Computational Acoustic Beamforming of Noise Source on Wind Turbine Airfoil

by

Chi Shing Li

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2014

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

A new method, Computational Acoustic Beamforming, is proposed in this thesis. This novel numerical sound source localization methodology combines the advantages of the Computational Fluid Dynamics (CFD) simulation and experimental acoustic beamforming, which enable this method to take directivity of sound source emission into account while maintaining a relatively low cost. This method can also aid the optimization of beamforming algorithm and microphone array design. In addition, it makes sound source prediction of large structures in the low frequency range possible.

Three modules, CFD, Computational Aeroacoustics (CAA) and acoustic beamforming, are incorporated in this proposed method. This thesis adopts an open source commercial software OpenFOAM for the flow field simulation with the Improved Delayed Detached Eddy Simulation (IDDES) turbulence model. The CAA calculation is conducted by an in-house code using impermeable Ffowcs-Williams and Hawkings (FW-H) equation for static sound source. The acoustic beamforming is performed by an in-house Delay and Sum (DAS) beamformer code with several different microphone array designs.

Each module has been validated with currently available experimental data and numerical results. A flow over NACA 0012 airfoil case was chosen as a demonstration case for the new method. The aerodynamics and aeroacoustics results are shown and compared with the experimental measurements. A relatively good agreement has been achieved which gives the confidence of using this newly proposed method in sound source localization applications.

Acknowledgements

I cannot express enough appreciation to my co-supervisors Prof. Fue-Sang Lien and Prof. Armaghan Salehian for their helpful advice and guidance. I would offer my gratitude to Prof. Siva Sivoththaman who funded this research. I would also like to thank Prof. William Melek and Prof. Zhongchao Tan for taking time from their busy schedule to be part of my reading committee. Finally, to my wife Ping, I would never have finished this without your patience and selfless support.

Dedication

I dedicate this thesis to my wife, Ping Ma, for her endless support, encouragement and love.

Table of Contents

List of Figures

List of Tables

Nomenclature

b	Beamforming signal
C	Courant number
c	Speed of sound
c_p	Pressure coefficient
D	Characteristic length/ Chord length
e	Internal energy
F	Face flux
$\boldsymbol{f}$	External force
f	Face connecting control volumes
G	Free space Green's function
k	Turbulence kinetic energy
M	Mach number
n_i	Normal vector tensor
P_{ij}	Compressive stress tensor
p	Pressure
$\mathrm{p_m}$	Pressure signal at microphone
Q	Q criterion
$\boldsymbol{q}$	Heat flux
$\boldsymbol{r}$	Radial direction vector
r	Gradient ratio
$\mathbf{r_m}$	Microphone location vector
Re	Reynolds number
$\boldsymbol{S}_f$	Area vector of face f
S_ϕ	Source term of scalar ϕ
T_{ij}	Lighthill tensor
t	Time
$\boldsymbol{u}$	Flow velocity vector
u	Flow velocity magnitude
u_i, u_j, u_k	Flow velocity tensors
V	Volume

$\boldsymbol{v}$	Surface velocity vector
$\mathrm{w_m}$	Weighting of microphone
$\boldsymbol{x}$	Observer location vector
x_i, x_j, x_k	Position tensor
$\boldsymbol{y}$	Source location vector

Γ_ϕ	Diffusion coefficient of scalar ϕ
Δ_m	Time delay
ψ_{VL}	Van Leer limiter function
δ_{ij}	Kronecker delta
ζ	Volume viscosity
$\boldsymbol{\kappa}$	Steering direction unit vector
μ	Dynamic Viscosity
μ_t	Eddy viscosity
ρ	Density
τ	Retarded time
ν	Kinematic viscosity
ϕ	General scalar

Subscripts/Accent

$_0$	Value at steady state
$_n$	Component in normal direction
$_\infty$	Free stream value
$'$	Fluctuation
$\widehat{\ }$	Unit vector
$\dot{\ }$	Time derivative
$\bar{\ }$	Average
$\square^2$	Wave operator

1 Introduction

The global wind report 2012 shows that the global wind power market grew by more than 10% compared to 2011, which represents that nearly 45 GW of new wind power have been brought on line in year 2012, bringing the total global wind power to 282.5 GW at the end of 2012, representing cumulative market growth of more than 19%, an excellent industry growth rate given the economic climate [1].

As an emerging industry, wind energy develops very fast, with the average annual growth rate at more than 20% over the last decade. Increasing number of installed wind turbines has been seeing across the world, which not only shows support for this type of renewable energy, but also draws objections to wind industry's further development.

People complain about the negative influence brought by the wind turbines, e.g., visual obstruction of the wind farm, environmental concerns about birds and bats killed by the wind turbines, noise generated by the wind turbines, etc. For wind turbine noise, there has not been a consensus about its relation to human health. Some people say that the level of wind turbine noise is very low and should not cause any problem. Some people report that they have developed some syndromes which are caused by long term exposure to wind turbine noise [2].

At the same time, more and more attention has been paid on the wind turbine noise research in both industrial and academic fields. Researchers are developing and experimenting different tools for wind turbine noise prediction and corresponding noise reduction methods. Inspired by one experimental method for noise source detection (acoustic beamforming), this thesis developed a new numerical method for the source simulation based on CFD and CAA calculation. The proposed new methodology is described in Section 1.2. Section 1.1 gives a brief research background of the current study.

1.1 Background

Wind Energy has come into the focus of the modern society as a clean alternative for power generation since the international oil crisis in 1973. The increase in oil price, the environmental pollution from the use of the fossil fuel, together with the shortage of the natural resources (oil, natural gas and coal), all of these factors have prompted the development of renewable energy including wind energy.

Larger wind turbines with higher capacity are being designed to meet the increasing demand of the market. Currently, the largest wind turbine around the world, Vestas V164, produces 8 MW of power, has an overall height of 220 m [3]. In the meantime, at least five companies around the world are developing 10+ MW wind turbines. In Ontario, the Minister of Energy is expecting to have wind energy account for 10% of total power generation by year 2030 [4].

However, the harvesting of wind energy is not without drawbacks. One major concern is the noise generated by the wind turbines. As the size of wind turbine grows, the noise it produces increases as well. In Canada, there are eight 100 MW wind farms located in the southern part of Ontario currently, with a total of 6736 wind turbines being built [5]. Some of them are built very close to the residential area as shown in Figure 1.1. Many complaints have been reported regarding the noise issue from individuals living in close proximity to wind turbine establishments. People are reporting sickness with the syndromes of headache, ringing in the ears, insomnia and dizziness. Noise problems have already been proven crucial in planning discussions on the siting of potential wind farms.

Figure 1.1: Wind farm near residence in Wolfe Island [5]

To tackle the wind turbine noise issue, the noise sources and mechanisms need to be understood thoroughly before the introduction of noise reduction methods. Wind turbine noise can be classified into two categories: mechanical noise and aerodynamic noise. Mechanical noise is the noise induced by the movement of mechanical components like gearbox and generator. This type of noise is transmitted along the structure of the turbine and radiated from external surfaces. Mechanical noise does not increase with the size of the wind turbine proportionally and can be reduced efficiently by well-known engineering methods like mufflers and enclosures.

On the other hand, aerodynamic noise, which increases as the wind turbine gets bigger, is caused by air flowing over the tower and the moving blades and the interaction between the moving blades and tower. The aerodynamic noise can be divided into three groups, low-frequency noise, inflow-turbulence noise and airfoil self-noise [6].

Low frequency noise lies in the frequency range between 20-200 Hz. It is caused when the rotating blade encounters localized flow deficiencies, e.g., the flow around a tower, inflow gradient, local stall, etc.

The inflow-turbulence noise is created by atmospheric turbulence interacting with the blades. It has not been fully understood. Acoustic measurements did not indicate any systematic effects on the produced noise due to changes in the inflow turbulence.

The last group of noise, airfoil self-noise, can be further classified into five categories: trailing edge noise, blade tip noise, laminar boundary layer vortex shedding noise, stalled noise and the noise due to the blade surface imperfection.

To reduce the noise level, one needs to find the noise source and also understand the mechanism of the noise generation and propagation. This thesis focuses on the methodology of the noise source localization.

Currently, there are a few computational and experimental methods to perform aerodynamic noise source localization.

The computational method used to determine aeroacoustic source is by using Computational Fluid Dynamics (CFD) method. The pressure time history at any point in the simulation domain can be obtained from CFD simulation [7]–[9]. This pressure fluctuation data can then be converted to sound pressure level (SPL) plot which shows the SPL variation at different frequencies, one can use this plot to determine the sound source level and the corresponding frequency range.

However, there are some limitations adopting the numerical method to localize the noise source. First, this method can only be applied on situations that the noise source is on the solid surface (wall boundary). For noise induced by turbulent flow in space, it is difficult to identify the noise source location because both noise generated and passing through that particular point in the flow field affects the pressure time history. Secondly, the CFD based computational method does not include the effect of directivity, nor does it account for the attenuation and shielding effect of surrounding structures.

The most common experimental method to locate noise source is acoustic imaging. The two main acoustic imaging techniques are near-field acoustic holography (NAH) [10][11] and acoustic beamforming [12]–[14] (the detailed discussion about these two methods will be shown in Chapter 4). Generally, NAH is adopted when the target size is small, the distance to target is close and the interested noise frequency falls in the low frequency range (< 200 Hz). On the other hand, acoustic beamforming is used when the target size is larger, located at a further distance and the interested noise frequency is in the mid to high frequency range (> 500 Hz) [15].

However, there are still limitations for using acoustic beamforming experimentally for source localization. First of all, the accuracy of beamforming result is highly dependent on the design and compatibility of beamformer algorithm and microphone array design (details will be shown in Section 4.2). Even though there are many guidelines and studies outlining the possible setups which can produce good result, a general solution which can fit in all situations is still not available. Hence, in order to achieve accurate results, the algorithm and array combination needs to be tailored for each particular case. This process is very time consuming and usually the number of microphone needed during the tailoring process is more than that for a particular measurement, which results in a high cost to complete the tailoring process during the experiment.

The contribution in this research is the proposal of a new method, the Computational Acoustic Beamforming method, which implements the acoustic beamforming theory in a simulation environment. This method combines the advantages of the relatively low cost CFD simulation and the widely accepted method of acoustic source localization by using acoustic beamforming. Section 1.2 describes this methodology.

1.2 Computational Acoustic Beamforming

The newly proposed method, Computational Acoustic Beamforming, takes advantage of the benefits from the computational and experimental methods while improving upon the

shortcomings from both methods. The basic idea is to apply the acoustic beamforming numerically. All the data needed and generated in the acoustic beamforming experiment are calculated using either CFD or Computational Aeroacoustics (CAA) technique. The flowchart of this new method is shown in Figure 1.2.

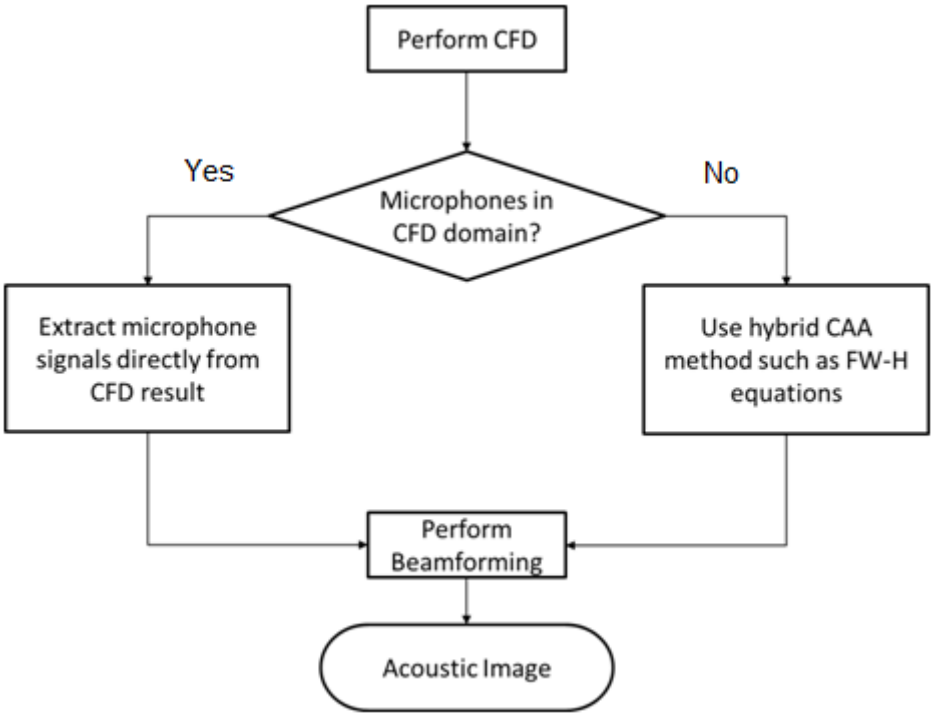

Figure 1.2: Flowchart of Computational Acoustic Beamforming

The preprocessing for the CFD simulation includes geometry and mesh generation. The pressure time history data can then be calculated and stored as the CFD simulation runs. If the microphone locations are within the CFD computational domain, the pressure data will be stored and passed directly to the acoustic beamforming solver. If the microphone locations are not within the CFD computational domain, the CAA technique — in this thesis, the impermeable Ffowcs Williams and Hawkings (FW-H) formulation — is adopted to calculate the pressure fluctuation data at the microphone locations. The output of the FW-H solver serves as the input for the acoustic beamforming solver, which generates the beam pattern at each interested frequency band. The results are then projected onto the plane of interest to obtain the acoustic image.

Due to the use of the CAA technique, the Computational Acoustic Beamforming method allows the microphones to be placed outside of the CFD computational domain. This feature can decrease the size of the computational domain so that the CFD simulation will be less computational demanding. Furthermore, the use of the beamforming method generates an acoustic image that is direction dependent, i.e., different microphone array locations give different results.

The newly proposed method, Computational Acoustic Beamforming, can also be adopted during the microphone array design stage before carrying out the acoustic beamforming measurement. One can test different beamformers and microphone array patterns and also the combinations of those two by running different simulations. With the rapid advancement in computational power, the cost of this simulation would be much lower than actually testing those different settings experimentally. Once an optimized arrangement is found, only one set of experimental measurements needs to be performed and the experimental data, in turn, can be used to validate the simulation results.

Another potential development from this idea is to use NAH instead of acoustic beamforming in the proposed method. The limitations of NAH described in Section 4.1, such as microphone array size and density, do not exist in the numerical environment. This allows NAH to be performed with larger object at a higher frequency in the virtual environment. However, this method is not studied in this thesis due to the availability of experimental results.

As one can see from Figure 1.2, there are three modules in Computational Acoustic Beamforming. In this thesis, each module will be discussed in detail in each chapter. Chapter 2 describes the CFD methodology and introduced the software used for the simulation. Chapter 3 gives a brief review of the CAA method and shows the derivation of the FW-H impermeable formulation. Chapter 4 reviews different acoustic source localization methods and introduces different

beamforming algorithms. The validations of CAA and acoustic beamforming modules are shown in the first part of Chapter 5 (Section 5.1 and 5.2). In the second part of Chapter 5 (Section 5.3), Computational Acoustic Beamforming is performed on a flow over airfoil case. The results are compared with the experimental data followed by the discussion and analysis. Chapter 6 draws the conclusion of the current study and also lists the future work for the improvement.

2 CFD Methodology

CFD technique is adopted in this thesis for the flow field simulation. It uses Finite volume method (FVM) to evaluate the governing equations in the form of algebraic equations. FVM is conservative and easy to apply on unstructured mesh. This method is used in many CFD commercial packages. In this thesis, an open source code, OpenFOAM, is chosen for the flow field simulation. Section 2.1 explains the governing equations of Newtonian fluid. Section 2.2 provides a brief introduction to turbulence models. Section 2.3 explains the solver used in this research.

2.1 Governing Equations

The governing equations for three dimensional, unsteady and compressible Newtonian fluid are the conservation of mass, conservation of momentum and conservation of energy as shown below:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \boldsymbol{u} = 0; \tag{2.1}$$

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = -\nabla p + \nabla \cdot (\mu \nabla \boldsymbol{u}) + \left(\zeta + \frac{\mu}{3}\right) \nabla (\nabla \cdot \boldsymbol{u}) + \boldsymbol{f}; \tag{2.2}$$

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} e) = -\nabla \cdot (\rho \boldsymbol{u}) + (\mu \boldsymbol{u} \nabla \boldsymbol{U}) + \rho \boldsymbol{f} \cdot \boldsymbol{u} - \nabla \cdot \boldsymbol{q}, \tag{2.3}$$

where ρ is density, t is time, $\boldsymbol{u}$ is velocity, p is pressure, μ is dynamic viscosity, ζ is volume viscosity, $\boldsymbol{f}$ is external force, e is internal energy and $\boldsymbol{q}$ is heat flux.

The maximum Mach number, $M = u/c$, in the demonstration case shown in Chapter 5 is less than 0.1. Theoretically, the flow can be assumed incompressible when the Mach number is less than 0.3. In this study, a constant and homogenous density assumption is made throughout the whole computational domain. However, note that for incompressible flow, the density field needs not to be uniform. The density can vary due to stratification but the flow can still be treated as

incompressible. The incompressible flow only requires that the density of a fluid element does not change in time as it moves through space.

For incompressible flow, the energy equation is decoupled from the continuity and momentum equations. Due to the insignificant temperature change in the test case, the influence of the temperature change can be neglected, so that the isothermal condition is assumed.

The constant density, isothermal assumption together with neglecting the external forces simplified Equation (2.1) to (2.3) into

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2.4}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u}\boldsymbol{u}) = -\nabla p + \nabla \cdot (\nu \nabla \boldsymbol{u}), \tag{2.5}$$

where $\nu = \frac{\mu}{\rho}$ is kinematic viscosity. The left hand side of Equation (2.5) represents the change in momentum of fluid element owing to the unsteadiness in the flow and the convection by the flow. This change is balanced by the two terms shown in right hand side of Equation (2.5): the isotropic stress owing to the pressure field and the viscous stresses.

