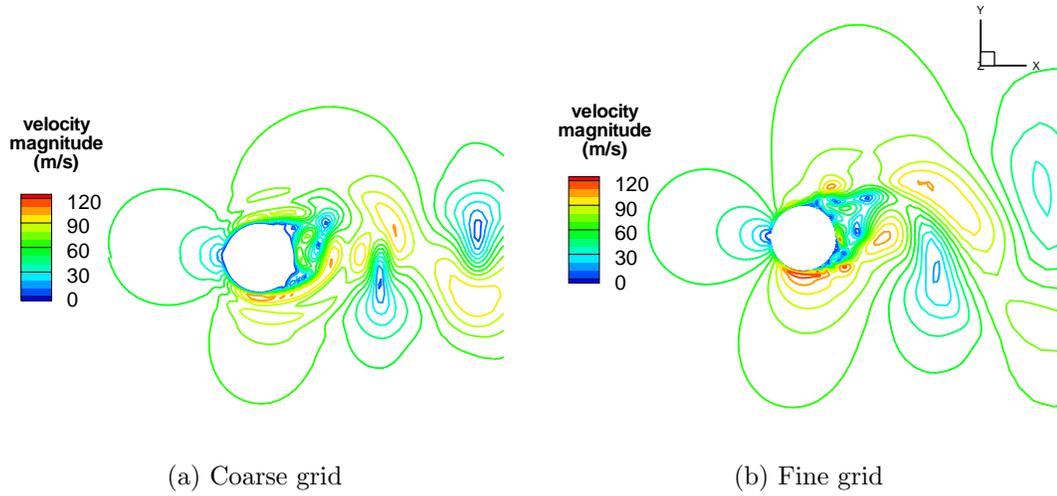


Figure 5.3: Contours of velocity magnitude for the coarse and fine grid cylinder

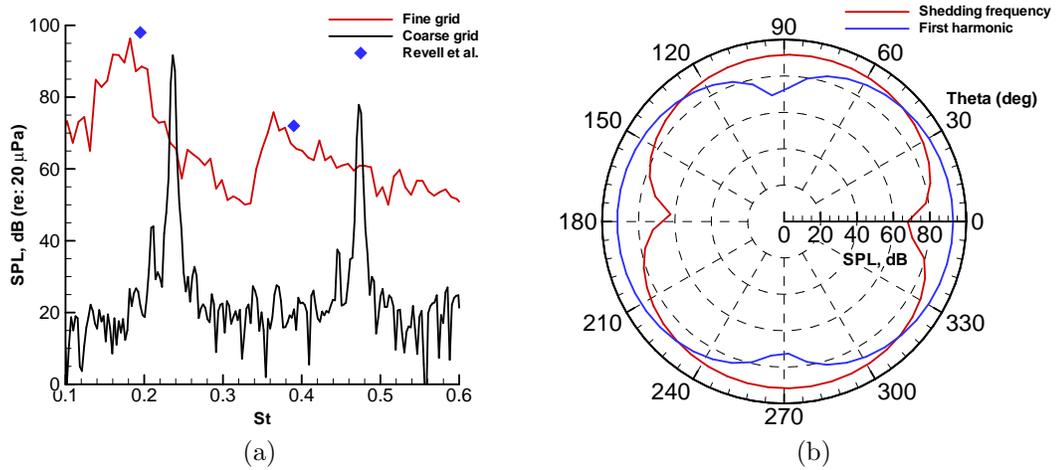


Figure 5.4: SPL results of cylinder with NLDE - PSU-WOPWOP prediction (a) compared with experimental data in the transverse direction [25] and (b) directivity plots for the predicted shedding frequency and its first harmonic for the fine cylinder

## 5.2 Airfoil Noise Prediction

The NLDE code has been tested using an NACA 0012 airfoil section and a BSDS flatback wind turbine blade section. The benefits of the flatback blade section are discussed later. Noise computations similar to the circular cylinder cases are shown, and likewise the noise is generated from laminar vortex shedding and boundary

layer interaction. Broadband noise prediction relies on the quality of turbulence resolved in the boundary layer. If the NLDE are to be solved in the trailing edge region only, these turbulent statistics need to be injected upstream at the entrance to the computational domain. Although the NLDE method is envisioned to predict TBL-TE noise, the ability to provide the NLDE code with turbulence at the inflow is currently limited in this aspect. Therefore, all airfoil noise prediction cases are based on the same technique used to trigger the unsteady circular cylinder shedding. The entire airfoil domain will be solved, but with the grid resolution greatly increased in the trailing edge region. All grids have a wall normal grid spacing of  $y^+ \approx 1$ . The NLDE solutions are run until unsteady laminar vortex shedding and boundary layer interaction is seen around the trailing edge. Noise predictions will be shown for both the NACA 0012 and the BSDS flatback airfoil. A similar set of observer positions as described in the cylinder shedding cases was placed surrounding the trailing edges, but at a radius of five chord lengths.

### 5.2.1 NACA 0012

An NACA 0012 series airfoil with a 0.1% trailing edge thickness (relative to chord) is used to represent a “sharp” trailing edge relative to the flatback airfoil design. The grid used is pictured in Figure 5.5 and consists of 1128 points around the chord and 100 stretched points in the wall normal direction. Uniform  $M = 0.2$ ,  $Re_c \approx 4.5$  million mean flow is assumed with pressure and density of 101 kPa and 1.2 kg/m<sup>3</sup>, respectively. Figure 5.6 is the pressure perturbation solution after 0.025 seconds at 0° angle of attack. Acoustic pressure contours can be seen in Figure 5.7, where the pressure waves are generated by the interacting boundary layers and vortex shedding near the trailing edge. Figure 5.8a is the SPL spectrum for an observer placed five chords away from the trailing edge in the transverse direction at 90° . The dominant SPL is at the trailing edge shedding frequency, calculated to be approximately 2300 Hz. Figure 5.8b is the directivity plot of the SPL at the shedding frequency.

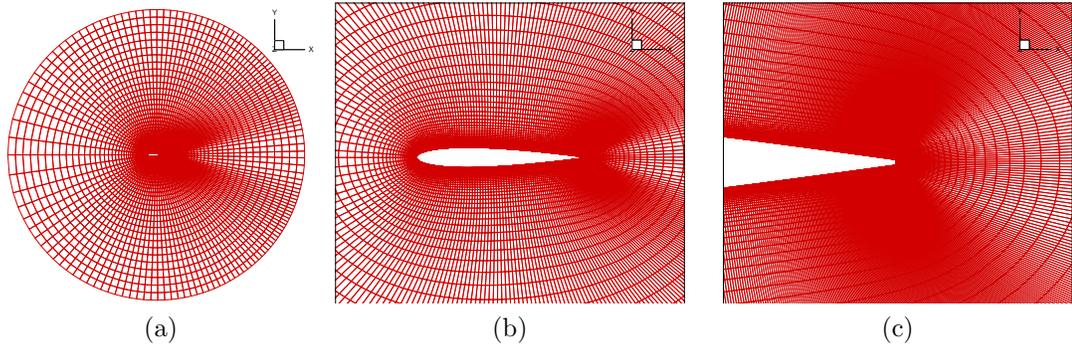


Figure 5.5: NACA 0012 grid for NLDE noise prediction

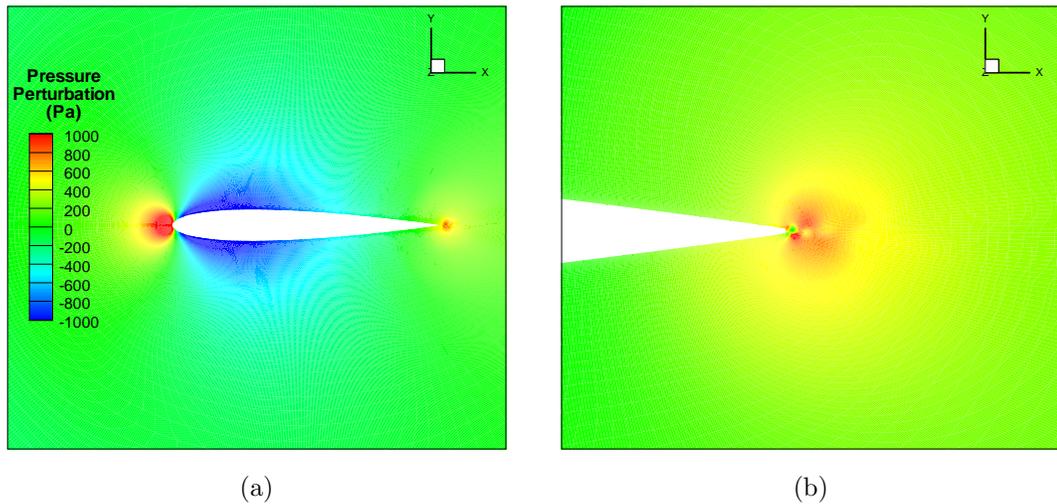


Figure 5.6: Pressure perturbation solution after 0.025 seconds for the NACA 0012 airfoil at  $Re_c \approx 4.5$  million and  $0^\circ$  angle of attack (a) for the entire airfoil section and (b) for a zoomed in trailing edge region.

### 5.2.2 BSDS Flatback

Sandia National Laboratories uses an innovative turbine airfoil known as a “Flatback” airfoil. The flatback design is used on the Blade Systems Design Study (BSDS) rotor in the root region, to increase strength in the structure while maintaining aerodynamic performance[5]. A section of a flatback blade is pictured in Figure 5.9. Unlike a truncated trailing edge which degrades aerodynamic performance, the flatback airfoil trailing edge is opened uniformly along the camber line in an attempt to maintain aerodynamic and acoustic performance. The airfoil

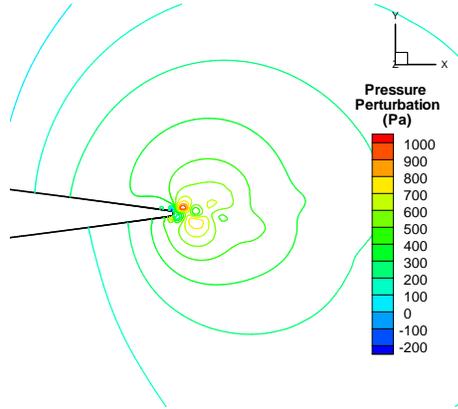


Figure 5.7: Pressure perturbation contours after 0.025 seconds for the NACA 0012 airfoil at  $Re_c \approx 4.5$  million and  $0^\circ$  angle of attack.

