PARALLEL METHODS FOR UNSTEADY, SEPARATED FLOWS AND AERODYNAMIC NOISE PREDICTION

A Thesis in
Aerospace Engineering
by
Anupam Sharma

© 2001 Anupam Sharma

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2001
We approve the thesis of Anupam Sharma.

Date of Signature

______________________________  ______________________________
Lyle N. Long                   
Professor of Aerospace Engineering
Thesis Adviser

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

______________________________  ______________________________
Dennis K. McLaughlin           
Professor of Aerospace Engineering
Head of the Department of Aerospace Engineering
I grant The Pennsylvania State University the nonexclusive right to use this work for the University’s own purposes and to make single copies of the work available to the public on a not-for-profit basis if copies are not otherwise available.

______________________________

Anupam Sharma
ABSTRACT

Two problems involving unsteady, separated flows are numerically investigated. The first problem involves the simulation of the airwake over the LPD 17 ship, and the second deals with the numerical prediction of the aerodynamic noise from a cone in a uniform flow.

The flow over the San Antonio class LPD 17 ship is simulated using Computational Fluid Dynamics (CFD) and parallel computers. Both steady-state and time-accurate results are presented for two yaw cases. The computations are done for inviscid flow with no turbulence modeling. Although the flow is highly turbulent and boundary layer effects may not be insignificant, the essential features such as vortex shedding are captured in these simulations because the body has sharp edges. Time accurate simulations are performed for up to 200,000 time steps. The steady state solution is obtained by time averaging the time accurate data, or by running the code in pseudo-time. The steady-state solution and the frequency spectrum are compared with the wind tunnel experiments and are found to be in good agreement. The dominant Strouhal number is found to be of the order unity.

Aerodynamic noise from a cone in a uniform flow is computed using the Ffowcs Williams-Hawkings (FW-H) equation. The time accurate flow data is obtained using a finite volume flow solver on an unstructured grid. The FW-H equation is solved for surface integrals over a permeable surface away from the cone. Predictions from the FW-H code
are compared with direct calculations by the flow solver at a few observer locations inside the computational domain. A very good qualitative match is obtained. Sound directivity patterns in the azimuthal and in the longitudinal directions are presented. The FW-H code is also validated against a model problem of a monopole in a uniform mean flow.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>ABSTRACT</th>
<th>iii</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td>xii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>xv</td>
</tr>
<tr>
<td>1  Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2  Parallel Computers and the Flow Solver</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Parallel Computers</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Flow Solver</td>
<td>6</td>
</tr>
<tr>
<td>2.2.1 Governing Equations</td>
<td>7</td>
</tr>
<tr>
<td>3  LPD 17 Ship Airwake Simulations</td>
<td>9</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>9</td>
</tr>
<tr>
<td>3.2 The Unstructured Grid</td>
<td>12</td>
</tr>
<tr>
<td>3.3 Simulations and Comparison with the Experiments</td>
<td>16</td>
</tr>
<tr>
<td>3.3.1 Pseudo Steady State</td>
<td>17</td>
</tr>
<tr>
<td>3.3.2 Time Accurate Simulations</td>
<td>23</td>
</tr>
</tbody>
</table>
3.4 Time-Averaged Solution .............................................. 26
3.5 Conclusions ............................................................. 35

4 Aerodynamic Noise Prediction using Unstructured Grids ........... 42
  4.1 Introduction ........................................................... 42
  4.2 The Grid ............................................................... 43
  4.3 CFD Results .......................................................... 47
  4.4 Far-Field Noise Prediction .......................................... 52
    4.4.1 The FW-H code and its Validation .......................... 56
    4.4.2 Results for the Cone ......................................... 59
    4.4.3 Sound Directivity ............................................. 60
  4.5 Conclusions .......................................................... 65

5 Future Work ............................................................ 66

REFERENCES ............................................................. 68

A Velocity Potential from a Monopole in a Uniform Mean Flow ....... 73
  A.1 Particular case : $U_0 = U_0\hat{i}$ .................................. 74

B Parallel Algorithm to Obtain Directivity Patterns ...................... 77
LIST OF FIGURES

2.1 Cluster of PCs on shelves - COCOA ................................. 4
2.2 Rack-mounted machines - COCOA2 ................................. 5
2.3 Parallel speed up for COCOA and COCOA2 ......................... 6