2.2 Turbulence Models

2.2.1 Direct Numerical Simulation (DNS)

The simplified governing equations as shown in Equation (2.4) and (2.5) can be solved directly. However, in order to capture the flow characteristics of all scales in the whole domain, there are strict rules on the spatial and temporal discretization. As a rule of thumb for DNS, the number of grid needs to be the same or higher than $Re^{9/4}$, and the time step is the same or smaller than $Re^{-1/2}$, where Re is the Reynolds number. Reynolds number is defined as

$$Re = \frac{\rho u D}{\mu}, \tag{2.6}$$

where D is the characteristic length of the solid body, which represents the chord length in the demonstration case in Chapter 5.

For a typical turbulent flow where Re > 10^6, the computing power required to run DNS is prohibitive. Therefore different turbulence modeling closures have been proposed.

2.2.2 Reynolds Averaged Navier Stokes model (RANS)

The RANS equations are derived from the instantaneous Navier-Stokes equations using Reynolds decomposition, which decomposes an instantaneous quantity into its time-averaged and fluctuating quantity. Using ensemble averaging, the momentum equation becomes

$$\rho\bar{u}_j\frac{\partial\bar{u}_i}{\partial x_j} = \rho\overline{f_i} + \frac{\partial}{\partial x_j}\left[-\bar{p}\delta_{ij} + \mu\left(\frac{\partial\bar{u}_i}{\partial x_j} + \frac{\partial\bar{u}_j}{\partial x_i}\right) - \rho\overline{u_i'u_j'}\right], \quad \textbf{(2.7)}$$

where δ_{ij} is Kronecker delta. The term $(-\rho\overline{u_i'u_j'})$ is the Reynolds stress tensor. This nonlinear term requires additional modeling to close the RANS equation for solving, and has led to the creation of many different turbulence models.

Joseph Boussinesq in 1877 postulated that the momentum transfer caused by turbulent eddies can be modeled with an eddy viscosity, which presents a simple relationship between Reynolds stresses and the mean flow [16]. Boussinesq assumption states that the Reynolds stress tensor is proportional to the trace-less mean strain rate tensor and can be written as

$$-\rho\overline{u_i'u_j'} = \mu_t\left(\frac{\partial\bar{u}_i}{\partial x_j} + \frac{\partial\bar{u}_j}{\partial x_i} - \frac{2}{3}\frac{\partial\bar{u}_k}{\partial x_k}\delta_{ij}\right) - \frac{2}{3}\rho k\delta_{ij}, \quad \textbf{(2.8)}$$

where μ_t is a scalar property called the eddy viscosity and k is turbulence kinetic energy. For incompressible flow, the third term in the parent is zero.

Various models have been proposed with the Boussinesq hypothesis to solve the eddy viscosity. They are classified in terms of number of transport equations solved in addition to the RANS equations. Table 2.1 provides selected examples of eddy viscosity models.

Table 2.1: Different eddy viscosity models [17]

Zero-equation/Algebraic models	Mixing Length, Cebeci-Smith, Baldwin-Lomax
One-equation models	Wolfstein, Spalart-Allmaras, k-model
Two-equation models	k-ε, k-ω, k-τ, k-L

2.2.3 Large Eddy Simulation (LES)

Large Eddy Simulation resolves large scales of the flow field solution allowing better fidelity than alternative approaches such as RANS methods. It also models the smallest (and most expensive) scales of the solution, rather than resolving them as DNS does.

LES was first proposed by Joseph Smagorinsky to simulate atmospheric air currents [18]. Kolmogorov's theory of self-similarity imlpies that the large eddies of the flow are dependent on the geometry while the smaller scales are more universal [19]. This feature allows one to explicitly solve for the large eddies in a calculation and implicitly account for the small eddies by using a subgrid-scale model (SGS model), in order to obtain a solution that is representative of the physical phenomena while keeping the computational cost down.

2.2.4 Hybrid RANS-LES

2.2.4.1 Detached Eddy Simulation (DES)

The original DES utilizes the Spalart-Allmaras RANS model for small turbulence scale region close to a wall and uses the SGS model for other small turbulence scale region that is away from the wall [20]. The model is switched by means of a limiter, which compares the distance to wall to the local grid-spacing. This allows a smooth transition of RANS-solved attached boundary layer to the SGS-solved separated flow region.

However, this method is very dependent on the mesh quality near the wall. If the mesh is small enough to trigger the SGS model in the boundary layer while too coarse to actually support the

solving of the boundary layer using LES, it cannot capture all the velocity fluctuations. Moreover the eddy viscosity will be reduced, as well as the modeled Reynolds stresses, without the introduction of resolved stresses to restore the balance. This phenomena is called Modeled Stress Depletion (MSD) in literature [21]. This unbalanced stresses then reduce the skin friction and lead to Grid Induced Separation (GIS).

2.2.4.2 Delayed Detached Eddy Simulation (DDES)

The Delayed Detached Eddy Simulation (DDES) was formulated in order to avoid the effect of MSD when an ambiguous grid mentioned in Section 2.2.4.1 is used in DES [22]. The DDES is a simple modification of the classical DES. The key point is that it uses some blending functions to define the length scale instead of a simple limiter. If such blending functions indicate that a point is inside the boundary layer, it refuses to transfer to LES mode [23]. As a consequence, the transition between RANS and LES will be more abrupt. This means that DDES detects boundary layers and maintains the full RANS mode inside, instead of comparing the grid spacing alone. This method of detecting the boundary layer depends on the turbulence viscosity, which is part of the solution.

2.2.4.3 Improved Delayed Detached Eddy Simulation (IDDES)

The objective of this model is to combine the advantages of the Wall Modeled LES (WMLES), and the DDES capabilities [24]. If the inlet condition contains unsteady turbulent flow, the WMLES is adopted; otherwise the regular DDES is used. The WMLES allows solving turbulent wall boundary flow at higher Reynolds number without over-refinement of the mesh. For instance, the wall-stress model proposed by Schumann considered the first off-wall points in the logarithmic layer and used empirical derived wall functions together with velocities to calculate an estimate for wall stresses at the boundary [25].

2.3 OpenFOAM

The CFD software package used in this research is OpenFOAM, which is an open source CFD software and is widely used in most areas of engineering and science. The reasons to choose this CFD package are as follow:

1. OpenFOAM is a free software;
2. Parallelization can be set up with ease.

For most commercial CFD packages, there is a very high cost in licensing to run a highly parallelized job, like LES. OpenFOAM is a very cost effective alternative and thus gained popularity in both academia and industry. Section 2.3.1 to 2.3.4 explains the discretization scheme and solvers used in OpenFOAM in this thesis.

2.3.1 Spatial Discretization

OpenFOAM uses the FVM and stores the flow properties, such as pressure, velocity, etc. at the center of the control volume (CV). This co-located methodology can be used on an unstructured polyhedral grid with arbitrary grid elements. A variety of available interpolation, discretization schemes and matrix solution methods is available in the OpenFOAM package and can be selected at runtime. One can choose different schemes to apply on each term in the governing equations. Although optimizing these schemes could have resulted in more accurate or faster computation, it is beyond the scope of this research. This thesis only presents the settings which were used for this research.

FVM is based on a discretization of integral form of the conservation equations. The computational domain is divided into a set of discrete control volumes, then the discretized integral forms of the conservation equations are applied at each of the many contiguous control volumes. This results in a linear algebraic system of N equations for the N unknowns stored in the CV center.

Figure 2.1 shows an example of a CV, where *P* and *N* are the centroids of two control volumes with the distance d between them, *f* represents the face connecting these two control volumes with area of S_f.

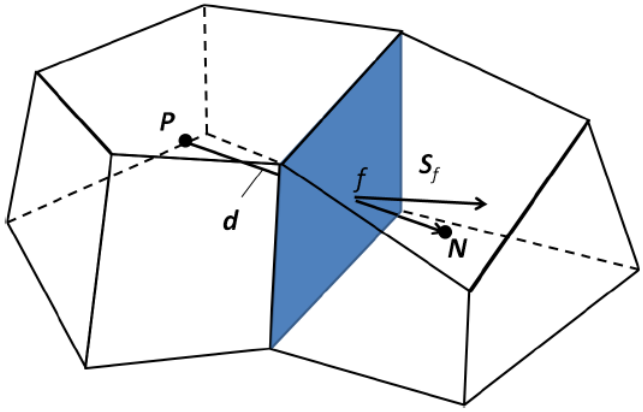

Figure 2.1: Neighboring control volumes [26]

Consider a general transport equation of scalar ϕ,

$$\frac{\partial \rho\phi}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}\phi) = \nabla \cdot \left(\rho\Gamma_\phi \nabla\phi\right) + S_\phi(\phi), \quad \textbf{(2.9)}$$

where Γ_ϕ is diffusion coefficient and $S_\phi(\phi)$ is the source term.

The second term from the left hand side is the convection term and the two terms on the right hand side represents diffusion and source terms, respectively. With the FVM, integrating Equation (2.9) in time and space yields

$$\int_t^{t+\Delta t} \left[\frac{\partial}{\partial t}\int_{V_P} \rho\phi dV + \int_{V_P} \nabla \cdot (\rho \boldsymbol{u}\phi) dV\right] dt = \quad \textbf{(2.10)}$$

$$\int_t^{t+\Delta t} \left[\int_{V_P} \nabla \cdot \left(\rho\Gamma_\phi \nabla\phi\right) dV + \int_{V_P} S_\phi(\phi) dV\right] dt.$$

Here, the flow quantities, such as velocity and pressure, are assumed to vary linearly. The volume integral can be evaluated with the surface integral using the following relations:

Integral of variable: $$\int_{V_P} \phi dV = \phi_P V_P; \quad (2.11)$$

Integral of divergence: $$\int_{V_P} \nabla \cdot (\boldsymbol{u}\phi) dV = \sum \boldsymbol{S}_f (\boldsymbol{u}\phi)_f ; \quad (2.12)$$

Integral of gradient: $$\int_{V_P} \nabla \phi dV = \sum \boldsymbol{S}_f \phi_f . \quad (2.13)$$

Based on the divergence theorem shown in Equation (2.12), the convection term in Equation (2.10) can be evaluated as

$$\int_{V_P} \nabla \cdot (\rho \boldsymbol{u}\phi) dV = \sum_f \boldsymbol{S}_f \cdot (\rho \boldsymbol{u}\phi)_f = \sum_f F \, \phi_f , \quad (2.14)$$

where $F = \boldsymbol{S} \cdot (\rho \boldsymbol{U})$, is the face flux through the control volume P. ϕ_f is the face value which is approximated by the centre values of the property ϕ from the neighboring control volumes. Different discretization schemes can be used here to calculate the face values of control volume P based on different order of accuracy and stability requirements.

For example, the Van Leer limiter is adopted for the face values evaluation of the convective term in the turbulence transport equation in this thesis [27].

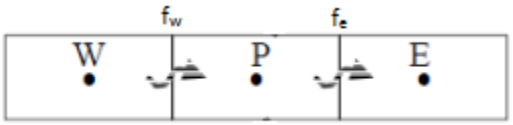

Figure 2.2: Three neighboring control volumes and their interfaces

As shown in Figure 2.2, the face value at the east face of control volume P is calculated using

$$\phi_{f_e} = \phi_P + 0.5\psi(r)(\phi_P - \phi_W), \quad (2.15)$$

where r is the gradient ratio and defined as

$$r = \frac{\phi_E - \phi_P}{\phi_P - \phi_W}, \quad (2.16)$$

and the Van Leer limiter function is defined as

$$\psi_{VL}(r) = \frac{r + |r|}{1 + |r|}. \quad \textbf{(2.17)}$$

Applying Equation (2.13), the diffusion term on the right hand side of Equation (2.10) becomes

$$\int_{V_P} \nabla \cdot (\rho\Gamma_\phi \nabla\phi) dV = \sum_f \boldsymbol{S}_f \cdot (\rho\Gamma_\phi \nabla\phi)_f = \sum_f (\rho\Gamma_\phi)_f \boldsymbol{S}_f \cdot (\nabla\phi)_f. \quad \textbf{(2.18)}$$

For incompressible flow, $\rho\Gamma_\phi = \mu$, which is the dynamic viscosity of the fluid, is a constant. When the mesh is non-orthogonal, an additional term is added to the dot product shown in Equation (2.18) to account for the non-orthogonal effect. Note that when the mesh has high non-orthogonality, this added term's value can be very large and can cause some stability issue. Thus a limiter is usually used with it.

2.3.2 Temporal Discretization

For the temporal discretization, there are several schemes to choose from. The simplest way is the explicit scheme,

$$\frac{\phi^n - \phi^{n-1}}{\Delta t} = F(\phi^{n-1}). \quad \textbf{(2.19)}$$

The value of ϕ at the next time step is only dependent on the value from the previous time step. Though straightforward, this method only has first order accuracy and has stability issue when Courant condition,

$$C = \frac{u\Delta t}{\Delta t} \leq C_{max} \quad \textbf{(2.20)}$$

where $C_{max} = 1$, is violated. To overcome the stability issue in the explicit scheme and achieve a higher order of accuracy, this thesis adopts the second order implicit backward differencing scheme to discretize the unsteady term in Equation (2.10), which is formulated as

$$\frac{\frac{3}{2}^n \phi^n - 2\phi^{n-1} + \frac{1}{2}\phi^{n-2}}{\Delta t} = F(\phi^n). \quad \textbf{(2.21)}$$

This implicit time discretization scheme allows the Courant number to be higher than unity while still remains stable during the simulation.

2.3.3 Pressure-Velocity Coupling

In this thesis, the Pressure Implicit with Splitting of Operators (PISO) algorithm is adopted in the transient simulation for incompressible flow. PISO algorithm was first proposed by Issa in 1986 [28]. The basic idea of PISO is that it freezes the discretization of the momentum equation while repeating a number of pressure correctors. In such setup, the first pressure corrector will create a conservative velocity field, while the second and followings will establish the pressure distribution. However, since the multiple pressure correctors are used with a single momentum equation, which is only true when the Courant number is small, thus the pressure-velocity coupling is much stronger than the non-linear coupling. In this thesis, two pressure corrections are specified in solver settings in OpenFOAM. The process in PISO can be summed up as follows:

1) Set the boundary conditions;
2) Solve the discretized momentum equation to compute an intermediate velocity field;
3) Compute the mass fluxes at the cells faces;
4) Solve the pressure equation;
5) Correct the mass fluxes at the cell faces;
6) Correct the velocities on the basis of the new pressure field;
7) Update the boundary conditions;
8) Repeat from 3) for the prescribed number of times;
9) Increase the time step and repeat from 1).

2.3.4 Solver

The governing equations describe the fluid flow as partial differential equations (PDE). These PDE's can be expressed as a set of algebraic equations for the whole computational domain by

using any of the available discretization schemes. When solving real life problem, like the flow over an airfoil demonstration case shown in Section 5.3, a very large number of equations is required to be solved. Therefore, an iterative solver is adopted.

A great amount of computational savings can be achieved by setting the solvers, preconditioners and tolerances in a suitable way. The equation solvers, tolerances and algorithms are controlled in the fvSolution dictionary in OpenFOAM, which is also included in Appendix III in this thesis.

The pressure equation is solved using the preconditioned conjugate gradient (PCG) solver with generalized geometric-algebraic multi-grid (GAMG) preconditioner and a Gauss-Seidel smoother with 4 sweeps specified. The main idea of multigrid method is to accelerate the convergence of a basic iterative method by global correction from time to time, accomplished by solving a coarse problem.

The velocity and modified turbulence viscosity terms are solved using Preconditioned Bi-Conjugate Gradient (PBiCG) solver for asymmetric LDU matrices, where LDU refers to Lower triangular, Diagonal, Upper triangular.

3 Computational Aeroacoustics

Computational aeroacoustics (CAA) is a branch of aeroacoustics that analyzes the generation and propagation of noise by turbulent flows through numerical methods. Relative to CFD, CAA is a young research area and has developed rapidly during the past two decades. CAA algorithms have found applications in many areas of aeroacoustics, like in jet noise, airframe noise, fan and turbomachinery noise, propeller and helicopter noise, duct acoustics, interior noise, sonic boom, etc. [29]–[36].

3.1 Background

Computational techniques for flow-generated sound can be classified into two broad categories: direct computation and hybrid computation.

Direct computation simulates the sound generation and propagation together with its fluid dynamic source field by solving the compressible flow governing equations. It takes into account of the influence from flow to acoustic field as well as the effect of acoustics on the flow field. Both DNS and LES technique can be used in direct computation. DNS, which avoids any modeling approximations and resolves all flow scales, provides a tool for studying sound generation mechanisms and generating databases for the development and evaluation of sound prediction models. However, due to its Reynolds number limitation (as shown in Section 2.2.1), the use of DNS is restricted to simple flow configurations at low to moderate Reynolds number. On the other hand, LES, which resolves only the dynamically important flow scales and models the effects of smaller scales, can extend this range of applicability and is still an area of active research. However, there are two reasons which prohibit the use of the direct computation method in most CAA applications:

1. A large computational domain is needed which must include all the sound sources of interest and the receiver locations. This can lead to prohibitive computer storage and unrealistic CPU time;
2. A scheme needs to be developed with extremely low numerical noise to compute sound waves accurately. The sound intensity is often five to six orders of magnitude smaller compared to the mean flow. The second order accurate finite volume schemes currently used in CFD may sabotage the aeroacoustics prediction due to its mathematical dissipation and dispersion behavior.

To tackle the second difficulty in the direct computation method in CAA, Tam et al. developed a wave number based scheme, dispersion-relation-preserving (DRP) finite difference scheme [37][38]. Using the same strategy of DRP scheme, Kim et al. derived an optimized implicit finite difference compact scheme utilizing the original high-order compact finite difference schemes of Lele [39][40]. Both schemes have the characteristics of low dissipation and dispersion error which are needed for wave problems. They applied their schemes in some CAA areas, like jet noise, fan noise and airframe noise, and achieved good agreement with the experimental data. However, besides the substantial computer resources needed (the first difficulty in the direct computation method in CAA) which restricts this method only to near-field prediction, the non-reflecting outflow boundary conditions, the complex geometries as well as the efficiency and accuracy of turbulence models provide new challenges to these numerical schemes.

On the other hand, a hybrid method allows the computation of flow to be decoupled from the computation of sound. It separates the computation domain into two regions, one describing the generation of sound (near-field), and the other describing the propagation of sound (far-field). This one-way coupling method leads to the major assumption of a hybrid method: the unsteady flow generates sound and modifies its propagation, but the sound waves do not affect the flow in any significant way. Thus, the principal application of the hybrid approach lies in flows at low fluctuating Mach numbers (weakly compressible) [41].