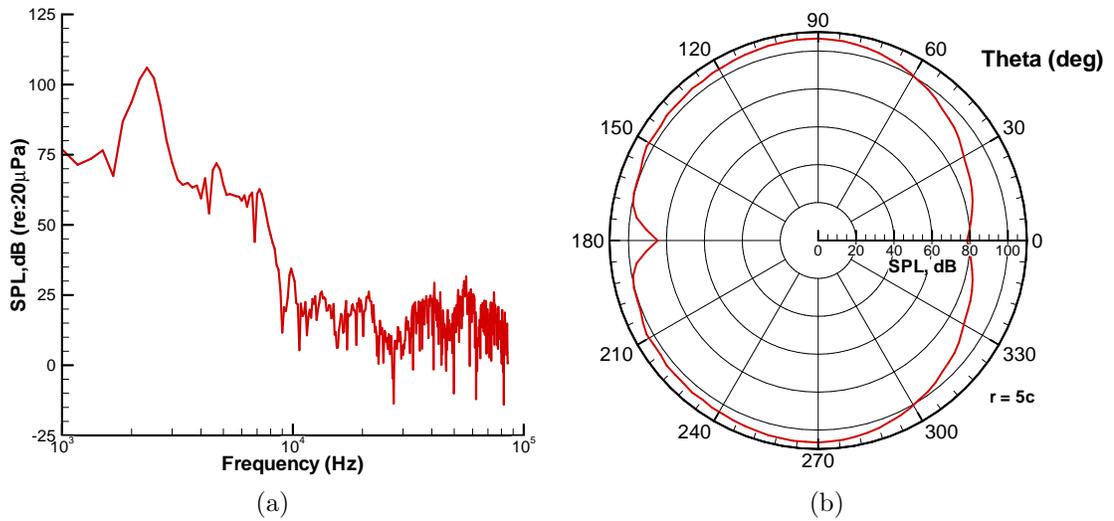


Figure 5.8: SPL results of NLDE - PSU-WOPWOP prediction for NACA 0012 airfoil (a) at observer placed at  $90^\circ$  (b) directivity plots at shedding frequency  $\approx 2300$  Hz. All observers are 5 chords from trailing edge.

design allows for increase in blade thickness without increasing chord beyond reasonable limits for transportation.[5] Figure 5.10 shows the grid used for the BSDS flatback airfoil blade section located at approximately mid-span of the BSDS rotor. The rotor has a nine meter radius and assuming a rotation rate of sixty revolutions per minute, the mid-span location is subject to a local flow velocity of approximately  $M=0.1$   $Re_c \approx 2.25$  million. The trailing edge thickness of the BSDS

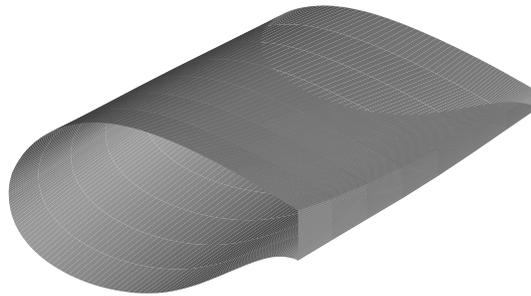


Figure 5.9: The Blade Systems Design Study “flatback” wind turbine blade (root section)

flatback blade section is 5%, which is 50 times thicker than the NACA 0012 “sharp” trailing edge. Again the mean flow is assumed uniform with pressure and density of 101 kPa and  $1.2 \text{ kg/m}^3$ . Figure 5.11 shows the pressure perturbation solution after 0.025 seconds at  $0^\circ$  angle of attack, after the blade travels approximately two chord lengths. Bluff body laminar shedding is observed behind the trailing edge as expected. Figure 5.12a is the SPL spectrum for an observer placed 5 chords away from the trailing edge in the transverse direction at  $90^\circ$ . The dominant SPL is at the trailing edge shedding frequency. Figure 5.12b is the directivity plot of the SPL at the shedding frequency of approximately 150 Hz.

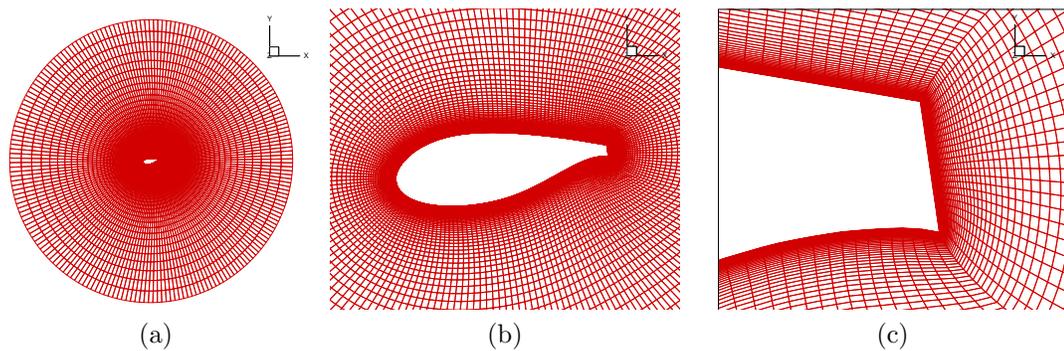


Figure 5.10: BSDS root grid for NLDE noise prediction

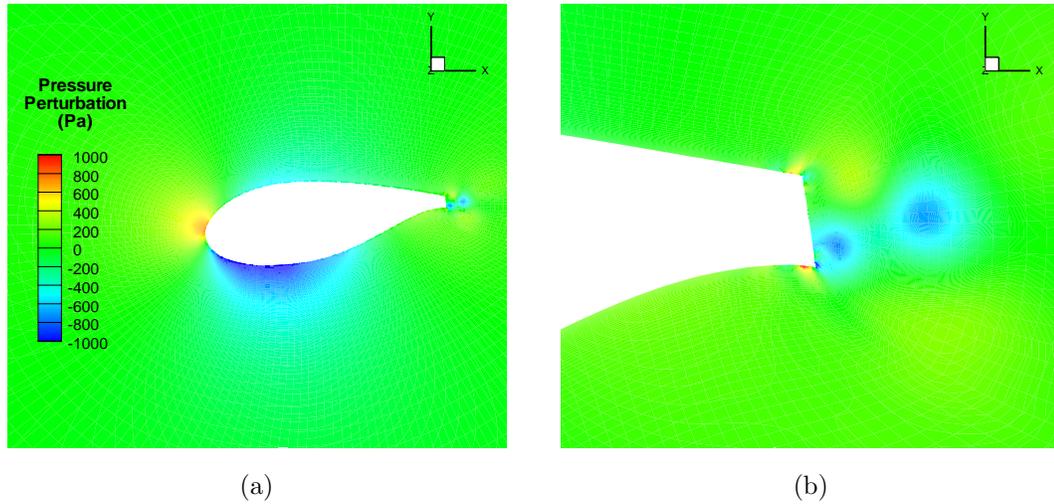


Figure 5.11: Pressure perturbation solution after 0.025 seconds for the flatback airfoil.

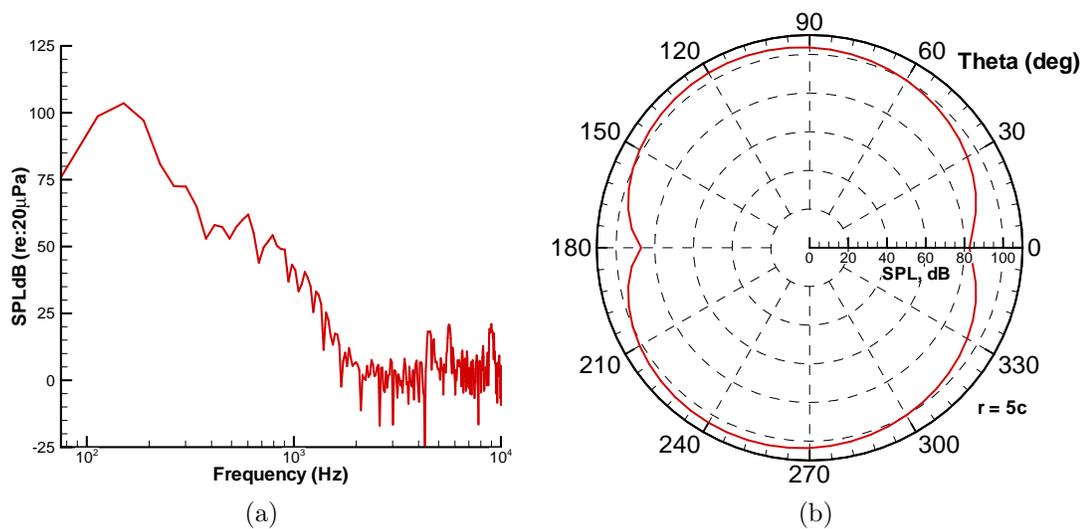


Figure 5.12: SPL results of NLDE - PSU-WOPWOP prediction for BSDS flatback airfoil (a) at observer placed at  $90^\circ$  (b) directivity plots at shedding frequency  $\approx 150$  Hz. All observers are 5 chords from trailing edge.

# Chapter 6

## Suggestions for Future Work and Summary

The first part of this chapter discusses the current limitations of the NLDE code as well as suggestions for improvement. Then, a summary of the completed work is discussed as well as suggestions for continuing the research.