For the calculation of the sound generation (near-field), both steady state (RANS, SNGR (Stochastic Noise Generation and Radiation) [42]), and transient (DNS, LES, DES, URANS) fluid field solutions can be used. Once the sound sources have been identified, the acoustic propagation can be computed. There are several strategies for the acoustic propagation calculation, like LEE [43], integral method [44]–[46], EIF (Expansion about Incompressible Flow) [47][48], APE (Acoustic Perturbation Equations) [49], etc. Since this thesis uses the integral method to calculate the sound pressure level at the microphone position for the demonstration case, below gives a short review of the integral method and a brief derivation of the FW-H equation used in this thesis.

Generally speaking, integral method can be classified into two categories: Kirchhoff method and acoustic analogy [50].

The Kirchhoff method is an approach to noise problems which takes advantage of the mathematical similarities between the aeroacoustics and electrodynamics equations [51]. It assumes that all sound sources are computed correctly by a CFD code and that the acoustic waves are propagated to a surface that surrounds all sources. However, this method requires the integral surface, which separates the sound sources and propagation calculations, to include all sound sources and be located in the linear wave region [52]. This is very hard to achieve in general. Ideally, the surface should be placed well away from the source region [50][53], but CFD solutions typically are not as well resolved or as accurate in regions of the flow field away from the body.

The other alternative using the integral technique is the acoustic analogy, which was established by Lighthill in 1952 [44]. The basic idea is to rewrite the governing N-S equations into an inhomogeneous wave equation form as

$$\frac{\partial^2 \rho}{\partial t^2} - c_0^2 \nabla^2 \rho = \frac{\partial^2 T_{ij}}{\partial x_i x_j}, \quad \textbf{(3.1)}$$

where the left hand side is a wave operator, the right hand side is identified as the acoustic source in the fluid flow, T_{ij} is the Lighthill tensor. Note that the left hand side is only applicable for ordinary waves. For problems where isotropic wave operator is not appropriate, Lighthill's equation is not applicable.

As an extension to Lighthill's acoustic analogy, Curle proposed a formal solution to Lighthill's analogy in 1955 to include the influence of the solid static boundaries [54]. This additional contribution is reflected in a new term $\frac{\partial f_i}{\partial x_i}$ on the right hand side of Equation (3.1), which represents the dipole type of noise.

In 1969, Ffowcs Williams and Hawkings generalized Lighthill's acoustic analogy approach to account for effects of generic surfaces and motions [55]. Typical applications for this analogy are the noise prediction of fans, helicopter blades and full scale wind turbines [56][57].

It is known that at low Mach numbers, the dipole sources are more efficient noise mechanisms than the quadrupole sources. Lighthill's eighth power law suggests that the radiated power or intensity of sound generated by quadrupole sources varies with the eighth power of the mean velocity whereas for dipole, it is proportional to sixth power of the mean velocity. The boundary surfaces will reflect and diffract the radiated sound, changing wave characteristics. Consequently, the acoustic field is often calculated by the contribution of the dipole sources only, and neglecting the quadrupole sources for low speed flow.

In this thesis, the FW-H acoustic analogy is adopted for the calculation of the sound propagation for the demonstration case shown in Chapter 5. Below gives a brief derivation of the FW-H equation and the physical meaning of each term.

3.2 Derivation

The derivation of the FW-H equation has been conducted by a research group member [58]. The relevant portion has been summarized in this section.

Consider a generic body immersed in a fluid, and the surface S of which is described by the equation $f(\boldsymbol{x}, t) = 0$, where $f < 0$ for points inside the body, $f = 0$ for points on the body surface and $f > 0$ for points outside the body. The problem can be modeled by replacing the body by a fluid at rest ($p' = 0,\ \rho = \rho_0, \boldsymbol{u} = 0$). The flow parameters will then have artificially introduced discontinuities across the surface $f = 0$. Ffowcs Williams and Hawkings rewrote the continuity and momentum equations (Equation (2.1) and (2.2) in Section 2.1) using generalized function into

$$\frac{\bar{\partial}\rho}{\partial t} + \frac{\bar{\partial}}{\partial x_i}(\rho u_i) = [\rho(u_n - v_n) + \rho_0 v_n]\delta(f), \tag{3.2}$$

$$\frac{\bar{\partial}}{\partial t}(\rho u_i) + \frac{\bar{\partial}}{\partial x_j}\left(\rho u_i u_j + P_{ij}\right) = \left[\rho u_i(u_n - v_n) + \Delta P_{ij} n_j\right]\delta(f), \tag{3.3}$$

where $\bar{\partial}$ denotes generalized differentiation, the subscript 0 represents the fluid properties at rest, $u_n = u_i n_i$ is the local fluid velocity in the direction normal to the boundary surface, $n_i = \partial f / \partial x_i$ is the component of the unit outward normal to $f = 0$, $v_n = -\partial f / \partial t$ is the local normal velocity of the boundary surface, P_{ij} is the compressive stress tensor with constant $p_0 \delta_{ij}$ subtracted.

The generalized function makes the conservation laws valid throughout the whole domain. The reason for this extension is that the wave equation can be valid in the entire three-dimensional space so that a simple free space Green's function can be used on the wave equation in unbounded space.

Assuming no external forces and mass sources, one can take $\bar{\partial}/\partial t$ of both sides of Equation (3.2), and take $\bar{\partial}/\partial x_i$ of both sides of Equation (3.3). This gives

$$\frac{\bar{\partial}^2\rho}{\partial t^2}+\frac{\bar{\partial}}{\partial x_i\partial t}(\rho u_i)=\frac{\bar{\partial}}{\partial t}\{[\rho(u_n-v_n)+\rho_0 v_n]\delta(f)\}, \quad (3.4)$$

$$\frac{\bar{\partial}}{\partial x_i\partial t}(\rho u_i)+\frac{\bar{\partial}}{\partial x_i\partial x_j}\left(\rho u_i u_j+P_{ij}\right)=\frac{\bar{\partial}}{\partial x_i}\left\{\left[\rho u_i(u_n-v_n)+\Delta P_{ij}n_j\right]\delta(f)\right\}. \quad (3.5)$$

Subtract Equation (3.5) from (3.4) yields

$$\frac{\bar{\partial}^2\rho}{\partial t^2}-\frac{\bar{\partial}}{\partial x_i\partial x_j}\left(\rho u_i u_j+P_{ij}\right) \quad (3.6)$$

$$=\frac{\bar{\partial}}{\partial t}\{[\rho(u_n-v_n)+\rho_0 v_n]\delta(f)\}$$

$$-\frac{\bar{\partial}}{\partial x_i}\left\{\left[\rho u_i(u_n-v_n)+\Delta P_{ij}n_j\right]\delta(f)\right\}.$$

Subtracting $\bar{\nabla}^2[c^2(\rho-\rho_0)]$ from both sides of Equation (3.6) yields the permeable FW-H equation in differential formulation:

$$\bar{\Box}^2[c^2(\rho-\rho_0)]=\frac{\bar{\partial}}{\partial t}\{[\rho(u_n-v_n)+\rho_0 v_n]\delta(f)\} \quad (3.7)$$

$$-\frac{\bar{\partial}}{\partial x_i}\left\{\left[\rho u_i(u_n-v_n)+\Delta P_{ij}n_j\right]\delta(f)\right\}+\frac{\bar{\partial}^2}{\partial x_i\partial x_j}\left[T_{ij}H(f)\right],$$

where the wave operator notation $\bar{\Box}^2=\left[(1/c^2)\left(\bar{\partial}^2/\partial t^2\right)\right]-\bar{\nabla}^2$, $T_{ij}=\rho u_i u_j+P_{ij}-c^2\rho'$, $\rho'=\rho-\rho_0$. If the perturbations are small at the observer location, the term $c^2(\rho-\rho_0)$ can be replaced by p' according to the linear wave propagation.

Equation (3.7) can be viewed as an inhomogeneous wave equation with the left hand side representing the wave propagation and the right hand side representing the noise sources. The first term at the right hand side refers to the monopole type of source, also called thickness noise. The monopole source term models the noise generated by the displacement of fluid as the body passes. The second term at the right hand side of Equation (3.7) refers to the dipole type of noise source, also called the loading noise. The dipole source term models the noise that results from

the unsteady motion of the force distribution on the body surface. Both of these sources are surface sources as indicated by the presence of the Dirac delta function $\delta(f)$ in Equation (3.7). The last term on the right hand side in Equation (3.7) is a quadrupole source term that acts throughout the volume that is exterior to the boundary surface $f = 0$ as indicated by the Heaviside function $H(f)$.

As mentioned above, for low Mach number flows, the quadrupole source is much less efficient than dipole source. The radiated sound intensity is proportional to the sixth power of the mean flow velocity for dipole type of noise whereas it is proportional to the eighth power for quadrupole type of noise. In the demonstration case in Section 5.3, the maximum Mach number throughout the whole domain is less than 0.1. Hence the flow can be deemed as incompressible flow and the quadrupole type of noise is well dissipated at the microphone array location. Thus it is reasonable to neglect the quadrupole source term for the acoustic calculation.

To realize the advantages of the FW-H equation, an integral formulation of the FW-H equation can be obtained with the use of the free-space Green's function $G = \delta(g)/4\pi r$, where $r = |\boldsymbol{x} - \boldsymbol{y}|$, $g = \tau - t + r/c$, τ is the retarded time, $\boldsymbol{x}$ is the observer location and $\boldsymbol{y}$ is the source location. Convolving the free-space Green's function with the monopole and dipole source terms respectively, for non-deformable surfaces, the integral formulation of the permeable FW-H equation can be written as

$$p'(\boldsymbol{x}, t) = p'_T(\boldsymbol{x}, t) + p'_L(\boldsymbol{x}, t), \tag{3.8}$$

$$4\pi p'_T(\boldsymbol{x}, t) = \int_{f=0} \left[\frac{\rho_0(\dot{U}_n + U_{\dot{n}})}{r(1 - M_r)^2}\right]_{ret} dS + \int_{f=0} \left[\frac{\rho_0 U_n(r\dot{M}_r + c(M_r - M^2))}{r^2(1 - M_r)^3}\right]_{ret} dS, \tag{3.9}$$

$$4\pi p_L'(\boldsymbol{x},t) = \frac{1}{c}\int_{f=0}\left[\frac{\dot{L}_r}{r(1-M_r)^2}\right]_{ret} dS + \int_{f=0}[\frac{L_r - L_M}{r^2(1-M_r)^2}]_{ret} dS \quad \textbf{(3.10)}$$

$$+\frac{1}{c}\int_{f=0}[\frac{L_r(r\dot{M}_r + c(M_r - M^2))}{r^2(1-M_r)^3}]_{ret} dS,$$

where $U_n = U_i\hat{n}_i$, $\dot{U}_n = \dot{U}_i\hat{n}_i$, $U_{\dot{n}} = U_i\dot{\hat{n}}_i$, $U_i = \left(1 - \frac{\rho}{\rho_0}\right)v_i + \frac{\rho u_i}{\rho_0}$, $\hat{n}_i$ is the component of the outward unit vector normal to surface, $\dot{L_r} = \dot{L}_i\hat{r}_i$, $L_r = L_i\hat{r}_i$, $L_M = L_iM_i$, $L_i = P_{ij}\hat{n}_j + \rho u_i(u_n - v_n)$, $M_r = M_i\hat{r}_i$ is the Mach number of the point on the boundary surface in the radial direction at the time τ, $\hat{r_i} = \frac{\partial r}{\partial x_i}$ is the component of unit radial vector $\frac{(\boldsymbol{x}-\boldsymbol{y})}{r}$. The subscript T and L stand for the thickness noise (monopole source) and loading noise (dipole source) respectively. The subscript ret stands for the retarded time.

In order to use the integral formulation of the FW-H equation, an integration surface needs to be defined. The integration surface can be placed at any location exterior to the body independent of whether or not the propagation is linear outside the surface. If the surface is placed on the body which is assumed to be impenetrable, then the non-linear propagation effects outside of the integration surface are taken into account by the quadrupole volume term, which acts throughout the volume that is exterior to the boundary surface (integration surface). By moving the integration surface away from the body, a sort of mixed formulation is obtained, in which part of the non-linearities is taken into account by the quadrupole volume terms outside of the integration surface, and part by the surface integrals. In this scenario, the “thickness” and “loading” noises do not have any physical significance. At the end, if the surface is far enough from the body, which encloses all physical sources inside, then no contribution is from the volume source, so that the quadrupole term can be legitimately neglected.

In the demonstration case, the integration surface coincides with the airfoil surface, hence the permeable FW-H equation can be simplified to the impermeable FW-H equation due to the

normal velocity of the fluid u_n is the same as the normal velocity v_n of the surface ($u_n = v_n$). In addition, there is no motion applied to the airfoil during the experiment, thus the boundary surface velocity equals to zero. Applying these two relationships to Equation (3.8) - (3.10) gives the impermeable FW-H equation for the stationary surface formulation:

$$4\pi p_L'(\boldsymbol{x},t) = \int_{f=0} [\frac{\dot{p}cos\theta}{cr}]_{ret} dS + \int_{f=0} [\frac{pcos\theta}{r^2}]_{ret} dS, \quad \textbf{(3.11)}$$

where θ is the local angle between normal to the surface and the radial direction $\boldsymbol{r}$ at the emission time, $cos\theta = \hat{n_i} r_i$.

Due to the zero velocity of the airfoil surface, monopole source term (Equation (3.9)) vanishes. This means for flow over airfoil case, only dipole type of noise is generated by the airfoil body, which results from the oscillating pressure on the airfoil surface.

The pressure fluctuation at the microphone position is calculated using Equation (3.11) by an in-house code with FW-H equation for static boundary. Both integrals are estimated at the retarded time τ. The time derivative of the pressure in the first integral on the right hand side of Equation (3.11) is approximated using the Stirling formula [59],

$$\dot{p}_{ret} = \left[\frac{p_{\tau+1} - p_{\tau-1}}{2\tau}\right] + \frac{\Delta\tau}{\tau}[p_{\tau+1} - 2p_\tau + p_{\tau-1}] + \frac{[3\Delta\tau^2 - 1][p_{\tau+1} - 3p_\tau + 3p_{\tau-1} - p_{\tau-2}]}{12\tau} \quad \textbf{(3.12)}$$

This formula gives spline fitting of the pressure-time distribution for every discretized cell. The original code was written to read all the required data at the beginning [58]. However in the current demonstration case, 11,512 cells have been meshed on the airfoil surface, with a total of 300,000 recording time steps. This results in a huge amount of data which translates to a large memory requirement for running the acoustic prediction. To tackle this non-feasible memory requirement problem, the code has been modified into a time advancing algorithm such that at

each time step, only the related flow properties data are read into the code for the calculation. As time loop advances, the current set of data is replaced by the data set which is used for the next time step calculation. Thus the required memory is significantly lowered. At each time step, according to the Stirling formula used in this code, only the data from previous two time steps(τ -1, τ -2), current time step (τ) and the data from the next time step (τ +1) are needed for the current time step calculation. Furthermore, this modification enables the FW-H code to work with the CFD simulation in real time for the acoustic prediction, i.e., the acoustic prediction can run simultaneously with the CFD simulation.

4 Acoustic Source Localization

This chapter presents information about two acoustic source localization techniques. Section 4.1 gives a brief description on NAH. Section 4.2 details the acoustic beamforming method and defines the formulation of beamformer and microphone array used in this study.

4.1 Nearfield Acoustic Holography (NAH)

NAH is an accurate technique for noise source localization [10]. It can provide a high resolution source map of a planar source. It is most commonly used in cases that low frequency noise is dominant and the size of the object to be measured is small. Figure 4.1 shows a typical microphone array setup for NAH application.

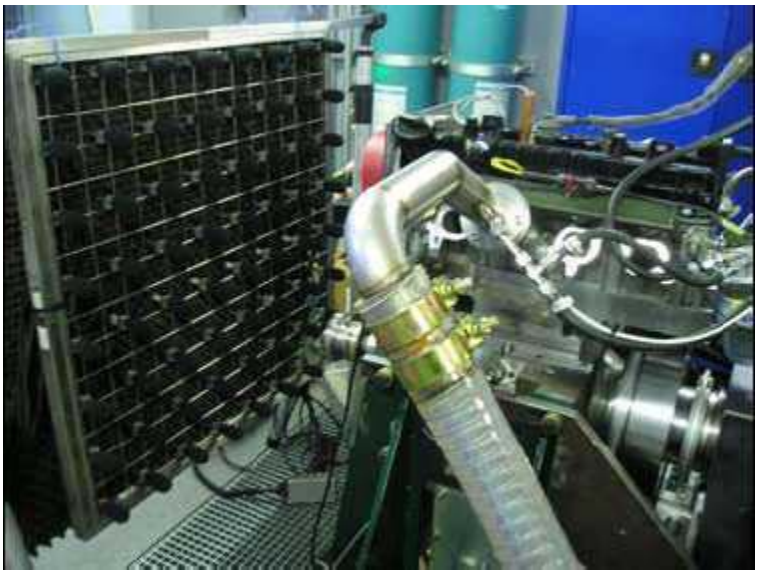

Figure 4.1: NAH measurement setup for engine noise source localization[1]

However, its application in high frequency noise and large source object is limited due to its fundamental requirements. First of all, it requires measurement to be taken from a rectangular grid of points that captures most of the sound radiation from the source into a half-space. This usually requires the microphone array to completely cover a noise source plus a 45° angle in all

[1] Source: ACB Engineering (www.acb-engineering.fr)

directions, as shown in Figure 4.1. Secondly, the spacing between microphones in the array must be less than half of a wavelength at the highest frequency of interest. These two limitations together make it cost prohibitive to perform NAH measurement over a large noise source when the interested frequency is relatively high. This leads to an effort to search for an alternate method that works in areas that NAH is not feasible.