### 6.1 Turbulence Injection

In order to confine the use of NLDE in a limited region such as a trailing edge, turbulence injection techniques need to be developed. The NLDE flow solver cannot presently predict the TBL-TE noise envisioned at the outset of the project because of lack of turbulent input data. Currently, the NLDE code is solving the entire blade domain but with increased resolution in the vicinity of the trailing edge. Accurate TBL statistics are needed for the NLDE to predict the scattering of turbulent energy from trailing edges. A comprehensive summary of the existing techniques for generating turbulent inflow statistics can be found in Reference [36].

The NLDE method has been used to reconstruct turbulent fluctuations in the past. Labourasse and Sagaut [14] used the NLDE to study pulsating channel flow and low pressure turbine blade turbulence. A recycling and rescaling method can be used to generate the turbulence over the surface of the wind turbine blade. As illustrated in Figure 6.1, random initialization is injected into a recycling region, where the NLDE code is used to develop the turbulence. At some region down-

stream, the solution is sampled, recycled and rescaled, and injected back into the inflow. After some time of recycling, the NLDE code should successfully generate realistic turbulence that is independent of the initialization. This turbulence will then interact with the trailing edge and create broadband noise.

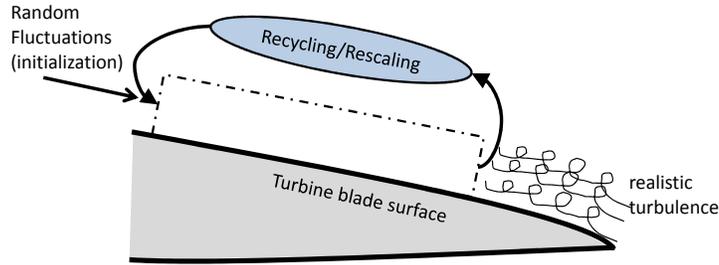


Figure 6.1: Turbulence injection, recycling, and rescaling technique proposed for the NLDE code

## 6.2 Extended NLDE Code Validation

The circular cylinder and airfoil cases presented in this thesis are a first attempt to test the capability of the new NLDE flow solver. They are by no means exhaustive and should be extended. Although vortex shedding noise and laminar boundary layer interaction noise was predicted, it was not compared with experimental data.

### 6.2.1 Circular Cylinder Cases

The vast majority of circular cylinder cases have been run in 2-D for computational efficiency. The 3-D capability of the code should be tested further. For the circular cylinder cases, a periodic boundary condition can be applied in the spanwise direction. This would allow three dimensional cylinders to be compared directly with the experimental work of Revell *et al.* [25].

### 6.2.2 Airfoil cases

In a real wind turbine or helicopter rotor application, the flow is rotational. This effect is certainly not captured in the 2-D mode and also not captured when run-

ning 3-D blade sections without the appropriate modifications to the governing equations[21]. In addition, results were presented for an extremely limited set of example cases. New cases need to be developed and run to investigate the effects of angle of attack and Mach number on the noise generated by the trailing edge. Also, the NLDE results should be compared with experimental data which was not performed in this thesis work.

### **6.2.3 Grid Refinement Studies**

The numerical schemes used in the code were selected for high order of accuracy. Although validation cases demonstrated the accuracy of the NLDE solver, the order of accuracy was not validated. To perform this validation, the grids in the Gaussian pulse validation cases should be refined to verify the order properties of the schemes. The goal of grid refinement is to generate a series of solutions using various grid sizes. If there are no coding mistakes, as the grid is refined, the observed order of accuracy should match the theoretical order of accuracy of the scheme.

## **6.3 NLDE Code Improvements**

### **6.3.1 Low Pass Filtering**

The low pass filtering described in Section 3.2.6 is currently performed by the master processor. If the low pass filter is turned on, at each time step the master processor collects the solution from each child processor, filters it, then redistributes the solution to the processors. This process is slow and reduces parallel speedup. This routine should be reprogrammed such that each subdomain is capable of filtering its own solution so no data collection needs to be performed at each timestep.

### **6.3.2 C-type Grid Support**

The code needs to be extended to support C-type grids. Currently, O-grids are wrapped around airfoil trailing edges with a finite thickness. This leads to poor

grid quality immediately behind the trailing edge when the thickness becomes very small (Figure 2.2b). The code was originally programmed for O-type grids for ease of use with circular cylinders and flatback type airfoils, but adding C-type support is a necessary addition for thin trailing edges.

### 6.3.3 Acoustic Data Surfaces

Acoustic data surfaces are used to extract the flow solution from the NLDE code and provide PSU-WOPWOP with the necessary data to predict the noise. Currently, the user can specify which layers in the existing NLDE grid to use as the acoustic data surfaces. A helpful improvement would be to have the NLDE code read in an arbitrary acoustic data surface file, and interpolate the NLDE solution onto it.

### 6.3.4 Multi Time Stepping

Viscous, compressible code performance is inhibited by extremely small time stepping. The spatial resolution needed for accurate turbulence prediction in the trailing edge region limits the time step limit as outlined in Section 3.2.2. Currently, a single step, five stage Runge-Kutta method is used. To simulate an airfoil traveling only a single chord length, this single step method has been found to require on the order of a million time steps. A multi step method would allow for larger time steps, greatly reducing the computer run time required for a sufficient physical solution time.

## 6.4 Summary and Future Direction

A computational aeroacoustic tool is under development that allows for a first principles based approach to predict TBL-TE noise in a reasonable computational time. The hybrid approach uses a RANS solver for a quick estimate of the mean flow, a NLDE solver which refines the computation in both space and time surrounding noise generating areas of interest, and PSU-WOPWOP to calculate the noise from the NLDE solution. The NLDE flow solver has been validated with exact solutions, and first attempts at airfoil noise prediction have been presented. The

circular cylinder and airfoil cases all experience vortex shedding in the wake which generates noise. This intermediate step demonstrates the ability of the NLDE code to resolve the unsteady flow and to provide time histories of the acoustic pressure for noise prediction.

Turbulence injection techniques should be one of the first areas of focus for the research to continue towards TBL-TE noise prediction. In addition, a RANS solver should be used to generate the mean flow and compared with results when using uniform mean flow. Ultimately, the NLDE should be solved in only the trailing edge region with a RANS mean solution, not the entire domain with a uniform mean flow. As the code develops, the 3-D accuracy should continue to be validated for full 3-D wind turbine rotor simulations. Specific span-wise blade sections of interest can be simulated individually with the NLDE code. These span-wise sections can be run simultaneously in PSU-WOPWOP for an overall estimate of the noise from the entire wind turbine rotor.

# Bibliography

- [1] Mauhmoud Abdelaal. Informal interview. US Marine Corps mechanic, November 19, 2006.
- [2] Danish Wind Industry Association. Gearboxes for wind turbines. <http://www.windpower.org/en/tour/wtrb/powtrain.htm>, September 19, 2003.
- [3] N. Atalla and S. Glegg. Ray-acoustics approach to fuselage scattering of rotor noise. *Journal of the American Helicopter Society*, 38(3):56–63, July 1993.
- [4] N. Atalla and S. Glegg. A geometrical acoustics approach for calculating the effects of flow on acoustic scattering. *Journal of Sound and Vibration*, 171(5):681–694, 1994.
- [5] D.E. Berg and M. Barone. Aerodynamic and aeroacoustic properties of a flatback airfoil (will it rumble or whisper?). Technical report, Sandia National Laboratories, WINDPOWER 2008, Houston, TX, June 2-4 2008.
- [6] K. Brentner and F. Farassat. Helicopter noise prediction: the current status and future direction. *Journal of Sound and Vibration*, 170:79–96, 1994.
- [7] T. Brooks and T. Hodgson. Trailing edge noise prediction from measured surface pressures. *Journal of Sound and Vibration*, 78(11):69–117, 1981.
- [8] T. Brooks, D. Pope, and M. Marcolini. Airfoil self-noise and prediction. Technical Report 1218, NASA, 1989.
- [9] W. M. Chan, R. J. Gomez III, S. E. Rogers, and P.G. Buning. Best practices in overset grid generation. In *AIAA Paper 2002-3191, AIAA 32<sup>nd</sup> Fluid Dynamics Conference and Exhibit*, St Louis, Missouri, June 2002.

- [10] T. Chyczewski, P. Morris, and L. Long. Large-eddy simulation of wall bounded shear flow using the nonlinear disturbance equations. In *AIAA Paper 2000-2007, AIAA 6<sup>th</sup> AIAA/CEAS Aeroacoustics Conference*, Lahaina, Hawaii, June 2000.
- [11] M. Dunn and A. Tinetti. Aeroacoustic scattering via the equivalent source method. In *AIAA Paper 2004-2937, AIAA 10<sup>th</sup> AIAA/CEAS Aeroacoustics Conference*, Manchester, United Kingdom, May 2004.
- [12] R.P. Hansen, L.N. Long, and P.J. Morris. Unsteady, laminar flow simulations using the nonlinear disturbance equations. Technical Report AIAA-2000-1981, American Institute of Aeronautics and Astronautics, 2000.
- [13] F.Q. Hu, M.Y. Hussaini, and J. Manthey. Low-dissipation and -dispersion runge-kutta schemes for computational acoustics. *Journal of Computational Physics*, 124:177, 1996.
- [14] E. Labourase and P. Sagaut. Reconstruction of turbulent fluctuations using a hybrid rans/les approach. *Journal of Computational Physics*, 182:301–336, 2002.
- [15] S. Lee, J. Erwin, and K. Brentner. Acoustic scattering of rotorcraft noise. In *Presented at the American Helicopter Society 64<sup>th</sup> Annual Forum*, Montreal, Canada, May 2008.
- [16] David Lewicki and Ron. Woods. Evaluation of low-noise, improved-bearing-contact spiral bevel gears. In *Presented at the American Helicopter Society 54<sup>th</sup> Annual Forum*, Pheonix, AZ, May 6–8 2003.
- [17] Lyle Long. A nonconservative nonlinear flowfiled splitting method for 3-d unsteady fluid dynamics. In *AIAA Paper 2000-1998, AIAA 6<sup>th</sup> AIAA/CEAS Aeroacoustics Conference*, Maui, Hawaii, June 2000.
- [18] T. Lund, X. Wu, and K. Squires. Generation of turbulent inflow data for spatially-developing boundary layer simulations. *Journal of Computational Physics*, 140:233–258, 1998.