4.2 Acoustic Beamforming

Beamforming rises as an alternative to NAH in mid- to high-frequency range with large-size object. This method is based on estimating the amplitude of plane (or spherical) wave incident towards the array from a chosen set of directions. The microphone array used in beamforming measurement does not need to be larger than the source object, hence distant measurement of large source object is possible, as shown in Figure 4.2. The reason that beamforming can perform high frequency measurement with a low number of microphones is because of its ability to use irregular array geometries. Under the same spatial density of microphones on the array, an irregular array design can keep the spatial aliasing effects to an acceptable level, while a regular grid fails when spacing is larger than half a wavelength [15].

There is also limitation to the beamforming method. Since the microphone array should be much larger than the wavelength of interest in order to get a fine resolution, it becomes cost prohibitive to set up an array to achieve fine resolution in low frequency.

The following sections will briefly introduce the two main components of beamforming: beamformer and microphone array.

Figure 4.2: Beamforming measurement of wind turbine noise[2]

4.2.1 Beamformer

Beamformer (or beamforming algorithm) is the part of the beamforming system that processes measured signal from microphone arrays into the output source map. Planar microphone arrays have a directivity perpendicular to the array surface. In order to form the source strength image, the directivity is steered electronically in a set of directions to form an angular resolution. This process is performed by the beamformer. In general, there are two major categories of beamformers based on whether the processing is before or after a Fourier Transform of the original time domain signal. The beamformer that processes the time domain signals directly is called a time domain beamformer, whereas one that processes the signal in frequency domain is called a frequency domain beamformer [60].

4.2.1.1 Time Domain Beamformer

As the name suggests, time domain beamformer works on time domain signals. Through development, there are many types and variations of beamformer designs that are tailored for different microphone arrays and source types. One of the simplest and most commonly used

[2] Source: ACB Engineering (www.acb-engineering.fr)

beamformer is a Delay-and-Sum beamformer [60]. The basics of such beamformer will be covered in the following section.

4.2.1.1.1 Delay-and-Sum (DAS) Beamformer

The DAS beamformer performs the steering by adding a delay stage to each channel of signal before they are summed together [61]. Figure 4.3 shows a simple schematic of the DAS beamformer. It can be seen that sources from the steered direction are aligned in time and it gives the largest amplitude when added together. At the same time, waves from other directions are not aligned in time. Their summed signal amplitude is lower than the steered direction.

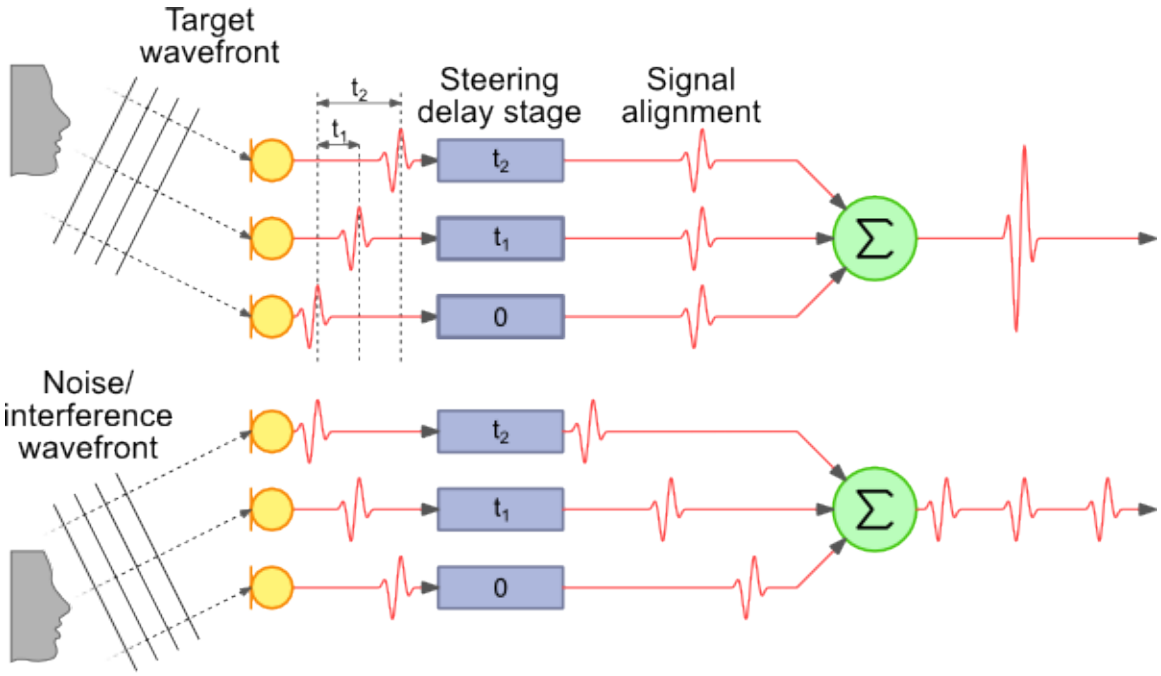

Figure 4.3: DAS beamformer schematic [61]

Mathematically, considering a planar array of M microphones, the beamforming signal using a DAS beamformer is represented by

$$b(\boldsymbol{\kappa}, t) = \sum_{m=1}^{\mathrm{M}} \mathrm{w_m p_m}\big(\mathrm{t} - \Delta_{\mathrm{m}}(\boldsymbol{\kappa})\big), \tag{4.1}$$

where w_m is the weighting applied to each microphone, p_m is pressure signal, Δ_m is the time delay used for steering. The weighting is a shading coefficient applied depending on the microphone array structure. It equals to 1 when the array is uniform. The time delay is calculated by

$$\Delta_m = \frac{\boldsymbol{\kappa} \cdot \mathbf{r_m}}{c}, \tag{4.2}$$

where c is the speed of sound, $\boldsymbol{\kappa}$ is the unit vector representing the direction and $\mathbf{r_m}$ is the vector location of each microphone. Figure 4.4 shows an example of a wave entering the microphone array and the resulting directional sensitivity plot. It is clearly shown that there is a main lobe in the focused direction, but there are also sidelobes in some other directions. It becomes evident that a good beamformer and array combination should have a low maximum sidelobe level (MSL) comparing to the main lobe in order to produce a clear result.

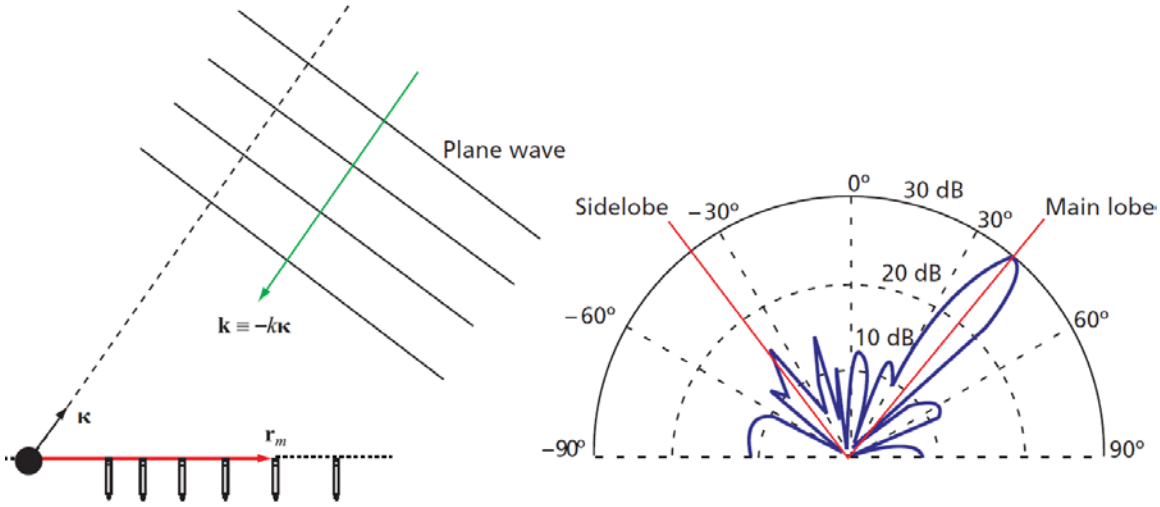

Figure 4.4: Schematic showing the geometry described by the equations and the relevant directional sensitivity diagram [15]

The beamforming result is then processed using Fourier Transform. Then the angular result is projected onto the relative surface where the noise source is located physically. This creates an acoustic image that can be used to identify the location of strong acoustic source.

However, there is no universal combination of the beamformer and microphone array which works for every single situation. Many different algorithms and microphone array designs have been researched and developed trying to minimize the MSL to give an accurate result. Some experimental and numerical tests are shown in Section 5.2.

4.2.1.2 Frequency Domain Beamformer

Contrary to a time domain beamformer, the frequency domain beamformer works on the signal in frequency domain. It employs a phase shift instead of a time delay to steer the microphone array. Since the beamformer is already working on the narrow band signal, its requirement on the sampling rate is lower than that of a time domain beamformer [62]. This is advantageous since the amount of computation and storage required is lower. However, care must be taken when selecting the sampling rate and the phase shift if a result similar to that form a time domain beamformer is to be obtained.

4.2.2 Microphone array

Microphone array is the other important component of a beamforming system. Even though beamformer can theoretically process recorded signals from almost any type of array, the design of array is very crucial to the quality of the final acoustic image. Different beamformers work best with different microphone arrays in different scenarios. Microphone arrays can be categorized into two main groups: regular arrays and irregular arrays.

4.2.2.1 Regular Arrays

One of the easiest ways to construct a planar array is a grid array, as shown in Figure 4.5.

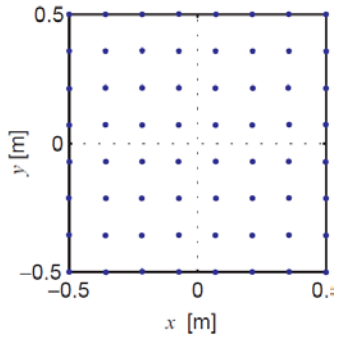

Figure 4.5: Grid array [15]

However, when higher frequency is required, the only way the grid array can adapt is to decrease the grid spacing. This is costly in terms of transducers required. The array may also lose acoustic transparency if the microphones are densely located.

Another efficient way to construct a regular array is the cross array shown in Figure 4.6. It is basically a combination of two linear arrays. This creates the drawback of having a high sidelobe level along the diagonal direction when used with the DAS beamformer. To overcome this problem, some weighting function is introduced when using the diagonal array. An example is shown in Section 5.2.

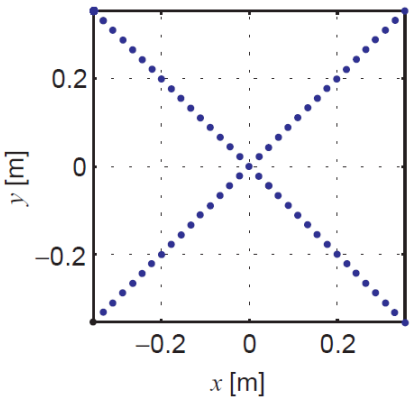

Figure 4.6: Cross array [15]

4.2.2.2 Irregular Arrays

One major advantage of beamforming comparing to NAH is the ability to utilize irregular arrays. The use of irregular arrays can prevent aliasing problem since the differences between the

location vector of any two microphones are not repeated [15]. An irregular array can be totally random like Figure 4.7.

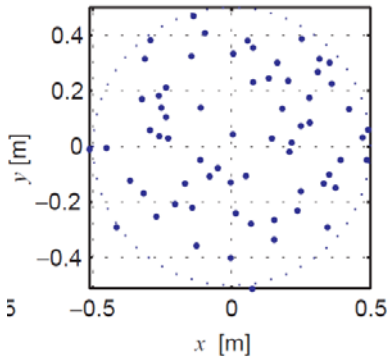

Figure 4.7: Random array [15]

However, even though the irregular arrays usually outperform the regular arrays, it is very difficult to design or optimize one. The relative easier ones to optimize are Archimedean spiral array and wheel array shown in Figure 4.8.

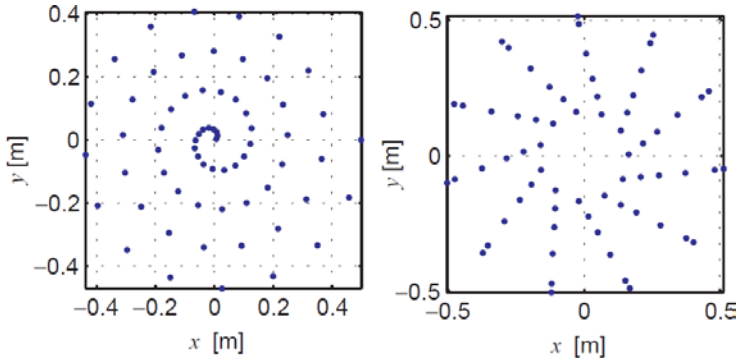

Figure 4.8: Archimedean Spiral array (left) and Wheel array (right) [15]

4.2.2.3 Performance

One way to test the performance of the microphone array and the beamformer combination is to measure a point source at a single frequency. Theoretically, the resulting acoustic image should only show one acoustic hotspot while the rest of the surface should have no emission. In a test case performed by Brüel & Kjær [15], a 5 kHz monopole source is placed 1 m in front of the

microphone arrays. The DAS beamformer is used to generate the acoustic map. The result is documented in Section 5.2.

5 Results and Discussion

A new sound source localization method, Computational Acoustic Beamforming, has been proposed in this thesis. The inspiration of this method is from acoustic beamforming which is an experimental tool to detect the sound source strength, frequency range and location. The central concept of this method is to set up a numerical environment to perform the acoustic beamforming simulations.

As seen in Figure 1.2, there are three modules in the proposed computational method, the CFD module, the CAA module and the acoustic beamforming module. This thesis adopts OpenFOAM to perform the CFD simulation. An in-house code using FW-H formulation is chosen for the CAA calculation. The acoustic beamforming module is coded by the author using Matlab with multiple microphone array designs for validation.

Section 5.1 shows the validation of the in-house code for the CAA calculation. The code adopts the FW-H impermeable formulation (Equation (3.12)) and the test case is the flow over two tandem cylinder case.

Section 5.2 provides the validation of the DAS beamformer with two different microphone array designs over a single frequency sound source, which is usually used during the validation process at the early stage of experiments.

Besides its application on the single point sound source, the proposed Computational Acoustic Beamforming method can also be adopted on a more complicated acoustic field when working with CFD package and CAA code. In Section 5.3, a flow over a NACA 0012 airfoil demonstration case is shown using the Computational Acoustic Beamforming method. The experimental settings of the demonstration case is outlined in Section 5.3.1, while the

corresponding CFD simulation settings and flow validation are listed in Section 5.3.2, followed by the acoustics results and analysis in Section 5.3.3.

5.1 CAA In-House Code Validation

The in-house CAA code uses the impermeable FW-H formulation for static sound source. The test case for the CAA code validation is the NASA tandem cylinder case [63]. The validation of the in-house CAA code has been performed by another member of the research group [58]. Figure 5.1 shows the SPL level at one microphone location and compared with the experimental data. The green squares represent the experimental data and the red line represents the numerical results. The in-house CAA code captures the main frequency and the SPL level very well. Note that the numerical result does not show the broadband noise due to the use of the RANS method in the flow field simulation. The RANS method averaged out the small fluctuations which are the source for the broadband noise, but is good enough for the tonal noise prediction.

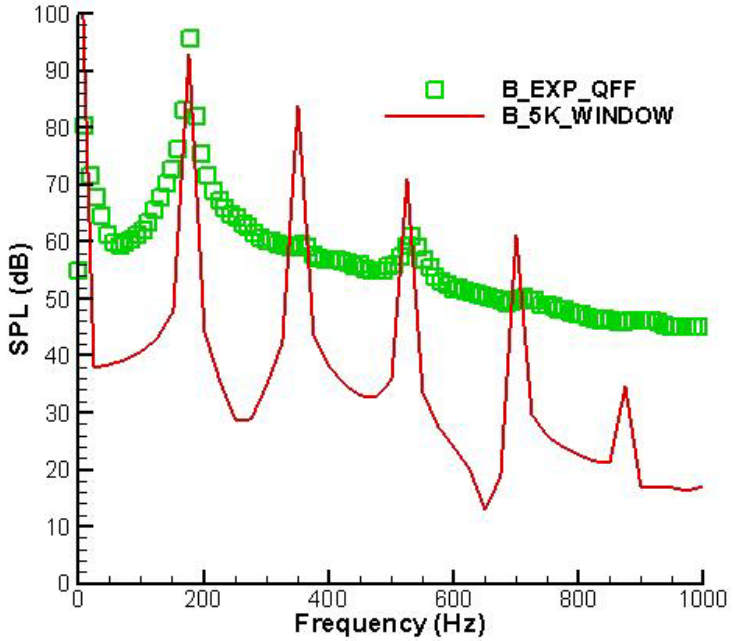

Figure 5.1: Acoustics results for tandem cylinder test case [58]

5.2 Beamforming Validation

Researchers have proposed different beamformers and microphone array arrangements. To validate their proposed combination, one experimental test is usually performed, which is to measure a point source with single frequency. Theoretically, the resulting acoustic image should only contain one acoustic hotspot while the rest of the surface should have no emission.

In this thesis, the time domain DAS beamformer is adopted for the signal processing. To validate the DAS beamformer code, one regular microphone array (65 channel cross array) and one irregular microphone array (66 channel Archimedean spiral array) have been chosen.

The experiment was originally performed by Brüel & Kjær with different microphone arrays to test its beamforming experiment system [15]. It used a 5 kHz monopole source as the sound source and placed it in front of the microphone array center at 1 m distance.

Figure 5.2 shows both the experiment and numerical results. The left panel shows the layout of the microphone arrays used in both the experiment and numerical simulation, the middle and right panels show the acoustic images generated by the experiment and numerical simulation respectively. The results are all normalized to 0 dB at the highest power location with a dynamic range of 15 dB.

Figure 5.2 (a) and (b) use the same microphone array arrangement but Figure 5.2 (b) introduces a Hanning weighting function to the DAS beamformer on each linear array. As mentioned in Section 4.2.2.1, this modification erases the sidelobes level along the diagonal direction which is inherent in this type of microphone array.

Figure 5.2 (a) and (c) compare the results from two different microphone arrays. The acoustic images generated by the numerical simulations match very well with the results from the

experiment. It clearly shows that the Archimedean spiral array generates a much cleaner acoustic image than the cross array when the DAS beamformer is used.