- [19] O. Marsden, C. Bogey, and C. Bailly. Direct noise computation around a 3-d naca 0012 airfoil. *AIAA Paper*, (0146-3705), 2006.
- [20] G. Mathur, J. O’Connell, R. JanakiRam, and C. Fuller. Analytical and experimental evaluation of active structural acoustic control (asac) of helicopter cabin noise. In *4<sup>th</sup> AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 14–17, 2002.
- [21] Steven Miller. Rotational effects on the aerodynamics of wind turbine airfoils. Master of Science Thesis, Penn State University, 2006.
- [22] T. Millott and W. Welsh. Flight test of active gear-mesh noise control on the s-76 aircraft. In *Presented at the American Helicopter Society 54th Annual Forum*, Washington, DC, May 20–22 1998.
- [23] P. Morris, Q. Wang, L. Long, and D. Lockard. Numerical predictions of high speed jet noise. *AIAA Paper*, (97-1598), 1997.
- [24] Phillip Morris, Lyle Long, Ashok Bangalore, and Qunzhen Wang. A parallel three-dimensional computational aeroacoustics method using nonlinear disturbance equations. *Journal of Computational Physics*, 133:56–74, 1997.
- [25] J.D. Revell, R.A. Prydz, and A.P. Hays. Experimental study of airframe noise vs. drag relationship for circular cylinders. Technical Report Lockheed Report 28074. Final Report for NASA Contract NAS1-14403, 2004.
- [26] R. H. Schlinker and R.K. Amiet. Helicopter rotor trailing edge noise. Technical Report CR-3470, NASA, 1981.
- [27] C. Shieh. Parallel numerical simulation of subsonic, turbulent, flow induced noise from two and three-dimensional cavities using computational aeroacoustics. PhD thesis, Penn State University, 2000.
- [28] C. Tam. Computational aeroacoustics: Issues and methods. *AIAA Journal*, 33(10), 1995.
- [29] C. Tam and Z. Dong. Wall boundary conditions for high-order finite-difference schemes in computational aeroacoustics. *Theoretical and Computational Fluid Dynamics*, 6:303–322, 1993.

- [30] C. Tam and J Hardin. Analytical solutions of the category 1, benchmark problems 1 and 2. In *NASA Conference Publication 3352, 2<sup>nd</sup> Computational Aeroacoustics Workshop on Benchmark Problems*, Langley Research Center, Hampton, VA, June 1997.
- [31] C. Tam and J. Webb. Dispersion-relation-preserving finite difference schemes for computational acoustics. *Journal of Computational Physics*, 107:262–281, 1993.
- [32] Claudio Testa, Sandro Ianniello, Giovanni Bernardini, and Massimo Gennaretti. Sound scattered by a helicopter fuselage in descent flight condition. In *AIAA Paper 2007-3497, AIAA 13<sup>th</sup> AIAA/CEAS Aeroacoustics Conference*, Rome, Italy, May 2007.
- [33] P. Thomas and C. Lombard. Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal*, 17:1030, 1979.
- [34] A. Tinetti, M. Dunn, and D. Pope. Fast scattering code (fsc) user’s manual version 2.0, March 2006.
- [35] M. Visbal and D. Gaitonde. On the use of higher-order finite-difference schemes on curvilinear and deforming meshes. *Journal of Computational Physics*, 181:155–185, 2002.
- [36] S. Xu and P. Martin. Assessment of inflow boundary conditions for compressible turbulent boundary layers. *American Institute of Physics*, 16(7), 2004.
- [37] W. Zhu, N. Heilskov, W. Shen, and J. Sørensen. Modeling of aerodynamically generated noise from wind turbines. *Journal of Solar Energy Engineering*, 127:517–528, 2005.

# Appendix A

## Acoustic Scattering

Acoustic scattering can greatly modify the noise signal of a helicopter rotor. The focus of acoustic scattering in this Appendix will be the scattering of rotor noise by solid bodies. Significant scattering of helicopter main rotor noise was demonstrated using a ray-acoustics approach by Atalla and Glegg [3][4]. In 2007, Testa *et al.* showed that the presence of the fuselage can either alleviate or increase the noise in the near and far-field [32].

Depending of the orientation of the rotor and the body, the fuselage can modify the directivity and intensity of the rotor noise. Lee, Erwin, and Brentner showed that tail rotor scattering can be significant due to the small wavelength of the noise relative to the size of the fuselage. First attempts at the scattering of quad tilt rotors was also presented [15]. This Appendix summarizes the work performed by Lee, Erwin, and Brentner to predict acoustic scattering.

NASA Langley's Fast Scattering Code (FSC) is an acoustic prediction tool that calculates how sound is scattered by the presence of solid bodies. A detailed derivation and the associated equations used in the code are presented by Dunn and Tinetti [11]. In summary, the code predicts how solid bodies alter the radiation of sound at specific frequencies. The methodology is to use PSU-WOPWOP to provide the FSC with the necessary acoustic variables at these frequencies, and use the FSC to compute the scattered noise field. Figure A.1 is a schematic of the problem solving process. First, a geometry file such as a helicopter fuselage is input to the geometry module in the FSC. The FSC generates collocation points which are used in the acoustic scattering prediction. These collocation points are

then given to PSU-WOPWOP, which calculates acoustic pressure gradient and acoustic velocity at each collocation point. This solution is given back to the FSC, along with similar acoustic information at any desired set of observer positions. The FSC scattering utility then predicts the scattered noise field.

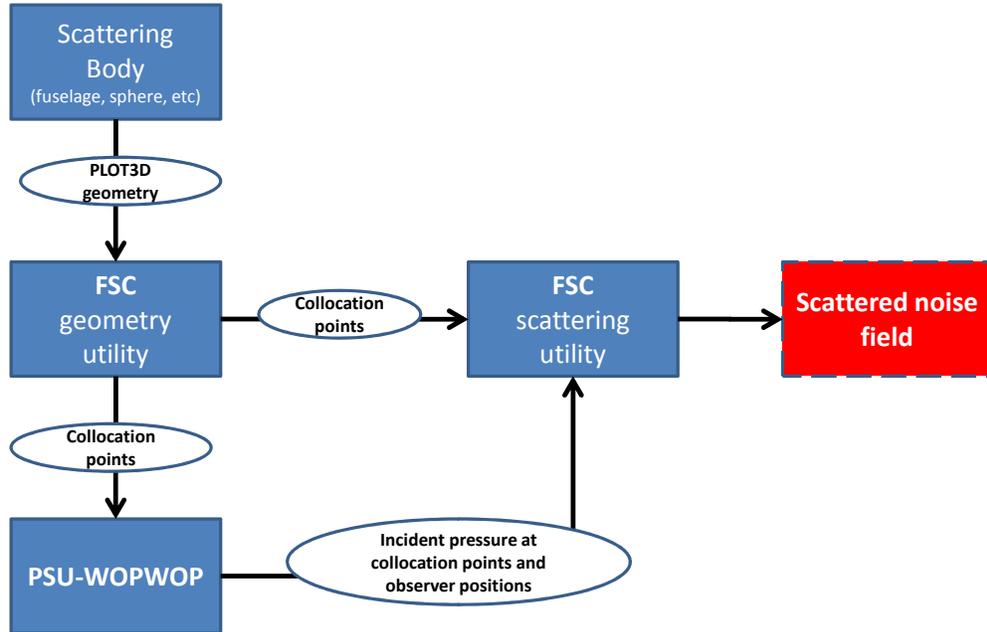


Figure A.1: Acoustic scattering problem solving process.

In the next sections, a description of the coupling between PSU-WOPWOP and the FSC will be presented. In addition, validation cases and rotor scattering predictions will be summarized. Detailed analysis of the results can be found in Reference [15].

## A.1 Coupling with PSU-WOPWOP

An addition was made to PSU-WOPWOP which outputs acoustic data required by the FSC. The input files to the FSC require specific ASCII formatting. The details of these formats can be found in the FSC User Manual [34]. When the appropriate flags are turned on in the PSU-WOPWOP input namelist file, PSU-WOPWOP outputs the necessary acoustic data at the scattering body collocation

points and the observer positions. Because the FSC is a frequency domain code and PSU-WOPWOP is a time domain code, PSU-WOPWOP outputs the results at specific frequencies after performing a Fast Fourier Transform (FFT) of the acoustic pressure time histories. The frequencies are analyzed individually in the FSC for the scattered solution.

## A.2 Validation Case

The validation case chosen by the authors was the scattering of a point source by a sphere. The setup of the case is illustrated in Figure A.2. A sphere of radius  $a = 1$  meter is placed next to a point monopole source located at a distance  $r_1 = 2$  meters from the center of the sphere. Results are only presented here for one frequency,  $ka = 10$  and  $\lambda = 0.6284$  meters. A ring of observers was placed surrounding the sphere located at  $r/a = 1.5$ .