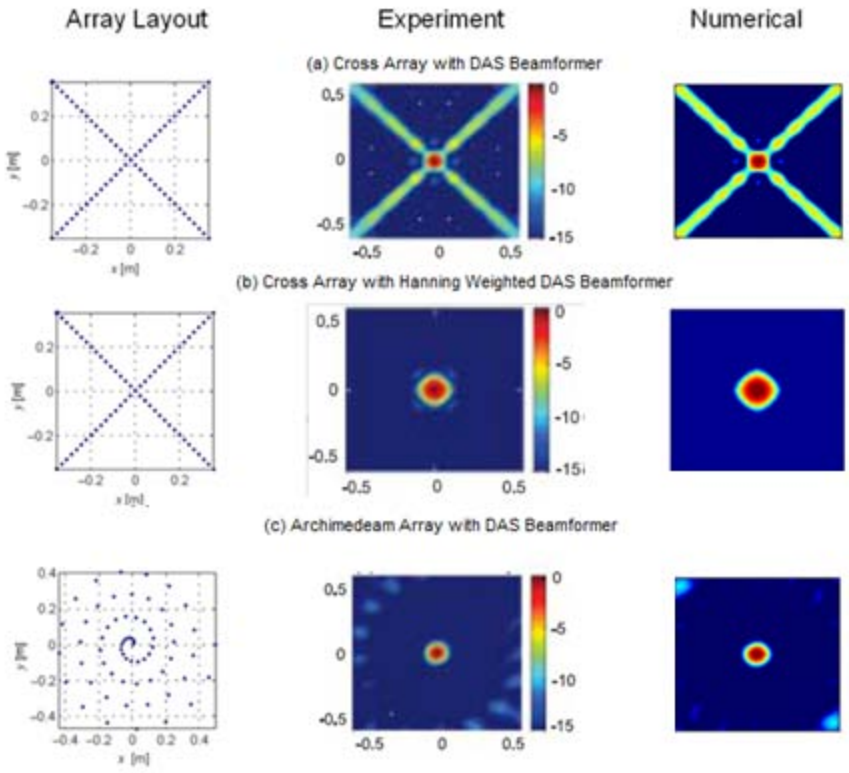

Figure 5.2: Acoustic beamforming result from experiment [15] **and numerical simulation**

When comparing the experimental and numerical results, both the theoretical high power center and sidelobes are showing the same patterns in the two different microphone array arrangements. Furthermore, the acoustic images from numerical simulation show fewer low power dots than that from the experiment. That can be attributed to the fact that there may be background or electrical noise in the experiment which does not exist in the numerical simulation. This gives confidence on the use of acoustic beamforming in a numerical environment.

One can also conclude that from Figure 5.2, even towards the same type of sound source, different microphone array designs or a slight tuning in the beamformer algorithm has significant effects on the output of the final acoustic image.

5.3 NREL Aeroacoustic Testing of Wind Turbine Airfoils

The Advanced Turbulent Flow Research Group and the Vibrations and Acoustics Laboratories of Virginia Polytechnic Institute and State University (Virginia Tech) undertook an extensive aerodynamic and acoustic experimental study of three wind turbine airfoils provided by the National Renewable Energy Laboratory (NREL), including a NACA 0012 airfoil of 0.914m chord [64].

The motivation of the experiment is to develop an experimental database of measurements that contains airfoils' aerodynamic and acoustic data at realistic Reynolds numbers. This database provides a reference for wind turbine designers to balance the aerodynamic and aeroacoustic performance when designing a quiet wind turbine. Such wind turbines can be sited closer to population centers where their power is needed, and can be deployed in greater numbers in such locations.

In this section, the NACA 0012 airfoil case is chosen as the demonstration case to validate the Computational Acoustic Beamforming method. Section 5.3.1 introduces the experimental setup. Section 5.3.2 shows the flow field setup and result using CFD. Section 5.3.3 presents the acoustic wave propagation result from CAA and also the acoustic image from acoustic beamforming algorithm. Comparisons between the numerical results and the experimental measurements are shown and the differences are analyzed.

5.3.1 Experimental Setup

The testing took place at the Virginia Tech Stability Wind Tunnel in its newly developed anechoic configuration. The stability wind tunnel has an anechoic system which consists of an acoustic test section flanked by two anechoic chambers. The upper and lower walls in the acoustic test section are acoustically treated with a series of perforated steel-sheet panels bonded to a layer of Kevlar cloth and backed by foam wedges, which form a smooth and acoustically

non-reflecting flow surface. There are two large rectangular openings on either side of the sidewalls which serve as acoustic windows. Large tensioned panels of Kevlar cloth cover these openings, permitting the sound to pass and containing the bulk of the flow. Sound generated in the test flow exits the test section through these into the anechoic chambers to either side. The rest of the sidewalls are treated the same as explained in the upper and lower walls.

The experiment collected data for each airfoil model at various effective angles of attack (AoA) ranging from zero lift to stall condition (-14° to 14° depending on the airfoil), and various flow speeds ranging from 28 m/s to 66 m/s for nominal chord Reynolds numbers of 1,500,000 to 3,800,000.

Aerodynamic flow measurements consist of static pressure distributions on the airfoil surfaces, wake-profile measurements downstream of the mid-span of the airfoil and single hot-wire measurements in the vicinity of the trailing edge to obtain mean velocity, turbulence quantities and spectra. Figure 5.3 shows the locations of some of the flow measurements.

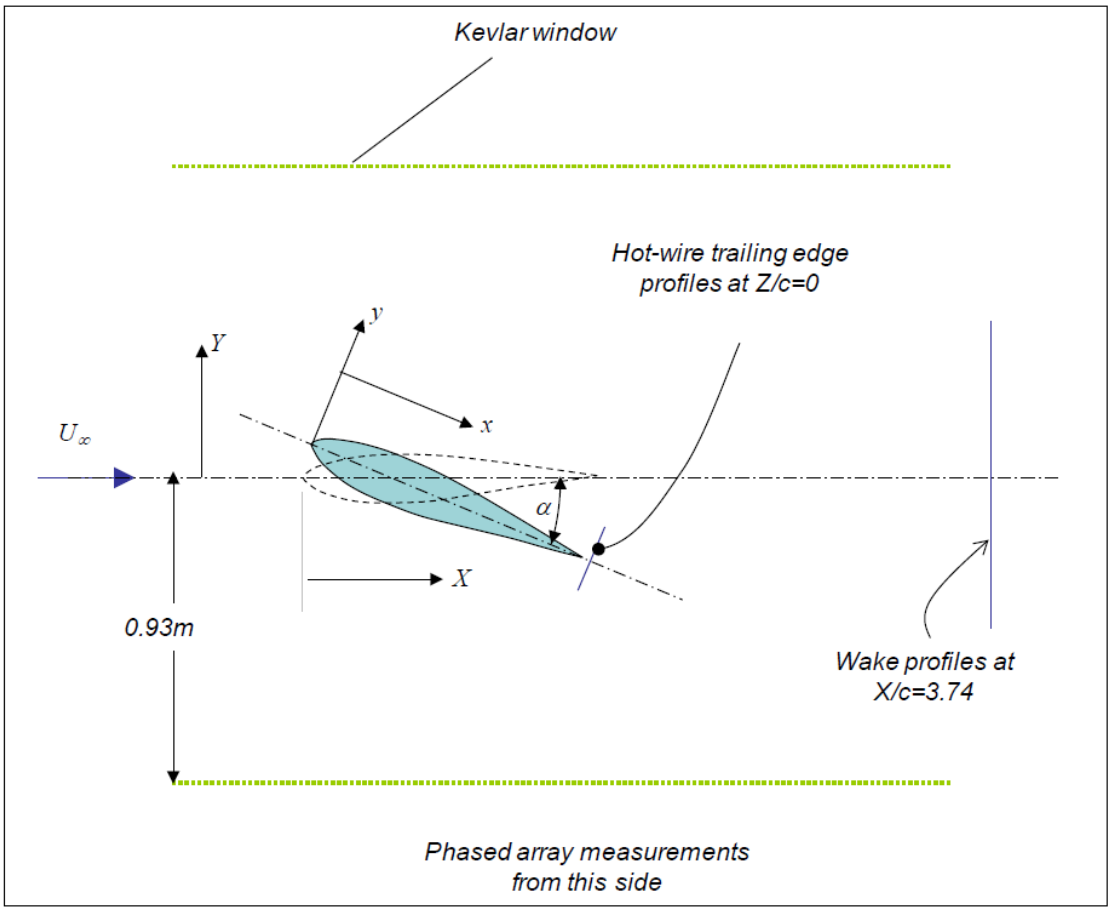

Figure 5.3: Flow properties measurement locations and coordinates[64]

The acoustic data was measured by two 63-microphone phased array systems: the equal-aperture spiral array and the star array. A sampling frequency of 51,200 Hz was used. A conventional frequency domain beamformer with diagonal removal was used to generate the acoustic maps. This algorithm incorporates the convective effect of the flow in the test section and the flow velocity discontinuity between the test section and the anechoic chamber. Figure 5.4 shows the position of the microphone arrays relative to the airfoil model.

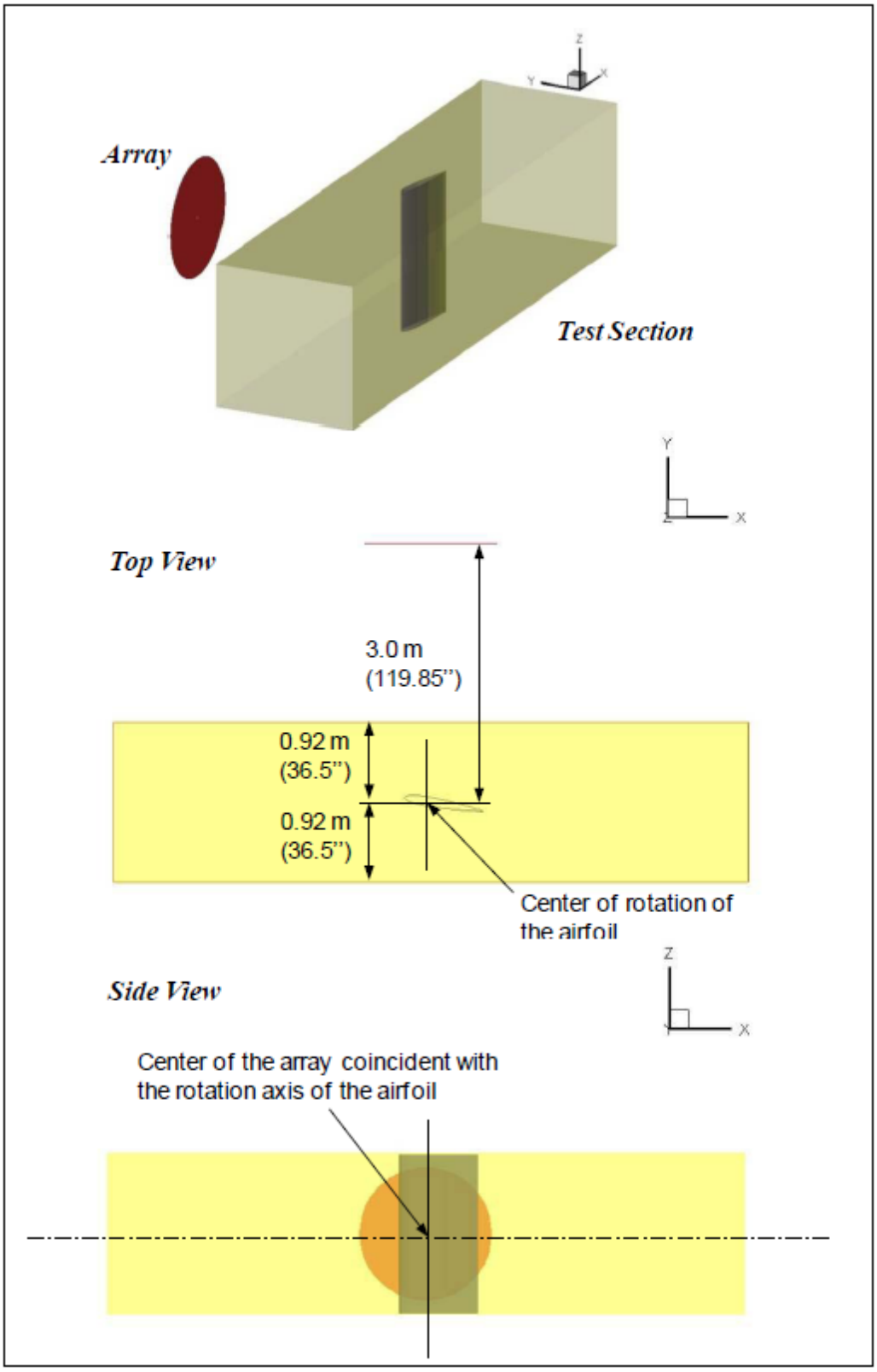

Figure 5.4: Location of microphone array in experiment[64]

To validate the proposed Computational Acoustic Beamforming method, one set of the acoustic and aerodynamic measurement of the airfoil NACA 0012 was chosen as the demonstration case. The detailed information of the measurement is listed in Table 5.1.

Table 5.1: Experimental case parameters

Airfoil Geometry	NACA0012
Geometric AoA	$-10.256°$
Effective AoA	$-8°$
Tripping	No
Inlet Velocity	28.16 m/s
Tunnel Temperature	65.9°F
Chord Length	0.914 m
Re	1.71×10^5
M	0.0822

5.3.2 Flow Field Simulation

5.3.2.1 Geometry and Mesh

The airfoil geometry is generated using CATIA and imported to Ansys ICEM CFD as shown in Figure 5.5. An unstructured 3D mesh is generated using Ansys ICEM CFD as shown in Figure 5.6. The mesh consists of 321,496 elements, with most of them being tetrahedrals and a few wedge elements on the boundary. The computational domain spans 2D upstream, 6D downstream and 0.1D in the span-wise direction where D represents the chord length of the airfoil. A finer grid was created surrounding the airfoil and in the wake region which extends to 3.5D downstream. The finest mesh envelopes approximately 0.2D centered at the trailing edge of the airfoil.

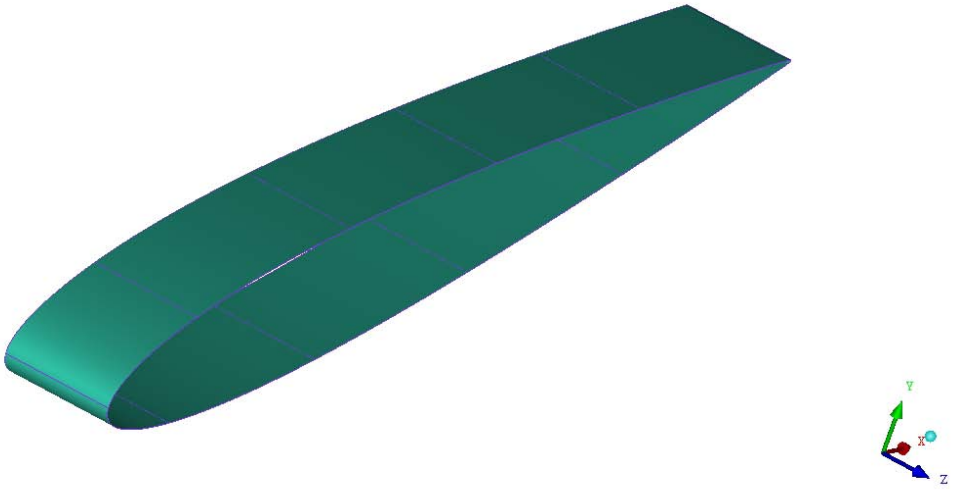

Figure 5.5: Geometry of the airfoil

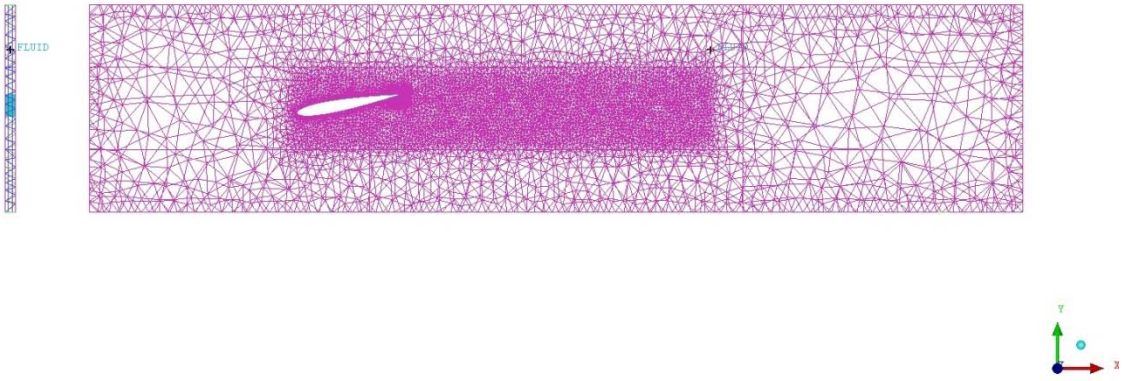

Figure 5.6: Overall mesh of the domain (front and side view)

As seen in Figure 5.7, five layers of wedge elements are generated surrounding the airfoil surface having cell centroids with a target minimum wall-normal distance y+ of 30. Figure 5.8 shows the mesh on the airfoil in the span-wise direction.

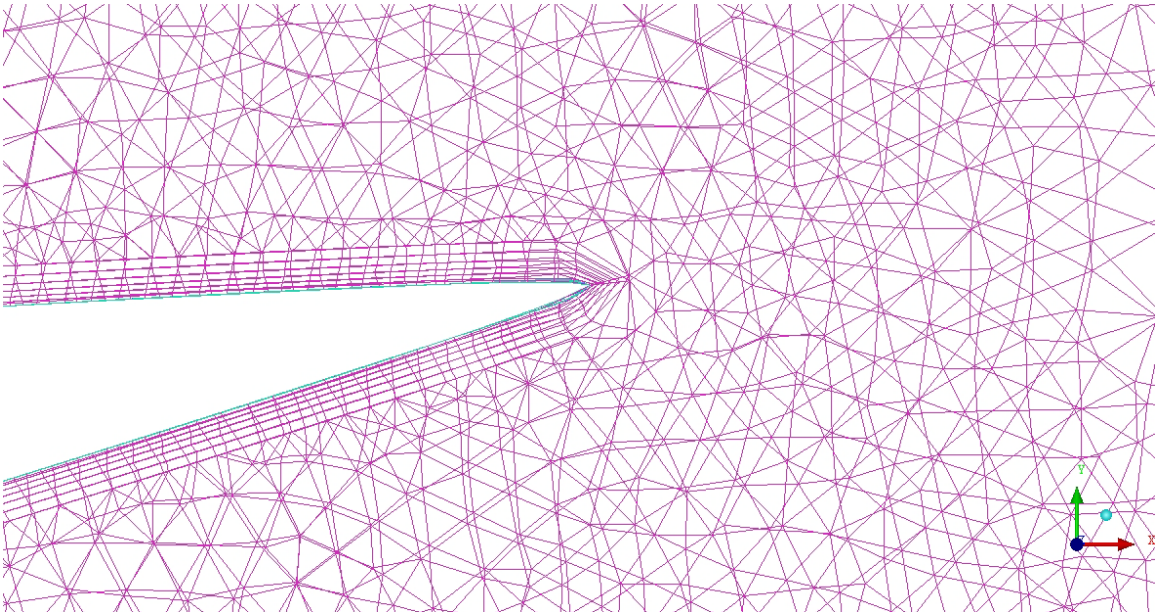

Figure 5.7: Refined mesh at the trailing edge

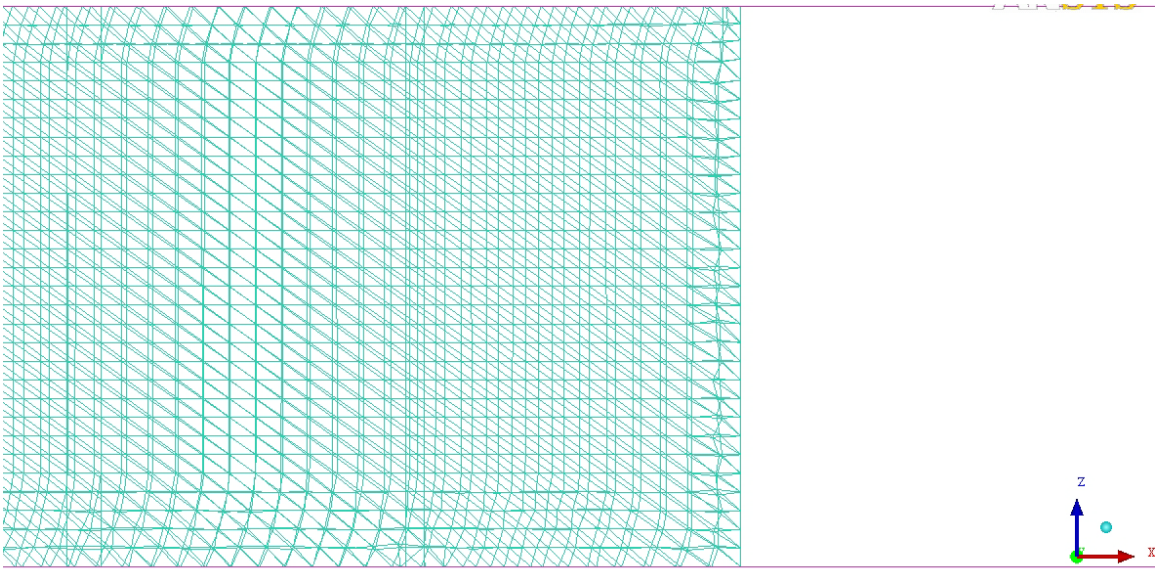

Figure 5.8: Mesh in span-wise direction

5.3.2.2 Boundary Conditions and Solver Settings

As discussed in Chapter 2, the flow simulation was conducted by OpenFOAM, where the three dimensional unsteady N-S equations are solved using a cell centered FVM. Appendix II shows different discretization schemes used in the momentum, pressure and turbulent transport

equations. Appendix III lists the control and solver settings for the simulation. Table 5.2 lists the boundary conditions applied in the simulation.

Table 5.2: Boundary conditions in CFD simulation

	Inlet	Outlet	Wall	Airfoil	Symmetry Planes
Pressure	zeroGradient	fixedValue Value = 0	zeroGradient	zeroGradient	symmetryPlane
Velocity	fixedValue Value = (28.16 0 0)	inletOutlet inletValue = (28.16 0 0)	fixedValue Value = (0 0 0)	fixedValue Value = (0 0 0)	symmetryPlane
nuTilda	inletOutlet inletValue = 0	inletOutlet inletValue = 0	nuSgsUSpaldingWallFunction Value = 1e-10	nuSgsUSpaldingWallFunction Value = 1e-10	symmetryPlane
nuSGS	zeroGradient	zeroGradient	nuSgsUSpaldingWallFunction Value = 1e-10	nuSgsUSpaldingWallFunction Value = 1e-10	symmetryPlane

The simulation runs 11.5 s flow time in total. The flow becomes fully developed at around 10 s flow time. The pressure fluctuation data starts recording after that for a duration of 1.5 s. It is used for both the flow and acoustic field analysis.

5.3.2.3 Flow Field Result

Figure 5.9 shows the comparison of the mean pressure coefficient distribution on the airfoil surface with clean surface and no trip configuration. The circles represent the numerical results and the line represents the results from panel method (XFOIL). The panel method used here is a standard linear-vortex panel method. About 200 panels were used to represent the shape of the airfoil. The formula used to calculate the pressure coefficient is

$$c_p = \frac{p - p_\infty}{p_{o\infty} - p_\infty} = \frac{p - p_\infty}{\frac{1}{2}\rho u_\infty^2}, \quad \textbf{(5.1)}$$

where p is the local pressure, $p_{o\infty}$ and p_∞ are the reference free-stream stagnation and static pressures, and u_∞ is the free-stream velocity.

Figure 5.9 top plot shows the numerical results compared with the panel method. The simulation results match well with the panel method and a good agreement was obtained on the pressure side of the airfoil. However, the numerical results underpredict the mean pressure coefficient along the

suction side at the first part of the airfoil as well as the suction peak. This underprediction leads to a lower lift force which is calculated as a pressure integral along the airfoil surfaces.

Figure 5.9 bottom plot shows the experimental measurement of the mean pressure coefficient compared with the panel method. The experiment conducted at 8.1 degree of effective AoA with $Re = 1.48 \times 10^6$. Despite the minor difference on the Reynolds number and effective AoA between the numerical prediction and experiment, the simulation agree well with the measurement. Note that the panel method has a drop in c_p close to the trailing edge while both the experiment and simulation stays flat. This is an inherent drawback for the inviscid 2-D panel method on predicting c_p at the trailing edge of the airfoil while vortices are presenting. The good agreement between the measurement and CFD results at the trailing edge area shows that the simulation is sufficient to provide realistic result for the flow field at the trailing edge.

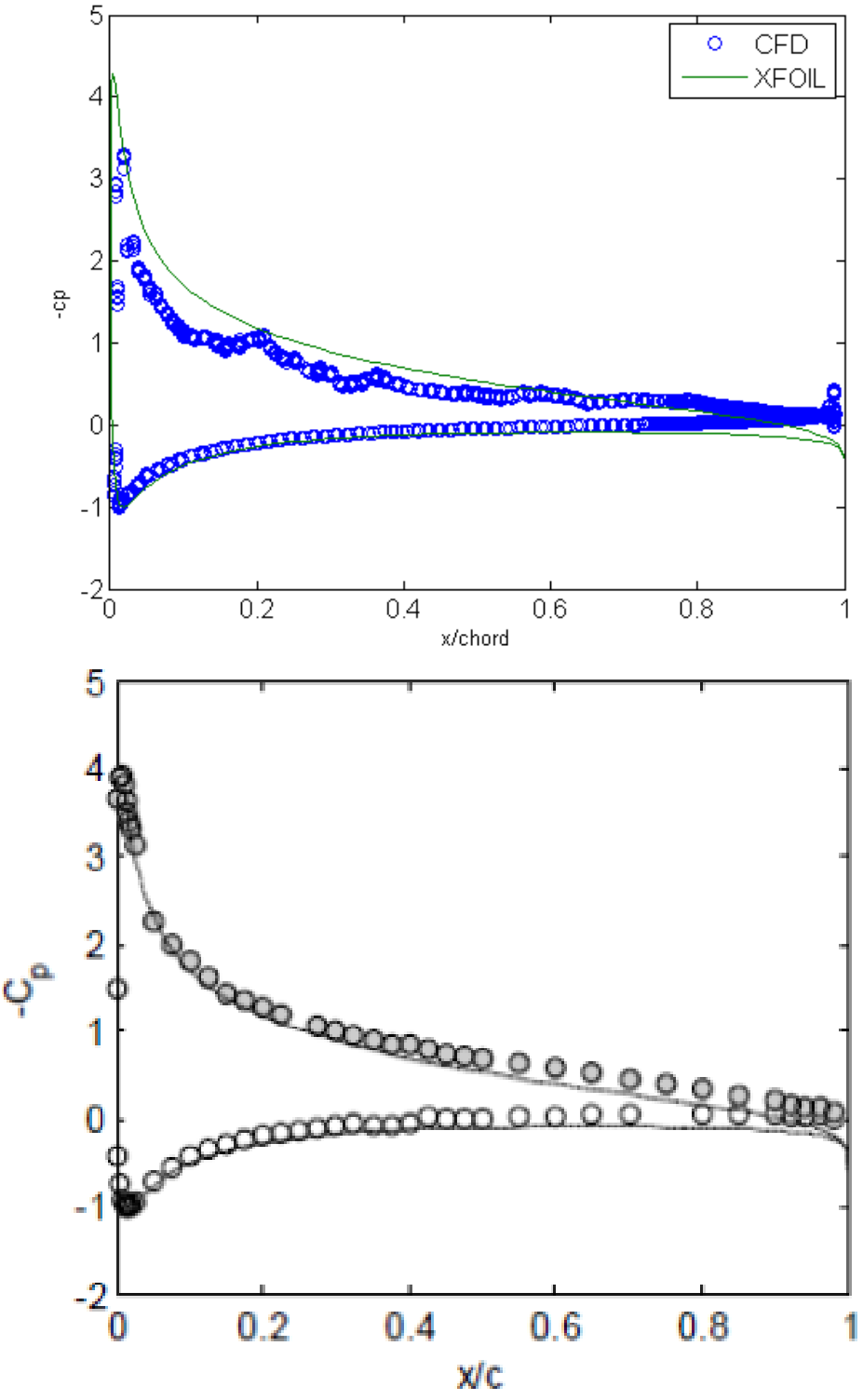

Figure 5.9: Pressure coefficient (c_p) comparison of CFD vs XFOIL (Top, AoA = 8° and Re = 1.71x10^5) and experiment vs XFOIL (Bottom, AoA = 8.1° and Re = 1.48x10^6)[64]

Figure 5.10 shows the span-wise component of the velocity contour in the wake region. Due to the computational resources limitation, the span can only extend 1/10 of the chord length, which is relatively short for flow to develop in this direction. However, even with this short span width, a significant variation of the velocity can still be observed which indicates a high three dimensional effect for this type of flow.

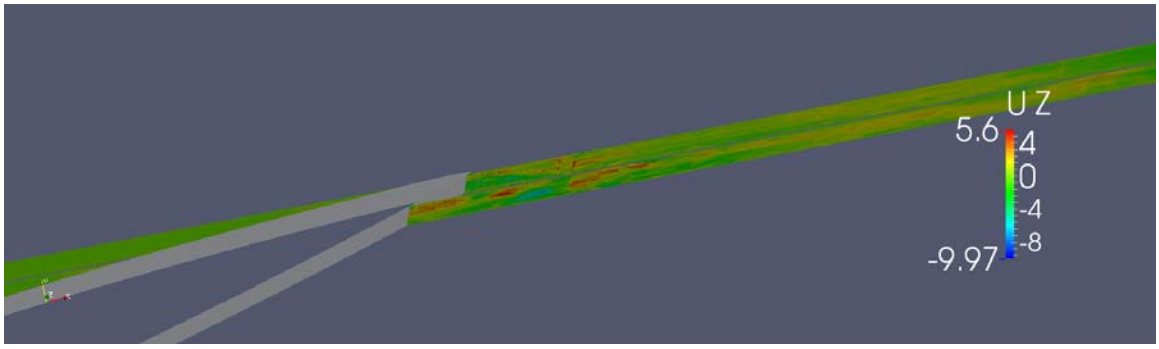

Figure 5.10: Span-wise velocity component close to trailing edge

Figure 5.11 and Figure 5.12 illustrate the velocity vector and streamline around the airfoil and on the suction side of the airfoil. The flow is attached on the pressure side whereas a small disturbance was observed close to the leading edge on the suction side. This fluctuation develops along the main flow direction and starts to affect the flow in the span-wise direction at 1/2 chord length, which causes the flow separation and vortices towards the trailing edge area.

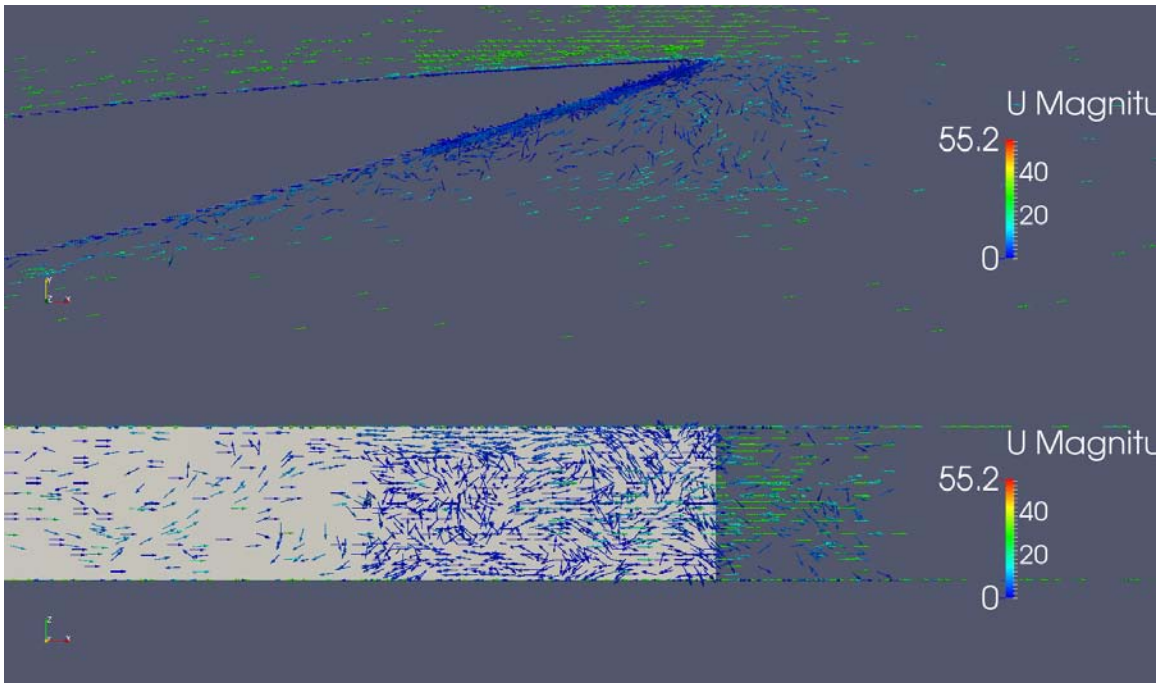

Figure 5.11: Vector plot of velocity close to trailing edge

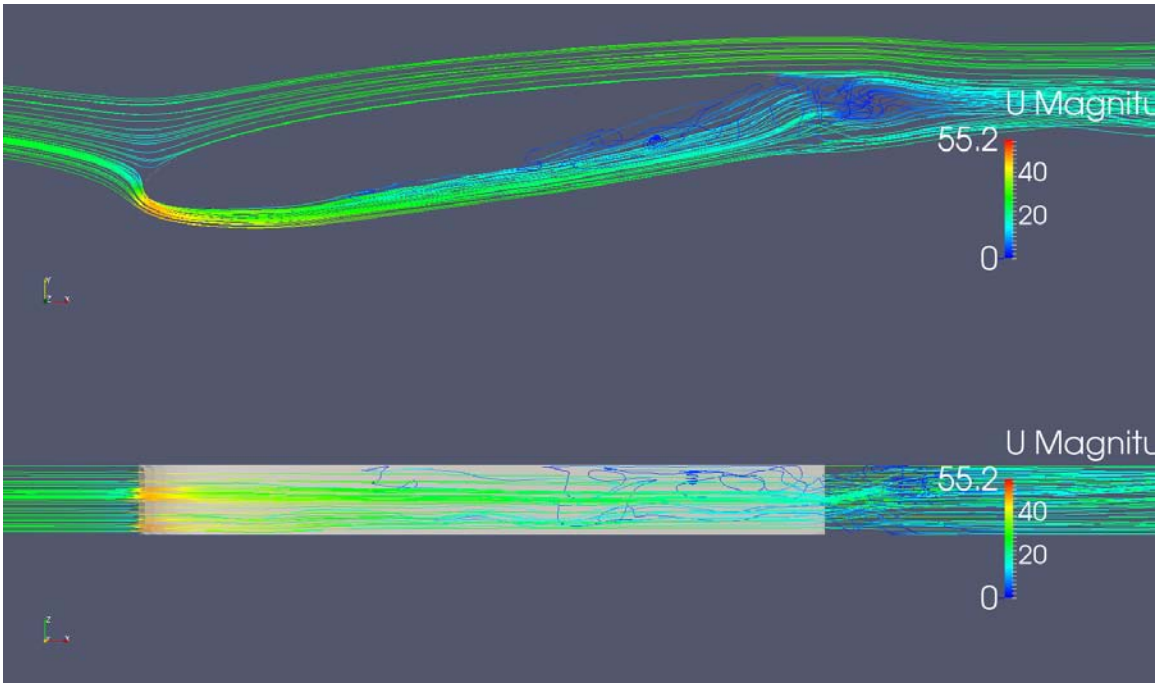

Figure 5.12: Streamline plot of velocity over the airfoil

The velocity magnitude on the mid plane is shown in Figure 5.13. The flow starts to separate at around 1/5 of the chord length. The simulation successfully predicts the vortex shedding from the

trailing edge which propagates about one and a half chord length downstream. However, the shape of the vortices is not prominent, which might present problem for the acoustic prediction as shown in Section 5.3.3. The SPL at the peak frequency is not obvious in the microphone signal plot in Figure 5.16. For a relatively coarse mesh used in this thesis, this exhibits difficulties for CFD to predict the flow properties correctly and might be the reason for the under prediction of the mean pressure coefficient shown in Figure 5.9.