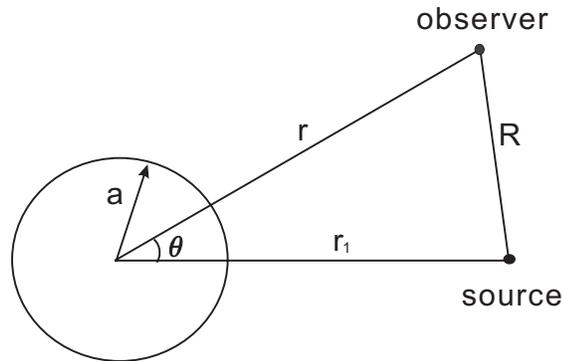


Figure A.2: Setup of the scattering sphere validation case.[15]

Figure A.3 is a comparison of the PSU-WOPWOP and FSC prediction versus the analytical solution. Excellent agreement is achieved, and the effect of acoustic scattering is significant in this case. The total field is increased toward the source due to constructive interferences between the incident and scattered field, and the opposite effect (destructive interference) is found on the opposite side of the sphere from the source. Figure A.4 is an SPL contour plot of the incident and total field on an observer plane five meters below the rotor. Significant acoustic scattering is observed from the presence of the sphere.

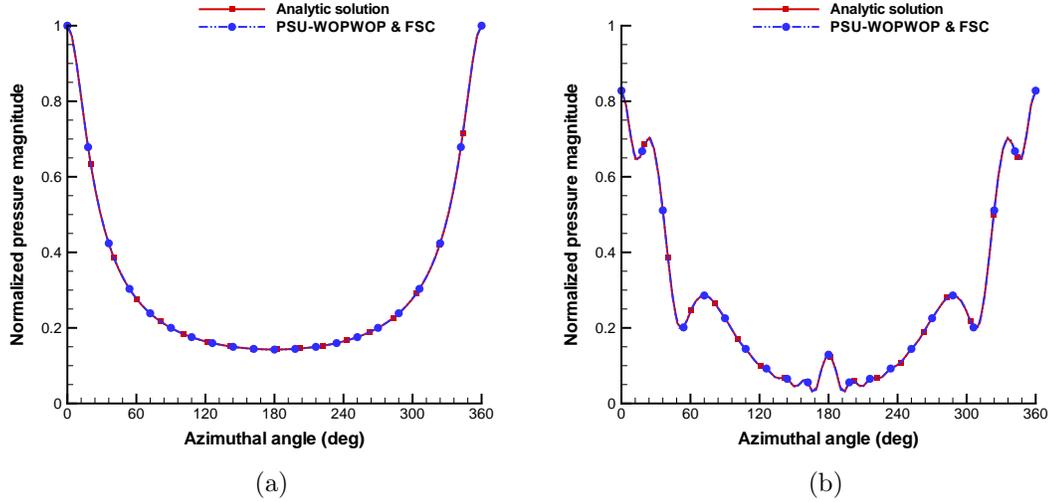


Figure A.3: Validation of a point source scattering problem. The pressure magnitude is normalized by the magnitude of the incident pressure at  $\theta = 0$ : (a) the incident field, (b) the total field.[15]

## A.3 Acoustic Scattering of Rotorcraft

First attempts at the prediction of the scattering of rotor noise from helicopter fuselages were also performed. Brief results for two cases will be presented here. First is the scattering of tail rotor noise by a BO105 fuselage in hover. The second is the scattering of noise from a Quad Tilt Rotor (QTR) conceptual aircraft. Reference [15] presents the results in more detail.

### A.3.1 BO105 Tail Rotor Scattering

A BO105 helicopter was chosen as a test configuration to investigate the scattering effect that a fuselage has in the presence of tail rotor noise. The helicopter configuration used is illustrated in Figure A.5, where only the tail rotor (colored red) is considered. The operating conditions of the tail rotor are given in Table A.1. The tail rotor noise wavelength is comparable to the size of the fuselage, so acoustic scattering is expected.

Figure A.6 shows the incident and total sound pressure level on an observer plane located eleven meters below the main rotor hub location. Only the results

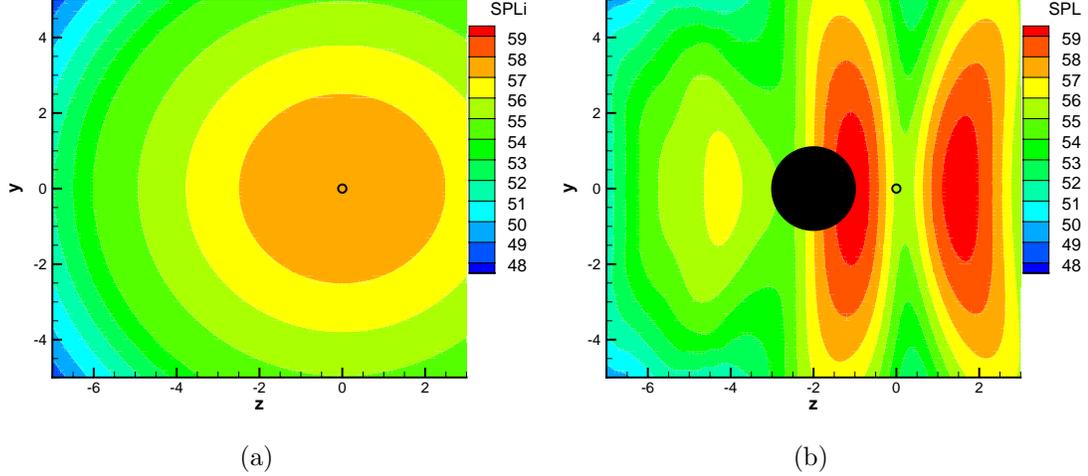


Figure A.4: Sound pressure level for a point source scattering validation case. The observer grid is located 5 m below the source: (a) the incident field, (b) the total field.[15]

Table A.1: Operating conditions of the BO105 tail rotor.[15]

rotor radius (m)	0.95
rotor solidity	0.12
blade chord (m)	0.18
angular velocity (rad/s)	232.63
hovering tip Mach number	0.65
blade passage frequency (Hz)	74.05

for tail rotor noise are presented at the second harmonic of tail rotor BPF. It can be seen that the fuselage modifies the incident noise field considerably, especially on the sides of the fuselage where the incident field had very low SPL.

### A.3.2 Quad Tilt Rotor Scattering

The acoustic scattering results of a conceptual Quad Tilt Rotor (QTR) aircraft will be summarized in this section. The QTR is illustrated in Figure A.7. The blade geometry and operating conditions are given in Table A.2. The fifth harmonic of the rotor BPF, 91.66 Hz, is selected to demonstrate the scattering effect.

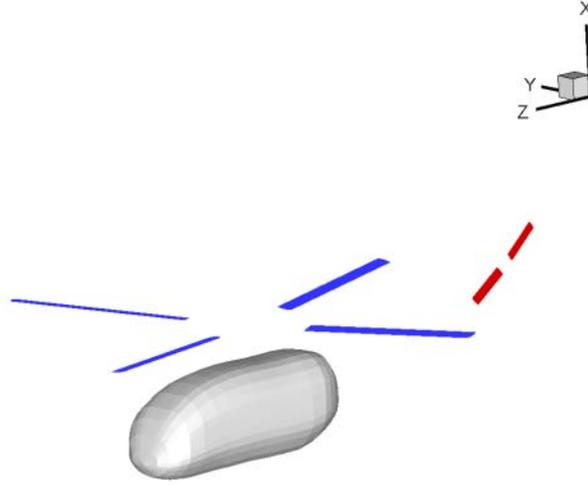


Figure A.5: BO105 helicopter configuration.[15]

Two different configurations were selected to evaluate the effect of relative rotor position on acoustic scattering. Figure A.8 is a side view of the QTR configurations. The first configuration has relatively high rotors as compared to the second configuration, where the body is elevated into the rotor plane.

Figure A.9 are the SPL contours on an observer plane located ten meters below the front rotor hub. A complex incident pressure field is due to constructive and destructive interference between the four rotors even with no scattering body present. When the rotor planes are located above the QTR body, little scattering is observed, while in the second case, the scattering effect is increased significantly.

Table A.2: Operating conditions of the QTR.[15]

rotor radius (m)	8.5
rotor solidity	0.0989
blade chord (m)	0.6
angular velocity (rad/s)	24.0
hovering tip Mach number	0.6
blade passage frequency (Hz)	15.28

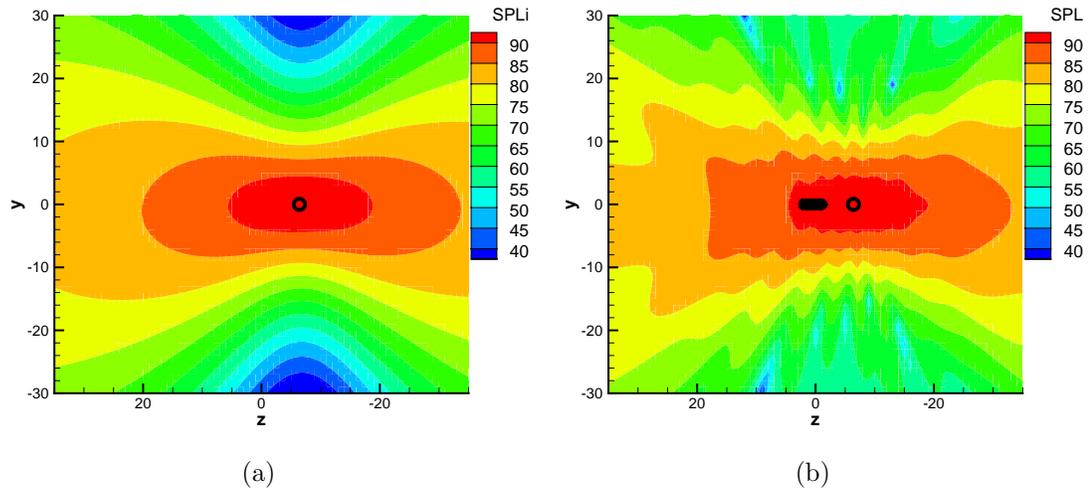


Figure A.6: Sound pressure level for the tail rotor noise at the second harmonic of tail rotor BPF. The observer grid is located 11 m below the main rotor hub: (a) the incident field, (b) the total field.[15]