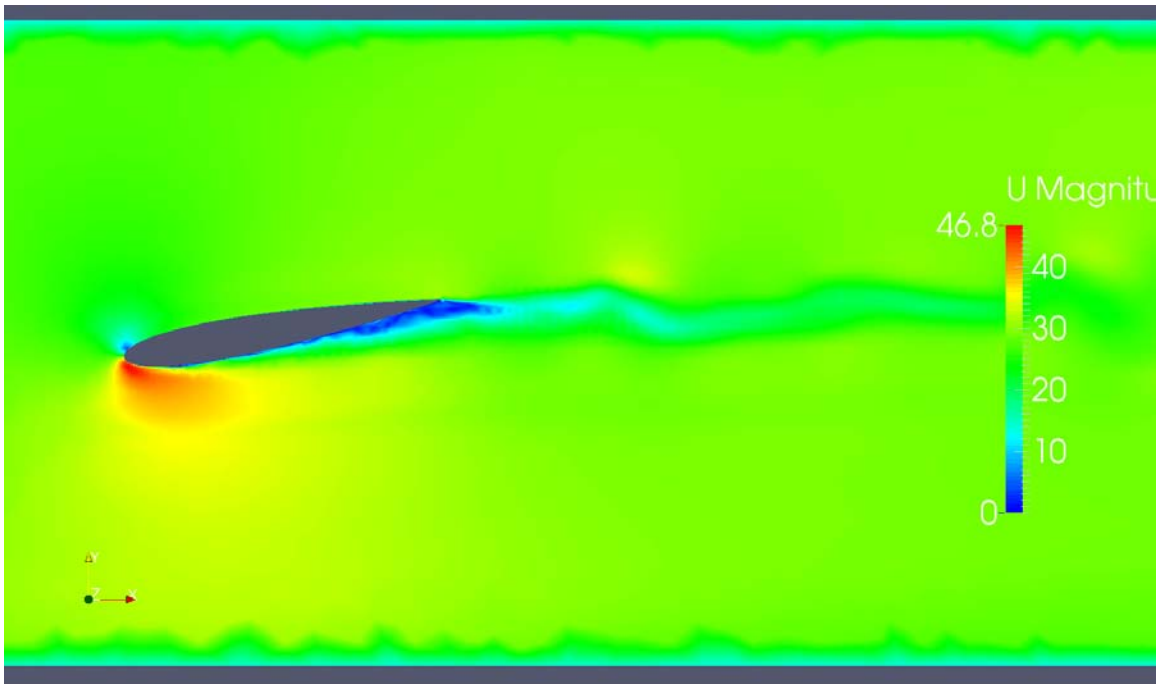

Figure 5.13: Velocity contour of the airfoil and wake region

Figure 5.14 shows the iso-surface of Q around the airfoil surface. The Q criterion for incompressible flow is defined as

$$Q = -\frac{1}{2}\left(S_{ij}S_{ij} - \Omega_{ij}\Omega_{ij}\right), \quad \textbf{(5.2)}$$

where

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial u_j} + \frac{\partial u_i}{\partial u_j}\right), \quad \textbf{(5.3)}$$

$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial u_j} - \frac{\partial u_i}{\partial u_j}\right). \quad (5.4)$$

As seen from Equation (5.2), Q is a local measure of the excess rotation rate relative to the strain rate. When the Q value is positive, it represents locations in the flow where the rotation dominates the strain and shear, hence it is a good representation of the turbulent structures.

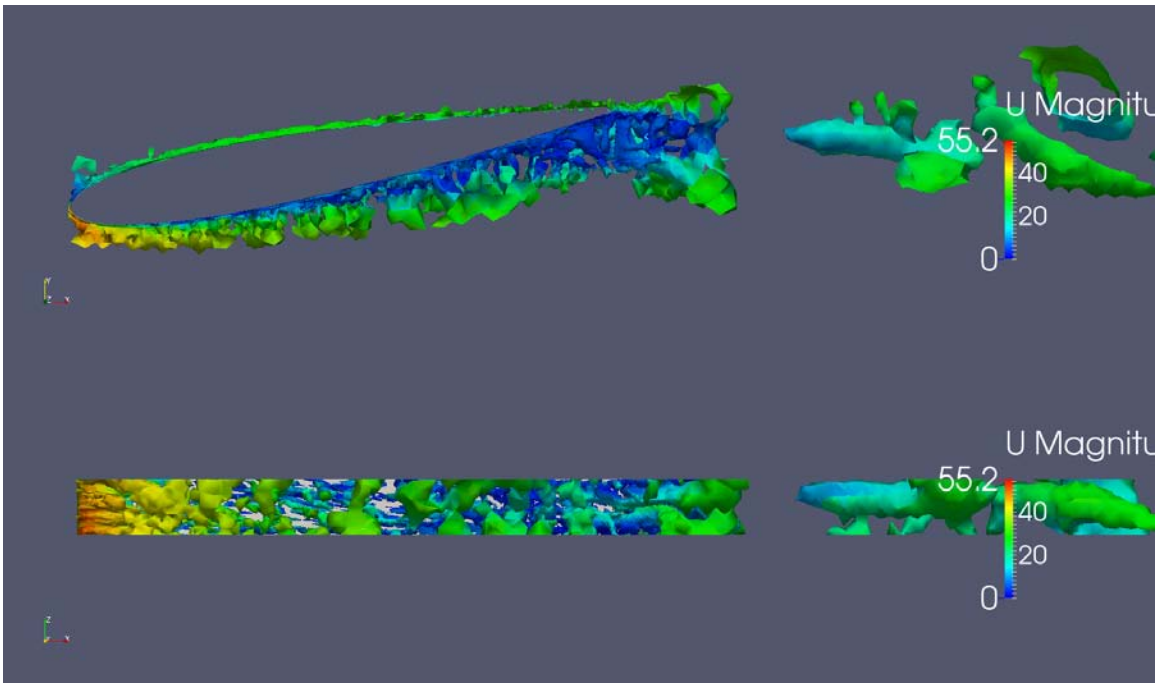

Figure 5.14 Isosurface of Q = 1000 with contour of velocity magnitude

5.3.3 Acoustics Results

An in-house Matlab code is used to calculate the acoustic wave propagation to the microphone array. The code adopts the FW-H impermeable formulation for static sound source. The code has been validated by the flow over two cylinders in tandem test case in Section 5.1. It gives good predictions compared to the experimental results.

The input of this in-house code is the pressure fluctuation on the airfoil surface from flow field simulation. As mentioned in Section 5.3.2.2, OpenFOAM records 1.5 s of pressure fluctuation data, with time step = 0.000005 s. This gives a total of 300,000 pressure fluctuation data points at

each of the 11,512 points on the airfoil surface. With such tremendous amount of data, a huge memory is required to run the original in-house code which requires the complete data set to be read into the memory at the same time. This makes the code impossible to run on a personal computer. To tackle this problem, the code has been modified in this thesis that at each time step, only the related time steps' pressure data is read and as the time loop advances, the current pressure data set will be replaced by the next set of the related pressure data. In this way, the memory required throughout the simulation is kept very low which makes the code running on a personal computer possible.

Figure 5.15 shows the microphone array layout for this simulation. The Archimedean Spiral array was chosen in this thesis to work with the DAS beamformer, as this combination gives the best result as shown in Figure 5.2. Figure 5.16 shows the calculated acoustic signal at the array center microphone on the left and the Fourier transform of the signal on the right. The SPL in the vertical axis is calculated by

$$SPL = 20\log_{10}\left(\frac{p'}{p_{ref}}\right), \tag{5.5}$$

where $p_{ref} = 0.00002Pa$.

The acoustic signals at other microphone locations are not presented because a similar pressure fluctuation pattern is observed with a slight shift in time only.

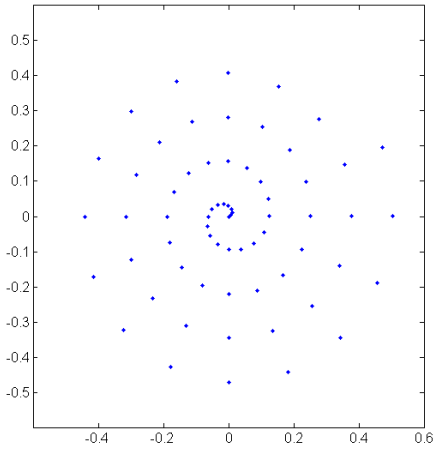

Figure 5.15: Archimedean spiral array

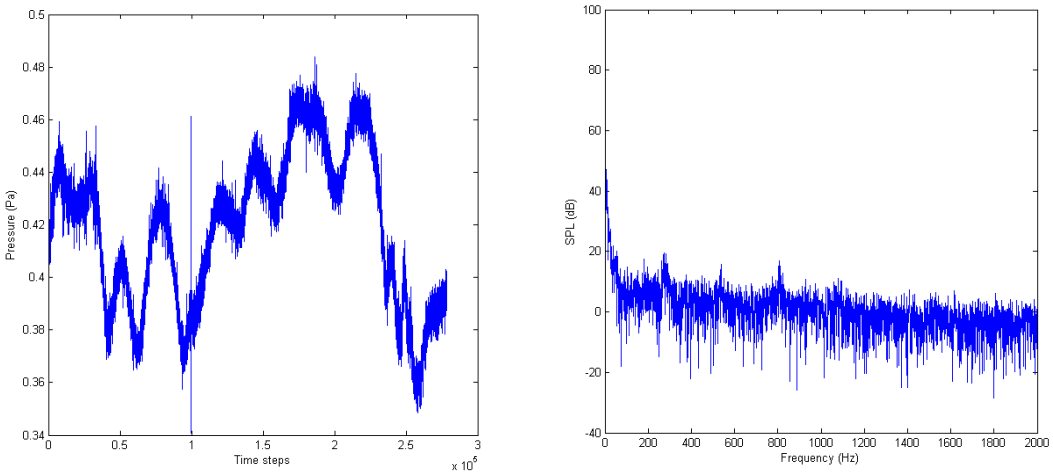

Figure 5.16: Pressure-Time history (left) and SPL-Frequency plot (right) of the array center microphone

As seen in Figure 5.16, the acoustic signal at the array center microphone shows a very irregular fluctuation. No periodic pattern is noticed during the recorded time period. Also, the FFT plot shows a relatively low magnitude in terms of SPL compared with the experimental data shown in Figure 5.17. Since the FW-H formulation adopted in the CAA code does not account for atmospheric attenuation during sound propagation, this even SPL distribution is probably a result of the CFD simulation, where the relative coarse mesh might have some turbulent energy dissipated.

However, the SPL shown in Figure 5.16 has two spikes at 800Hz and 1100Hz. These two frequencies correspond to the dominate trailing edge noise frequencies measured in experiment as shown in Figure 5.17. This gives confidence that even though the dissipation of turbulence energy is large due to the relatively coarse mesh, the characteristic noise frequencies can still be captured in the simulation.

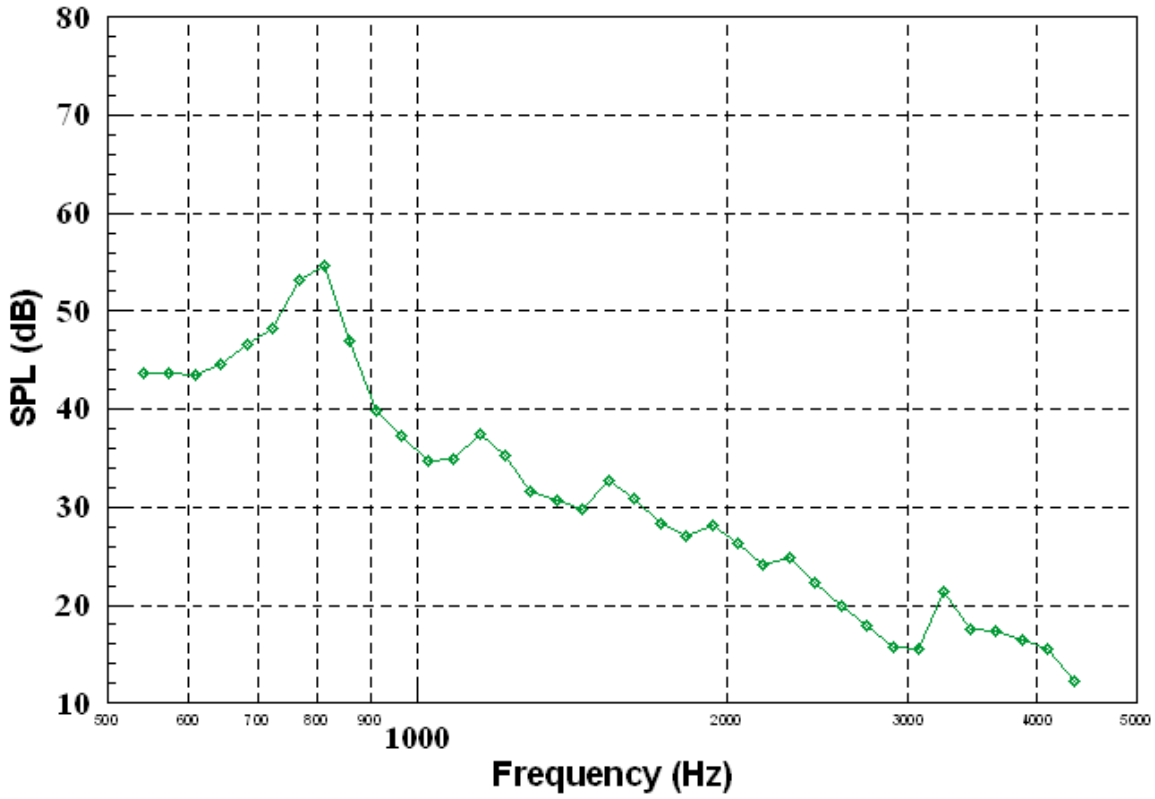

Figure 5.17: Trailing edge volume integral source level measured in experiment [64]

While the peak frequencies are well captured, the SPL at the peak frequencies from simulation is not as prominent as in the experiment. This may be attributed to a couple reasons. First of all, the duration of data recorded in the simulation is very short comparing to experimental measurement. The peak SPL will become more prominent than in the current result if longer pressure fluctuation data is used. Another possible reason is that due to the coarse mesh in the CFD domain, the different scales of the fluctuation gets mixed out earlier on the airfoil surface and in

the downstream which lead to large energy dissipation in these areas, so that the sound pressure level at the peak frequencies are lower than that measured in the experiment.

The microphone signals from the Archimedean spiral array are then input into the acoustic beamforming code. The core of the code is attached in Appendix IV.

The beamforming result in Figure 5.18 shows the acoustic map predicted at 800 Hz 1/3 octave band. The airfoil is located between x = 0 m (leading edge) and x = 0.914 m (trailing edge). Note that the scale in y direction is exaggerated in order to provide a better view of the result, since the simulated airfoil has only 0.1D in span-wise direction. As in Figure 5.2, the legend shows the relative SPL of the source strength normalized to 0dB as maximum.

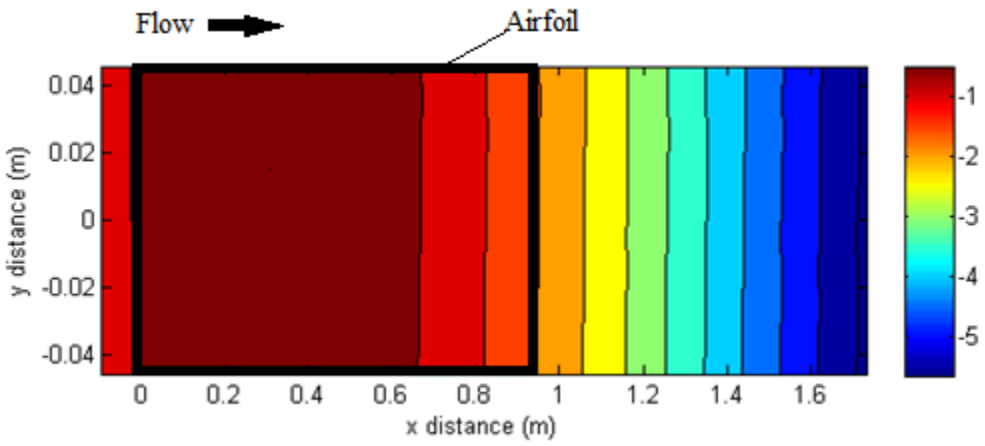

Figure 5.18: Beamforming result at 800Hz 1/3 octave band

There is very minimal variation observed in the span-wise direction which is expected. This is because only the center slice of the airfoil is used and there is no wall boundary on either end.

In the chord direction, the highest noise source level is overlapping the leading edge and the front two-third of the airfoil. Even though one would expect the noise source to be stronger at the trailing edge, experimental data is not available in this case to verify the speculation.

6 Conclusions and Future Work

A new method, Computational Acoustic Beamforming, is proposed and tested in this thesis as a computational sound source localization tool. This method combines the advantages of relatively low cost CFD simulation with the widely accepted method of acoustic source localization by using beamforming algorithm. The new methodology establishes a numerical environment to implement the acoustic beamforming experiment.

Three modules constituted this new method has been shortly reviewed and discussed in CFD, CAA and acoustic beamforming chapters. In this thesis, the CFD module adopts an open source commercial package, OpenFOAM. The CAA module uses an in-house code with the impermeable FW-H formulation for static sound source. The acoustic beamforming algorithm is written by the author of this thesis.

The CAA in-house code has been validated with the experimental data and other numerical results from the flow over two cylinders in tandem case. The acoustic beamforming code is validated by a typical single frequency sound source experiment conducted by Brüel & Kjær in Denmark.

The NACA 0012 airfoil experiment conducted in Virginia Tech wind tunnel has been chosen for the validation of Computational Acoustic Beamforming. The aerodynamic results show good agreement with the results from experiment and panel method. The peak frequencies are well captured but the SPL at these frequencies is underpredicted. In all, the test case results gave confidence in using the newly proposed Computational Acoustic Beamforming method in sound source localization application.

To increase the accuracy of the method and produce more precise results, there is some future work that needs to be done:

1. Generating a finer mesh to be used in the CFD simulation to capture the different turbulence scales and flow interaction;
2. Including focalization function in beamforming algorithm to account for spherical wave propagation;
3. Experimenting with different combinations of beamformers and microphone array designs.

7 References

[1] Global Wind Energy Council, “Global Wind Report,” 2010.

[2] E. Pedersen and K. Persson Waye, “Wind turbine noise, annoyance and self-reported health and well-being in different living environments.,” *Occup. Environ. Med.*, vol. 64, no. 7, pp. 480–6, Jul. 2007.

[3] “Vestas,” 2014. [Online]. Available: http://www.vestas.com/.

[4] “Ontario’ s Long-Term Energy Plan,” *Queen’s Printer for Ontario*, 2010. [Online]. Available: http://www.mei.gov.on.ca/en/pdf/MEI_LTEP_en.pdf.

[5] “Wind Ontario,” 2013. [Online]. Available: http://www.windontario.ca/.

[6] S. Wagner, R. Bareiss, and G. Guidati, *Wind Turbine Noise*. Ney York: Springer, 1996.

[7] T. Imamura and Y. Takahashi, “Unsteady Flow Simulation around Cylinder under Airfoil using Cartesian-based Flow Solver,” in *21st AIAA Computaional Fluid Dynamics Conference*.

[8] R. D. Sandberg and N. D. Sandham, “Direct Numerical Simulation of Turbulent Flow Past a Trailing Edge and the Associated Noise Generation,” *J. Fluid Mech.*, vol. 596, pp. 353–385, 2008.

[9] L. E. Jones and R. D. Sandberg, “Acoustic and Hydrodynamic Analysis of the Flow around an Aerofoil with Trailing-Edge Serrations,” *J. Fluid Mech.*, vol. 706, pp. 295–322, 2012.

[10] J. D. Maynard, E. G. Williams, and Y. Lee, “Nearfield acoustic holography : I . Theory of generalized holographyand the development of NAH,” *J. Acoust. Soc. Am.*, vol. 78, no. 4, pp. 1395–1413, 1985.

[11] W. A. Veronesi and J. D. Maynard, “Nearfield acoustic holography (NAH) II. Holographic reconstruction algorithms and computer implementation,” *J. Acoust. Soc. Am.*, vol. 81, no. 5, pp. 1307–1322, 1987.

[12] R. J. Lustberg, “Acoustic Beamforming Using Microphone Arrays,” MIT, 1993.

[13] U. Michel, “History of acoustic beamforming,” in *Proceedings of the Berlin Beamforming Conference*, 2006, pp. 1–17.

[14] X. Huang, “Real-time algorithm for acoustic imaging with a microphone array.,” *J. Acoust. Soc. Am.*, vol. 125, no. 5, pp. EL190–5, May 2009.

[15] J. J. Christensen and J. Hald, “Technical review - Beamforming, No.1,” 2004.

[16] J. Boussinesq, *Essai sur la théorie des eaux courantes*. Imprimerie Nationale, 1877.

[17] P. A. Durbin and B. A. Pettersson Reif, *Statistical Theory and Modeling for Turbuleng Flows*. Wiley, 2001, p. 302.

[18] J. Smagorinsky, “General Circulation Experiments With The Primitive Equations,” *Mon. Weather Rev.*, vol. 91, no. 3, pp. 99–164, 1963.

[19] A. N. Kolmogorov, “The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds Numbers,” *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 434, no. 1890, pp. 9–13, Jul. 1991.

[20] P. SPALART and S. ALLMARAS, “A one-equation turbulence model for aerodynamic flows,” in *30th Aerospace Sciences Meeting and Exhibit*, 1992.

[21] P. R. Spalart, “Detached-Eddy Simulation,” *Annu. Rev. Fluid Mech.*, vol. 41, no. 1, pp. 181–202, Jan. 2009.

[22] P. R. Spalart, S. Deck, M. L. Shur, K. D. Squires, M. K. Strelets, and A. Travin, “A New Version of Detached-eddy Simulation, Resistant to Ambiguous Grid Densities,” *Theor. Comput. Fluid Dyn.*, vol. 20, no. 3, pp. 181–195, May 2006.

[23] W. Haase, M. Braza, and A. Revell, *DESider – A European Effort on Hybrid RANS-LES Modelling*, vol. 103. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[24] M. L. Shur, P. R. Spalart, M. K. Strelets, and A. K. Travin, “A hybrid RANS-LES approach with delayed-DES and wall-modelled LES capabilities,” *Int. J. Heat Fluid Flow*, vol. 29, no. 6, pp. 1638–1649, Dec. 2008.

[25] U. Schumann, “Subgrid Scale Model for Finite Difference Simulations of Turbulent Flows in Plane Channels and Annuli,” *J. Comput. Phys.*, vol. 18, no. 4, pp. 376–404, 1975.

[26] O. Verhoeven, “Trailing Edge Noise Simulations using IDDES in OpenFOAM,” Delft University of Technology, 2011.

[27] B. Van Leer, “Towards the Ultimate Conservative Difference Scheme II. Monotonicity and Conservation Combined in a Second Order Scheme,” *J. Comput. Phys.*, vol. 14, no. 4, pp. 361–370, 1974.

[28] R. I. Issa, “Solution of Implicitly Discretized Fluid Flow Equations by Operator Splitting,” *J. Comput. Phys.*, vol. 62, pp. 40–65, 1986.

[29] H. Shen and C. K. W. Tam, “Numerical Simulatin of the Generation of Axisymmetric Mode Jet Screech Tones,” *AIAA J.*, vol. 36, pp. 1801–1807, 1998.

[30] E. Envia, A. Wilson, and D. Huff, “Fan Noise: A Challenge to CAA,” *Int. J. Comut. Fluid Dyn.*, vol. 18, no. 6, pp. 471–480, 2004.

[31] B. Singer and Y. Guo, “Development of Computational Aeroacoustics Tools for Airframe Noise Calculations,” *Int. J. Comut. Fluid Dyn.*, vol. 18, no. 6, pp. 455–469, 2004.

[32] K. Brentner, “An Efficient and Robust Method for Predicting Helicopter High-Speed Impulsive Noise,” *J. Sound Vib.*, vol. 203, no. 1, pp. 87–100, 1997.

[33] A. S. Lyrintzis, E. K. Koutsavdis, C. Berezin, J. Visintainer, and M. Pollack, “Kirchhoff Acoustic Methodology Validation and Implementation in the TiltRotor Aeroacoustic Codes (TRAC),” in *American Helicopter Society 2nd International Aeromechanics Specialists’ Conference*, 1995.

[34] R. J. Astley, R. Sugimoto, I. M. Achunche, M. F. Kewin, P. Mustafi, and E. P. Deane, “A Review of CAA for Fan Duct Propagation and Radiation, with Application to Liner Optimisation,” *Procedia Eng.*, vol. 6, pp. 143–152, 2010.

[35] X. Chen, X. Huang, and X. Zhang, “Sound Radiation from a Bypass Duct with Bifurcations,” *AIAA J.*, vol. 47, no. 2, pp. 429–436, 2009.

[36] A. Piacsek, L. Locey, and V. Sparrow, “Time-domain Modeling of Atmospheric Turbulence Effects on Sonic Boom Propagation,” in *14th AIAA/CEAS Aeroacoustics Conference (29th AIAA Aeroacoustics Conference).*

[37] Christopher K. W. Tam and J. C. Webb, “Dispersion-Relation-Preserving Finite Difference Schemes for Computational Acoustics,” *J. Comput. Phys.*, 1993.

[38] Christopher K. W. Tam, “Computational aeroacoustics - Issues and methods,” *AIAA J.*, vol. 33, no. 10, pp. 1788–1796, Oct. 1995.

[39] J. W. Kim and D. J. Lee, “Optimized compact finite difference schemes with maximum resolution,” *AIAA J.*, vol. 34, no. 5, pp. 887–893, May 1996.

[40] S. Lele, “Compact finite difference schemes with spectral-like resolution,” *J. Comput. Phys.*, vol. 103, no. 1, pp. 16–42, Nov. 1992.

[41] M. Wang, J. B. Freund, and S. K. Lele, “Computational Prediction of Flow-Generated Sound,” *Annu. Rev. Fluid Mech.*, vol. 38, no. 1, pp. 483–512, Jan. 2006.

[42] M. Billson, L. E. Eriksson, and L. Davidson, “Jet Noise Prediction using Stochastic Turbulence Modeling,” in *9th AIAA/CEAS Aeroacoustics Conference*, 2003.

[43] C. Bailly and D. Juve, “Numerical Solution of Acoustic Propagation Problems Using Linearized Euler Equations,” *AIAA J.*, vol. 38, no. 1, 2000.

[44] M. J. Lighthill, “On Sound Generated Aerodynamically, I,” *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 211, pp. 564–587, 1952.

[45] M. J. Lighthill, “On Sound Generated Aerodynamically, II,” *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 222, pp. 1–32, 1954.

[46] M. S. Howe, “Contributions to the Theory of Aerodynamic Sound,” *J. Fluid Mech.*, vol. 71, pp. 625–673, 1975.

[47] J. Hardin and D. S. Pope, “A New Technique for Aerodynamic Noise Calculation,” in *14th DGLR/AIAA Aeroacoustics Conference*, 1992.

[48] J. Hardin and D. S. Pope, “An Acoustic/Viscous Splitting Technique for Computational Aeroacoustics,” *Theor. Comput. Fluid Dyn.*, vol. 6, pp. 323–340, 1994.

[49] R. Ewert and W. Schroder, “Acoustic Perturbation Equations based on Flow Decomposition via Source Filtering,” *J. Comput. Phys.*, vol. 188, no. 2, pp. 365–398, 2003.

[50] K. S. Brentner and F. Farassat, “Analytical Comparison of the Acoustic Analogy and Kirchhoff Formulation for Moving Surfaces,” *AIAA J.*, vol. 36, no. 8, pp. 1379–1386, 1998.

[51] A. S. Lyrintzis, “Review: The Use of Kirchhoff’s Method in Computational Aeroacoustics,” *J. Fluids Eng.*, vol. 116, pp. 665–676, 1994.

[52] K. S. Brentner, E. K. Koutsavdis, and A. S. Lyrintzis, “A Comparison of Computational Aeroacoustic Prediction Methods for Transonic Rotor Noise,” *J. Aircr.*, vol. 34, no. 4, pp. 531–538, 1997.

[53] B. A. Singer, K. S. Brentner, and D. P. Lockard, “Simulation of Acoustic Scattering From a Trailing Edge,” *J. Sound Vib.*, vol. 230, no. 3, pp. 541–560, 2000.

[54] N. Curle, “The Influence of Solid Boundaries upon Aerodynamic Sound,” *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 231, no. 1187, pp. 505–514, 1955.

[55] J. E. F. Williams and D. L. Hawkings, “Sound Generation by Turbulence and Surfaces in Arbitrary Motion,” *Philos. Trans. R. Soc. London Ser. A Math. Phys. Sci. 19341990*, vol. 264, no. 1151, pp. 321–342, 1969.

[56] F. Farassat and G. P. Succi, “The prediction of helicopter rotor discrete frequency noise,” *Vertica*, vol. 7, no. 4, pp. 309–320, 1982.

[57] K. S. Brentner and F. Farassat, “Modeling aerodynamically generated sound of helicopter rotors,” *Prog. Aerosp. Sci.*, vol. 39, no. 2–3, pp. 83–120, Feb. 2003.

[58] P. Ma, “Computational-Fluid-Dynamics-Based Modeling of WInd Turbine Noise,” Waterloo, 2011.

[59] G. C. Waller, “Prediction of Flap-Edge Noise Using STAR-CD,” in *14th AIAA/CEAS Aeroacoustics Conference (29th AIAA Aeroacoustics Conference)*, 2008.

[60] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniqus*. Upper Saddle River: P T R Prentice Hall, 1993.

[61] A. Greensted, “Delay Sum Beamforming,” 2010. [Online]. Available: http://www.labbookpages.co.uk/audio/beamforming/delaySum.html. [Accessed: 20-Dec-2011].

[62] G. DeMuth, “Frequency domain beamforming techniques,” in *ICASSP ’77. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1977, vol. 2, pp. 713–715.

[63] D. Lockard, “Summary of the Tandem Cylinder Solutions from the Benchmark problems for Airframe Noise Computations-I Workshop,” in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011.

[64] W. Devenport, R. A. Burdisso, H. Camargo, E. Crede, M. Remillieux, M. Rasnick, and P. Van Seeters, “Aeroacoustic Testing of Wind Turbine Airfoils Aeroacoustic Testing of Wind Turbine Airfoils,” 2010.

Appendix I OpenFOAM ControlDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.0.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     pisoFoam;

startFrom       latestTime;

startTime       10;

stopAt          endTime;

endTime         11.5;

deltaT          5e-6;

writeControl    adjustableRunTime;

writeInterval   0.1;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression off;

timeFormat      general;

timePrecision   8;

runTimeModifiable true;

maxCo           0.5;

functions
{
    forces
    {
        type            forceCoeffs;
        functionObjectLibs ( "libforces.so" );
        outputControl   timeStep;
        outputInterval  1;

        patches
        (
            AIRFOIL
        );

        pName       p;
        UName       U;
        log         true;
```

```
        rhoName rhoInf;
        rhoInf      1.2041;
        CofR        ( 0 0 0 );
        liftDir     ( 0 1 0 );
        dragDir     ( 1 0 0 );
        pitchAxis   ( 0 0 1 );
        magUInf     28.16;
        lRef        0.914;
        Aref        0.0835396;
    }

    wallPressure
    {
        type            surfaces;
        functionObjectLibs ("libsampling.so");
        outputControl   timeStep;
        outputInterval  1;
        surfaceFormat   foamFile;
        interpolationScheme cell;
        fields
        (
            p
        );

        surfaces
        (
            AIRFOIL_WALL
            {
                type        patch;
                patches     (AIRFOIL);
                triangulate false;
            }
        );
    }
}


// ************************************************************************* //
```

Appendix II OpenFOAM fvSchemes

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.0.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default         backward;
}

gradSchemes
{
    default         Gauss linear;
}

divSchemes
{
    default         none;
    div(phi,U)      Gauss limitedLinearV 1;
    div(phi,k)      Gauss limitedLinear 1;
    div(phi,nuTilda) Gauss vanLeer;
    div((nuEff*dev(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default         none;
    laplacian(nuEff,U) Gauss linear corrected;
    laplacian(DkEff,k) Gauss linear corrected;
    laplacian((1|A(U)),p) Gauss linear corrected;
    laplacian(DnuTildaEff,nuTilda) Gauss linear corrected;
    laplacian(1,p) Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         corrected;
}

fluxRequired
{
    default         no;
    p;
}


// ************************************************************************* //
```

Appendix III OpenFOAM fvSolution

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.0.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    p
    {
        solver          PCG;
        preconditioner
        {
            preconditioner  GAMG;
            nVcycles        2;
            smoother        GaussSeidel;
            nPreSweeps      2;
            nPostSweeps     2;
            nCellsInCoarsestLevel    20;
            agglomerator    faceAreaPair;
            mergeLevels     1;
        }
        tolerance       1e-09;
        relTol          0.01;
    minIter    1;
    maxIter    50;
    }

    pFinal
    {
        solver          PCG;
        preconditioner
        {
            preconditioner  GAMG;
            nVcycles        2;
            smoother        GaussSeidel;
            nPreSweeps      2;
            nPostSweeps     2;
            nCellsInCoarsestLevel    20;
            agglomerator    faceAreaPair;
            mergeLevels     1;
        }
        tolerance       1e-08;
        relTol          0.01;
    minIter    1;
    maxIter    50;
    }

    U
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-09;
        relTol          0.1;
```

```
    }


    nuTilda
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-10;
        relTol          0.1;
    }
}

PISO
{
    nCorrectors     2;
    nNonOrthogonalCorrectors 0;
    pRefCell        0;
    pRefValue       0;
}
relaxationFactors
{
    default        1;
    p              0.3;
    U              0.4;
    nuTilda         0.7;
}


// ************************************************************************* //
```

Appendix IV Matlab Beamforming Code

```
ST = 10; %first file to read, aka start time
FT = 11.5; %last file to read, aka finish time
TS = 5e-6; %timestep size
c= 340;  % sound speed
p = dlmread('p_r7.dat');
r = dlmread('r7.dat');
center = [0 0 0];
grid = zeros(2010,0);
disp('read done');
for i = 1:201
    for j = 1:10
        x = (i-1)*10+j;
        grid(x,1) = -0.0914 + (i-1)*0.0914/10;
        grid(x,3) = -0.0914/2 + (j-1)*0.0914/10;
    end
end
disp('grid done');
grid2 = grid;
[a, b] = size(r);
deltam = zeros(a,1);
maxd = 0;
mind = 0;
result = zeros(18,2010);
n=size(p,1);
fs=5e-6;
for i = 1:2010
    kappa = (grid2(i,:)-[0 3 0])./norm((grid2(i,:)-[0 3 0]));
    temp = zeros(size(p,1),1);
    for j = 1:a
        deltam(j,1) = dot(kappa,(r(j,:)-[0 3 0]));
        deltam(j,1) = round(deltam(j,1)/c/TS);
        if deltam(j,1) > maxd
            maxd = deltam(j,1);
        elseif deltam(j,1) < mind
            mind = deltam(j,1);
        end
        prr = p(:,j);
        prr2 = circshift(prr, deltam(j,1));
        temp(:,1) = temp(:,1) + prr2;

    end
    temp((length(temp)-2000):length(temp),:) = [];
    temp(1:2000,:) = [];
    n=size(temp,1);
    temp = abs(fft(temp))/n;
    dfre=1/fs/n;
    fre=dfre*(0:(n-1));
    [otfre, ottemp] = narrow_to_one_third_octave(fre,temp);
    otspl=20*log10(ottemp/0.00002);
    result(:,i) = otspl;
    disp(strcat('grid ', num2str(i), ' finished'));
end
```