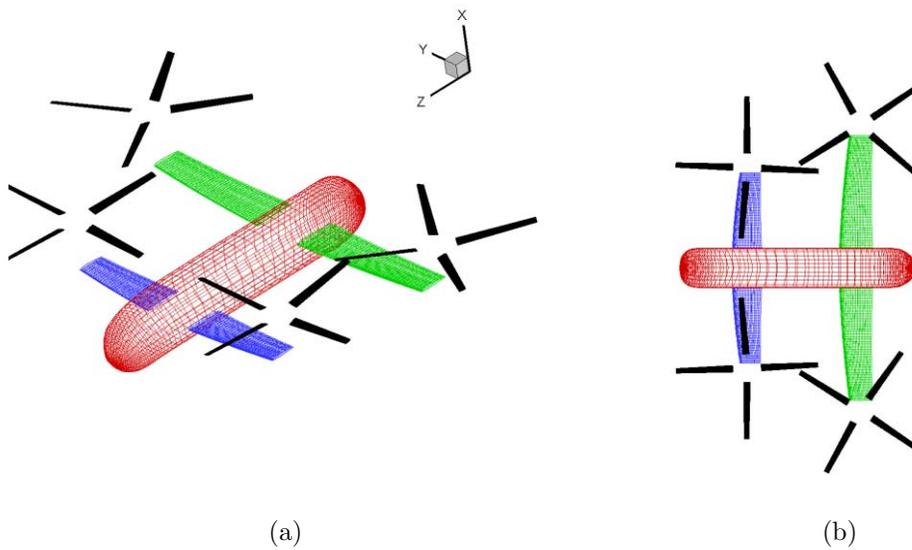


Figure A.7: The conceptual QTR selected for acoustic scattering investigation: (a) isometric view, (b) top view.[15]

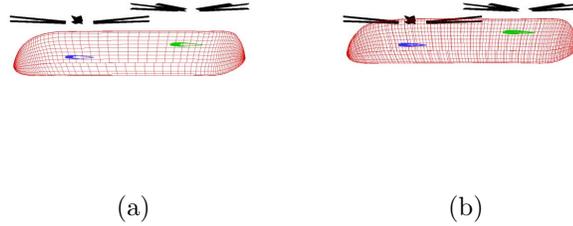


Figure A.8: Two QTR configurations: (a) high rotor configuration, (b) low rotor configuration.[15]

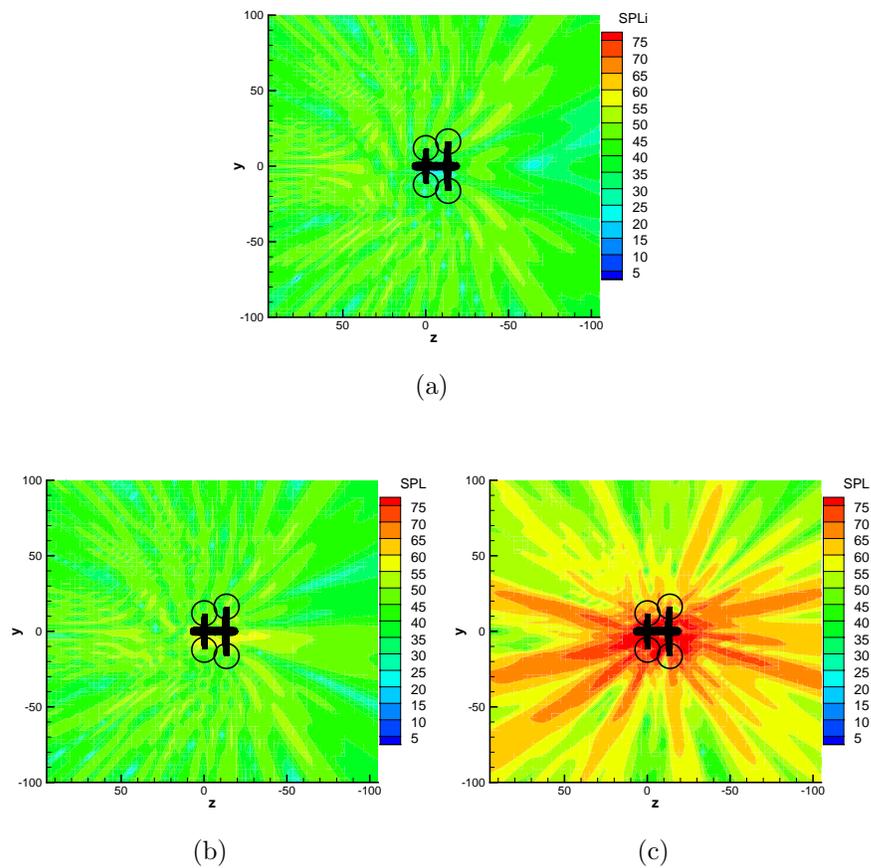


Figure A.9: Sound pressure level for the QTR noise at the fifth harmonic of rotor BPF. The observer grid is located 10 m below the front rotor hub: (a) the incident field, (b) the total field for high rotor configuration, (c) the total field for low rotor configuration.[15]

# Appendix B

## Source Code for Exact Solutions

In this appendix, the exact solutions along with MATLAB source code will be presented for the validation cases used with the NLDE code.

### B.1 Scattering of a Gaussian Pulse by a Circular Cylinder

For an initial Gaussian pressure distribution given by

$$p' = \exp [b ((x - x_s)^2 + y^2)] \quad (\text{B.1})$$

where  $b = \ln(2)/w^2$ , solution for the pressure field may be found at any location and time by evaluating

$$p(r, \theta, t) = \int_0^\infty A(r, \theta, \omega) \cdot \omega \cdot \cos(\omega t) d\omega \quad (\text{B.2})$$

Where  $A$  is calculated with a series of Hankel transforms

$$A(r, \theta, \omega) = \frac{e^{-\omega^2/(4b)}}{2b} \left\{ J_0(\omega \sqrt{r^2 + x_s^2 - 2rx_s \cos \theta}) + \sum_{k=0}^{\infty} Re \left[ \frac{\epsilon_k H_k^{(1)}(r\omega) \cos(k\theta)}{\pi\omega \left[ \frac{2k}{\omega} H_k^{(1)}(\omega/2) - H_{k+1}^{(1)}(\omega/2) \right]} \right] \cdot \int_0^\pi \frac{\omega(0.5 - x_s \cos \eta) J_1(\omega \sqrt{0.25 + x_s^2 - x_s \cos \eta}) \cos(k\eta)}{\sqrt{0.25 + x_s^2 - x_s \cos \eta}} d\eta \right\} \quad (\text{B.3})$$

where  $\epsilon_k = 1$  for  $k = 0$  and  $\epsilon_k = 2$  for all other  $k$ .

### B.1.1 MATLAB code

The MATLAB code is a collection of routines. The main code to execute is “main.m”. The code numerically integrates the above equations with a resolution definable by the user in “main.m”.

#### main.m

```
format long;
tic;

%the pulse is of the form
% p = e^(-ln2((x-x_s)^2+y^2)/w^2)
% where b = ln2 / w^2
% so p = e^(-b*((x-x_s)^2+y^2))
%The half-width of the Gaussian:
w = 0.2;
%The location of the center of the pulse:
x_s = 4.;
b = log(2)/w^2;
%Define an upper limit for the bound of the numerical integration:
kmax = 7;
omega_max = 50 ;
%Adjustable parameter:
d_omega = .01;
%The location of the point the solution is for:
```

```

r = 5.0;
theta = pi/2;
%Time can be explicitly set here for a solution at one time value.
% Uncomment the following two lines for the solution at a single point
% or leave them commented for a transient solution.
%t = 8.2;
%p = getp(r,theta,t,omega_max,d_omega,b,x_s,kmax)
i = 1; %initialize the counter
%This loop will give the solution from t = 6 to 10 seconds:
for t = 6 : 0.01: 10
    t
    tarray(i) = t;
    p(i) = getp(r,theta,t,omega_max,d_omega,b,x_s,kmax);
    toc
    i = i+1;
end
%
%Create a temp array to write the data:
xy = [tarray' p'];
%Write the data as time in the first column, and pressure in the second:
%Give the file a name:
filename = 'pointa.dat'
dlmwrite(filename,xy,' ')

```

### getA.m

This routine calculates Equation B.3.

```

function[A]=GetA(r,theta,omega,b,x_s,kmax)
%
%The leading exponential coefficient. Note
% the typo in the original equation (53)
% was given as
% coef = exp(-(omega^2)*b/(4(b)))/(2*b)
%
coef = exp(-(omega^2)/(4*b))/(2*b);
apart = getapart(r,theta,omega,x_s);
bpart = getbpart(r,theta,omega,x_s,kmax);
A = coef*(apart + bpart);

end

```

**getapart.m**

This routine calculates the term not included in the  $k$  summation of Equation B.3.

```
function [APart]=GetAPart(r,theta,omega,x_s)

APart = besselj(0,omega*sqrt(r^2+x_s^2-2*r*x_s*cos(theta)));

end
```

**getbpart.m**

This routine calculates the value of the  $k$  summation in Equation B.3.

```
function [BPart]=getbpart(r,theta,omega,x_s,kmax)

BPart = 0;
for k=0:kmax;
    Bpartold = BPart;
    if k==0
        eps = 1;
    else
        eps = 2;
    end
    temp = omega/2;
    [H,ierr]=besselh(k,1,r*omega);
    numerator = eps*H*cos(k*theta);
    [H1,ierr1] = besselh(k,1,temp);
    [H2,ierr2] = besselh(k+1,1,temp);
    %The following checks any of the above Hankel
    % function results for errors:
    if ierr~=0
        ierr,k,omega,H
        error('ierr is not zero')
    end
    if ierr1~=0
        ierr1,k,omega,H1
        error('ierr1 is not zero')
    end
    pause
end
```

```

if ierr2~=0
    ierr2,k,omega,H2
    error('ierr2 is not zero')
    pause
end
denominator = pi*omega*((k/temp)*H1 - H2);
cpart = getcpart(r,theta,omega,x_s,k);
BPart = BPart + real(numerator/denominator)*cpart;
end

```

### getcpart.m

This routine calculates the integral from 0 to  $\pi$  in Equation B.3 for each value of  $k$ .

```

function[CPart]=getcpart(r,theta,omega,x_s,k)

deta = pi/10;
CPart = 0;
for eta = 0:deta:pi

    temp = sqrt(0.25+x_s^2-x_s*cos(eta));
    %The original equation (53) was missing the
    % leading omega term in the following:
    numerator = omega*(0.5 - x_s*cos(eta))*besselj(1,omega*temp)*cos(k*eta);
    denominator = temp;
    %Numerical integration:
    CPart = CPart + (numerator/denominator)*deta;

end

```

### getp.m

This routine calculates the integral of Equation B.2.

```

function[p]=getp(r,theta,t,omega_max,d_omega,b,x_s,kmax)
p = 0.;
%Numerical integration - do not include omega=0 because then

```

```

% part of Equation 53 would be undefined:
for omega = d_omega : d_omega : omega_max
    pold = p;
    p = p + getA(r,theta,omega,b,x_s,kmax)*omega*cos(omega*t)*d_omega;
    convergence = (p-pold)/p;
    %If there is no change between the old value of p
    % and the new calculated value, then the integration
    % has converged and we can abort early:
    if convergence ==0
        fprintf('converged at')
        omega
        toc
        break
    end
end
end

```

## B.2 Propagation of a Gaussian Pulse with Mach 0.5 Background Flow

For an initial Gaussian pressure distribution given by

$$p' = \exp [b (x^2 + y^2)] \quad (\text{B.4})$$

where  $b = \ln(2)/w^2$ , solution for the pressure field may be found at any location and time by evaluating

$$p(x, y, t) = \rho = \frac{1}{2b} \int_0^\infty \exp^{-\omega^2/(4b)} \cos(\omega t) J_0(\omega \eta) \omega \cdot d\omega \quad (\text{B.5})$$

Where  $\eta = [(x - Mt)^2 + y^2]^{0.5}$ .

### B.2.1 MATLAB code

The MATLAB code is a collection of routines. The main code to execute is “main.m”. The code numerically integrates the above equations with a resolution definable by the user in “main.m”.

**main.m**

```

format long;
clear;
close all hidden;
tic;

%the pulse is of the form
%  $p = e^{(-\ln 2(x^2+y^2)/w^2)}$ 
% where  $b = \ln 2 / w^2$ 
% so  $p = e^{(-b*((x-x_s)^2+y^2))}$ 

%The half-width of the Gaussian:
w = 3;
%The location of the center of the pulse:
b = log(2)/w^2;

mach = 0.5
scale = 0.01

%Adjustable parameter:
omegamax = 1000 ;
domega = .01;

%The location of the point the solution is for:
x = 1.0;
y = 0.0;
%Time can be explicitly set here for a solution at one time value.
% Uncomment the following two lines for the solution at a single point
% or leave them commented for a transient solution.
t = 28.45;
%p = getp(x,y,t,mach,b,domega,omegamax,scale)

i = 1; %initialize the counter
%This loop will give the solution from t = 6 to 10 seconds:
for x = -100 : 1 : 100
    x
    xarray(i) = x;
    dlmwrite('x.dat',xarray')
    p(i) = getp(x,y,t,mach,b,domega,omegamax,scale);
    toc

```

```

    i = i+1;
end
%
%Create a temp array to write the data:
xy = [xarray' p'];
%Write the data as time in the first column, and pressure in the second:
%Give the file a name:
filename = 'tam_m.5_500.dat'
dlmwrite(filename,xy,' ')

```

### getp.m

```

function[p]=getp(x,y,t,mach,b,domega,omegamax,scale)
p = 0.;
%Numerical integration - do not include omega=0 because then
% part of Equation 53 would be undefined:
for omega = 0 : domega : omegamax
    pold = p;
    p = p + getA(x,y,t,mach,b,omega)*domega;
    convergence = (p-pold)/p;
    %If there is no change between the old value of p
    % and the new calculated value, then the integration
    % has converged and we can abort early:
    if convergence ==0
        fprintf('converged at')
        omega
        toc
        break
    end
end
end

p = p*scale/(2*b)

```

### getA.m

```

function[A]=GetA(x,y,t,mach,b,omega)
eta = ((x-mach*t)^2+y^2)^(0.5);
A = exp(-omega^2/(4*b))*cos(omega*t)*BesselJ(0,omega*eta)*omega;
end

```

# Appendix C

## NLDE Flow Solver User Manual

In this appendix, an abbreviated version of the NLDE flow solver user manual will be presented. Within there are instructions to compile the code in both serial and parallel. An example case is presented as well as all the possible input options for controlling how the code runs.

### C.1 Compiling and Running the NLDE Code

#### C.1.1 Compiling the NLDE Code on Linux Clusters

Compiling the code for serial and parallel runs is easy. There is a `Makefile` which points to the individual Fortran source files. Also, a file named `configure` is used to locate the necessary compilers and libraries at link time. First, make sure that the compiler you wish to use is in your path. To determine this, type

```
:>which compilename
```

where `compilename` can be `ifort`, `gfortran`, `f90`, etc. If no compiler was found, you can add the compiler to your path by modifying the `.bashrc` file in your home directory to include the line:

```
export PATH=/path/to/the/fortran/compiler:$PATH
```

and then typing `:>source .bashrc` to reference the new bash file.

During the creation of this User Manual, the COCOA4 cluster at Penn State was used to compile the code in both serial and parallel. In this case, the path looked like this (on a single line):

```
export PATH=/sbin:/usr/local/intel/fc/10.1.015/bin: ...
/usr/local/mvapich_intel_10-1.0.1/bin:$PATH
```

To compile the code in **serial**, enter the `/nlde/src/` directory and type:

```
:>F90=ifort ./configure
:>make
```

and a serial executable file named `nlde_s` will be created. To run the serial code, type:

```
:>./nlde_s
```

To compile the code in **parallel**, enter the `/nlde/src/` directory and type:

```
:>F90=mpif90 ./configure
:>make
```

and a parallel executable file named `nlde_p` will be created. To submit a job to run in parallel using the PBS scheduler, see the next section.

### C.1.2 Submitting a Parallel Job

To submit a parallel job, you must create a queue script file and have both this file and the `nlde_p` executable file in the same directory. A sample queue script file named `QCocoa4.sh` looks like:

```
#PBS -N testcase           !Submission name
#PBS -e errorfile         !Text file for any error output
#PBS -o outputfile       !Text file for other PBS output
#PBS -l nodes=4:ppn=4    !Number of processors and number of nodes
#PBS -l walltime=120:00:00 !Estimated run time (upper limit)

# This job's working directory
echo Working directory is $PBS_O_WORKDIR
```

```

cd $PBS_O_WORKDIR

echo Running on host `hostname`
echo Time is `date`
echo Directory is `pwd`
echo This jobs runs on the following processors:
echo `cat $PBS_NODEFILE`
# Define number of processors
NPROCS=`wc -l < $PBS_NODEFILE`
echo This job has allocated $NPROCS processors.

/usr/local/mvapich_intel_10-1.0.1/bin/mpirun ...
PBS_JOBID=$PBS_JOBID -np $NPROCS ...
-machinefile $PBS_NODEFILE /home/user/nlde/src/nlde_p > OUTPUT.txt

```

Then submit the job into the queue like this:

```

:>qsub QCocoa4.sh

```

You can check the status of your running job by typing:

```

:>qstat

```

and you can see the progress of your running job (code output is placed in `OUTPUT.txt`):

```

:>tail OUTPUT.txt

```

To submit a serial job to run in the background, simply change in the queue script:

```

#PBS -l nodes=1:ppn=1      !use only one processor

```

and the last line points to the serial executable file:

```

/home/user/nlde/src/nlde_s > OUTPUT.txt

```

### C.1.3 Preparing a Job to Run

Running a job is easy. The case file `nlde.nam` tells the program what job to run. This file must be in one of two places:

- For a serial run using the `./nlde_s` command, it must be in same directory as the `nlde_s` executable.
- For a parallel or serial run using a submission script, it must be in the same directory as the submission script.

Figure C.1 is a tree of the relative file location for an example case. The actual case is located in the `/validation/gauss_2d/` directory. When the `QCocoa4.sh` file is submitted, the executable located in `/nlde/src` runs. It opens `nlde.nam` which points the code to the job namelist file, `gauss_2d.nam`. `gauss_2d.x` is the PLOT3D grid and `gauss_2d.save` is the initial Gaussian pulse distribution. You can submit the run script from any directory, as long as the `nlde.nam` file is in the same directory, and the script points to the executable. For this example, the `nlde.nam` file should contain the following text, exactly:

```
&CaseIn
  CaseFolderName = './validation/gauss_2d/'
  CaseFileName   = 'gauss_2d.nam'
/
```

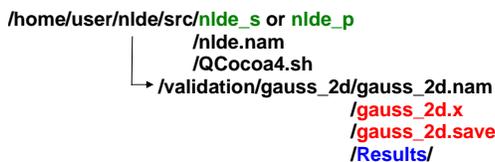


Figure C.1: Tree structure for example case

### C.1.4 The Job Namelist File

This namelist file contains everything needed to control how the code is run. The following is a list variables and their explanation organized by group.

#### &EnvironmentIn

#### General options

- `debugLevel` - controls how verbose the output of the code during execution is. Possible values are 1, 2, and 3 where 3 is the most verbose and 1 is the least. Default is 1.
- `restartFlag` - Turn this flag on to read a restart file and continue from where a previous run left off. The restart file must be in the job directory. This file is a PLOT3D Q file output by the last successfully completed NLDE run. The output file name after the last time step of the previous run is always located in the Results folder and named `restart.q`.
- `restartName` - Of the form “restart.q” or something similar. This is the name, in quotations, of the restart file in the case directory to read for the program restart.
- `restartStep` - An integer declaring the timestep the restart file corresponds to. For example, if the last timestep of the previous run was 600, then `restartStep=600`. The next timesteps will continue from 600. This tells the NLDE code how to label its subsequent output files.

### Input file paths

All file paths are relative to the directory where the program is being run from.

- `gridPath` - the full path to the RANS grid file. For example the default is `‘./grid.in’`.
- `QPath` - the full path to the RANS solution Q file. For example the default is `‘./solution.q’`.
- `NLDEGridPath` - the full path to the NLDE grid. The NLDE grid is optional and will only be used if specified here. If not specified, the flow solver will operate on the RANS grid listed above.

### Calculation options

- **UNIFORMFLOWFLAG** - This flag turns on the use of a uniform flow field for the mean flow instead of a RANS solution file. If this flag is on, the user must also specify the following:
  - **PINF** - the freestream pressure
  - **RHOINF** - the freestream density
  - **RHOINF**- the Mach number of the flow
  - **V\_X,V\_Y**, and **V\_Z** - the x, y, and z component of the flow direction
- **ViscFlag** - Whether or not to calculate the viscous terms during the integration. Default is `.true.`. This should be set to `.false.` for inviscid calculations.
- **CLFLAG** - This flag turns on the calculation of a lift force. The force is calculated by integration of pressure and area over the jMin surface of the grid only! Must also specify **CLX,CLY**,and **CLZ** which are the unit normals in the lift direction of the body.
- **CDFLAG** - This flag turns on the calculation of a drag force. The force is calculated by integration of pressure and area over the jMin surface of the grid only! Must also specify **CDX,CDY**,and **CDZ** which are the unit normals in the drag direction of the body.
- **LowPassFilterFlag** - applies the low pass filter routine of Visbal and Gaitonde [35]. Can be used with **NBLOWPASSFILTERS** to define the number of times to apply the filter at each substep. Default is one time.

### Output options

The default output is a PLOT3D solution file consisting of the instantaneous conserved flow variables.

- **SolutionPath** - A character string locating the directory, relative to the case directory, where the solutions will be written to.

### Time and time stepping options

- `tMin` - Minimum time for the integration
- `tMax` - Maximum time for the integration
- `delt` - The time step value for the integration. This should not be used if `CFL` is specified.
- `CFL` - If specified, the largest time step allowable by stability criteria is calculated and used. Should not be used if specifying `delt`.
- `ntSteps` - The number of timesteps to take in the current simulation. If restarting from a previous run, this is the number of additional timesteps to take beyond the integer `restartStep`. At the conclusion of this amount of timesteps, a final solution file will be written along with a restart file "restart.q".
- `ntWrite` - This is the amount of timesteps between writing the solution files. If not specified, no intermediate solutions will be written.
- `dtWrite` - This is the amount of time between writing the solution files. Should not be used with `ntWrite`.

## **&GridIn**

Within the `GridIn` section, integers can be specified such as the number of boundary conditions, number of initial conditions, number of specific points to save the solution, and number of acoustic data surfaces. If any of these integers are specified, then immediately following the `GridIn` section must be their respective namelist, in the order listed above.

- `GridNumber` - this is the integer number of the grid in the `GridPath` file which the solution is performed on. The default is 1.
- `iMin, iMax, jMin, jMax, kMin, kMax` - The range of the grid solution to use. If any of these variables are not declared, the minimums are defaulted to 1 and the maximums are defaulted to the maximum in the grid file.

- **nbBase** - an integer specifying the amount of changes of base to apply to the grid. This is adopted from PSU-WOPWOP. Immediately following the **&GridIn** must be **nbBase** number of **&CB** inputs (see PSU-WOPWOP manual for change of base documentation).
- **nbBC** - an integer specifying the number of boundary conditions to be declared on the domain. A **&BCIn** section must immediately follow the **&GridIn** section for each boundary condition desired.
- **nbIC** - an integer specifying the number of initial conditions to be declared on the domain. An **&ICIn** section must immediately follow the **&BCIn** sections for each initial condition desired.
- **nbPoints** - an integer specifying the number of specific locations to save the flow variables at after each timestep. The file saved will be a Tecplot data file containing the flow variables on the Y axis and time on the X axis. If **nbPoints** is specified and greater than zero, this integer number of **&PointIn** sections must appear immediately following the **&ICIn** section, or the **&BCIn** section if no initial conditions are specified.
- **nbADS** - an integer specifying the number of acoustic data surfaces for a solution to be written to.

### **&CB**

- PSU-WOPWOP change of base commands can be used here for grid motion.

### **&BCIn**

This section must appear in the namelist for each desired boundary condition on the domain. Also, every grid point on the domain must have a boundary specified. If there are unspecified boundaries, the code will abort and tell the user. See the NLDE code user manual for a list of supported boundary conditions. If a boundary conditions is not supported, the code will alert the user and abort.

- **BCType** - an integer specifying the type of boundary condition to be applied.

- **BCFace** - a string specifying what face the boundary condition is applied on. Possible strings are 'iMin', 'iMax', 'jMin', 'jMax', 'kMin', and 'kMax'.
- **iMin, iMax** - the lower and upper integer limits of the boundary condition in the i-direction. These integers should only be specified if the **BCFace** is either an j-Face or a k-Face.
- **jMin, jMax** - the lower and upper integer limits of the boundary condition in the j-direction. These integers should only be specified if the **BCFace** is either an i-Face or a k-Face.
- **kMin, kMax** - the lower and upper integer limits of the boundary condition in the k-direction. These integers should only be specified if the **BCFace** is either an i-Face or a j-Face.

## &ICIn

Specific initial conditions can be imposed on the perturbation flow variables at the beginning of a run. The initial conditions must be stored as a double precision Fortran unformatted Q file. The solution file must be the same dimension as the input RANS grid. The initial condition files are assumed to be dimensionalized when created by the user.

- **Filename** - a character string specifying the name of the initial condition file, relative to the directory the case namelist is in. For example, './initial.save'. If a filename is specified, none of the following parameters should be specified.
- OR-
- **iMin, iMax, kMin, kMax, jMin, jMax** - integers specifying the bounds on a UNIFORMLY DISTRIBUTED initial perturbation field. If one of these is specified, all must be specified.
  - **qp1, qp2, qp3, qp4, qp5** - Floating point numbers specifying the perturbation to be imposed on the conserved flow vector within the index limits described above.

**&SPIn**

- **iIndex,jIndex,kIndex** - The  $i$ ,  $j$ , and  $k$  index of the specific point in the RANS grid for the solution to be written for. The solution will be written in Tecplot data file format, with the flow variables on the  $Y$  axis and time on the  $X$  axis.

**&ADSIIn**

- **ADSFileName** - the name/path of the acoustic data surface to be output.
- **ntADSwrite** - an integer specifying the timestep interval to write the acoustic data. If no value is specified, the acoustic data will be written at every timestep. This could lead to a very large file size! If **ntADSwrite** is negative, an individual file will be written at each specified **ntADSwrite** to be used with some type of post-processing converter. If **ntADSwrite** is positive, the file will be written for the entire solution at the end of the run. CAUTION, if the run is terminated before it is supposed to end, no file will be written and your work will be lost!
- **ADSiMin** and **ADSiMax** - the  $i$  Min and Max limit of the grid to use for the ADS
- **ADSjMin** and **ADSjMax** - the  $j$  Min and Max limit of the grid to use for the ADS
- **ADSkMin** and **ADSkMax** - the  $k$  Min and Max limit of the grid to use for the ADS

NOTE - either the  $i$ ,  $j$ , or  $k$  min and maximum levels must be the same for a surface to be defined.

**C.1.5 PLOT3D File Format**

PLOT3D file format is used throughout the code for CFD grids and solution files. The format is Fortran Unformatted, double precision. The following is a segment of Fortran source code which generates the correct grid and solution “Q” files:

Grid file:

```
integer::i,j,k,nbi,nbj,nbk
double precision,dimension(:,:,:)::x,y,z
open(unit=outUnit,file=pathName,status='replace',form='unformatted')
  write(outUnit)1
  write(outUnit)nbi,nbj,nbk
  write(outUnit)x,y,z
close(outUnit)
```

Solution “Q” file:

```
double precision,dimension(:,:,:)::q
double precision::mach,alpha,reyn,time
integer::nbi,nbj,nbk
open(unit=outUnit,file=pathName,status='replace',form='unformatted')
  write(outUnit)1
  write(outUnit)nbi,nbj,nbk
  write(outUnit)mach,alpha,reyn,time
  write(outUnit)q
close(outUnit)
```

## C.2 Example: 2-D Gauss Pulse Case File

The following is an example of the NLDE case file used to generate the results found in Section 4.1.

```
&EnvironmentIn
  OVERFLOWGridPath = './gauss_2d.x'
  SolutionPath      = 'Results/'
  debugLevel       = 3
  ntsteps          = 600
  CFL              = 0.5
  tMin             = 0.0
  ntWrite          = 50
  lowpassfilterflag=.true.
  UNIFORMFLOWFLAG =.TRUE.
  RHOINF          = 1.2d0
  MACH             = 0.0D0
  PINF            = 101.0D3
```

```
V_X           = 1.0D0
V_Y           = 0.0D0
V_Z           = 0.0D0
/
&OVERFLOWIn
  OVERFLOWGridNumber = 1
  nbBC               = 6
  nbIC               = 1
/
&BCIn
  bcfac = 'iMin'
  bctype = 55
/
&BCIn
  bcfac = 'iMax'
  bctype = 55
/
&BCIn
  bcfac = 'jMin'
  bctype = 55
/
&BCIn
  bcfac = 'jMax'
  bctype = 55
/
&BCIn
  bcfac = 'kMin'
  bctype = 21
/
&BCIn
  bcfac = 'kMax'
  bctype = 21
/
&ICIn
  fileName = 'gauss_2d.save'
/
```