3.1 LPD 17 - a schematic showing the various compartments on LPD17. .......................... 13
3.2 The port out-board profile of LPD 17. ................................. 13
3.3 The surface grid on the LPD 17 generated using GridGen. ................................. 15
3.4 A slice of the volume grid at X = 108 m. ................................. 15
3.5 A slice of the volume grid at the center plane (Y = 0). ................................. 16
3.6 The points at which experimental data is collected for the 0° yaw case. ................................. 18
3.7 Recirculation and re-attachment behind the hangar. ................................. 19
3.8 Recirculation in between the two masts. ................................. 20
3.9 Velocity contours at the center plane (Y=0) for the 0° yaw case (pseudo steady-state). ................................. 21
3.10 Cp contours at the center plane (Y=0) for the 0° yaw case (pseudo steady-state). ................................. 21
3.11 Velocity contours at the center plane (Y=0) for the 30° yaw case (pseudo steady-state). ................................. 22
3.12 $C_p$ contours at the center plane ($Y=0$) for the 30° yaw case (pseudo steady-state). ........................................... 22

3.13 Velocity contours at $X = 153$ m for the 0° yaw case (pseudo steady-state). .................................................. 23

3.14 Velocity contours at $X = 153$ m for the 30° yaw case (pseudo steady-state). .................................................. 24

3.15 Frequency Spectrum of $'v'$ (0° yaw; location - $X = 170$ m, $Y = 0$, $Z = 17$ m). .................................................. 27

3.16 Frequency Spectrum of $'w'$ (0° yaw; location - $X = 170$ m, $Y = 0$, $Z = 17$ m). .................................................. 27

3.17 Frequency Spectrum of $|V|^2$ (0° yaw; location - $X = 170$ m, $Y = 0$, $Z = 17$ m). .................................................. 28

3.18 Frequency Spectrum of $'v'$ (30° yaw; location - $X = 170$ m, $Y = 0$, $Z = 25$ m). .................................................. 28

3.19 Frequency Spectrum of $'w'$ (30° yaw; location - $X = 170$ m, $Y = 0$, $Z = 25$ m). .................................................. 29

3.20 Frequency Spectrum for $|V|^2$ (30° yaw; location - $X = 170$ m, $Y = 0$, $Z = 25$ m). .................................................. 29

3.21 Velocity contours at the center plane ($Y=0$), for the 0° yaw case (simulated time-averaged solution). ......................... 30

3.22 Velocity contours at the center plane ($Y = 0$) for the 30° yaw case (simulated time-averaged solution). ......................... 31

3.23 Velocity contours at $X = 150, 165, 180, 195$ and 210 m for the 0° yaw case (simulated time-averaged solution). ............... 31
3.24 Velocity contours at $X = 150, 165, 180, 195$ and $210$ m, for the $30^\circ$ yaw case (simulated time-averaged solution). ... 32
3.25 Surface streamlines on $X = 153$ m plane, for the $30^\circ$ yaw case (simulated time-averaged solution). ... 33
3.26 Velocity contour plot at $X = 165$ m for the $0^\circ$ yaw case (simulated time-averaged solution). ... 34
3.27 Velocity contour plot at $X = 165$ m for the $0^\circ$ yaw case (experimental time-averaged solution). ... 34
3.28 Velocity contour plots at $X = 165, 180, 195, 210$ and $225$ m for the $0^\circ$ yaw case (simulated time-averaged solution). ... 35
3.29 Velocity contour plots at $X = 165, 180, 195, 210$ and $225$ m for the $0^\circ$ yaw case (experimental time-averaged solution). ... 36
3.30 Longitudinal velocity distribution along the center line ($Y = 0$ and $Z = 22$ m), for $0^\circ$ yaw (steady-state solution). Solid line - simulations; crosses - experiments. ... 36
3.31 Transverse velocity distribution along the center line ($Y = 0$ and $Z = 22$ m), for $0^\circ$ yaw (steady-state solution). Solid line - simulations; crosses - experiments. ... 37
3.32 Vertical velocity distribution along the center line ($Y = 0$ and $Z = 22$ m), for $0^\circ$ yaw (steady-state solution). Solid line - simulations; crosses - experiments. 37
3.33 Longitudinal velocity distribution across the deck ($X = 170$ m and $Z = 22$ m), for 0° yaw (steady-state solution). Solid line - simulations; crosses - experiments. ................................................................. 38

3.34 Transverse velocity distribution across the deck ($X = 170$ m and $Z = 22$ m), for 0° yaw (steady-state solution). Solid line - simulations; crosses - experiments. ................................................................. 38

3.35 Vertical velocity distribution across the deck ($X = 170$ m and $Z = 22$ m), for 0° yaw (steady-state solution). Solid line - simulations; crosses - experiments. ................................................................. 39

3.36 Longitudinal velocity distribution above the deck ($X = 170$ m and $Y = 0$), for 0° yaw (steady-state solution). Solid line - simulations; crosses - experiments. ................................................................. 39

3.37 Transverse velocity distribution above the deck ($X = 170$ m and $Y = 0$), for 0° yaw (steady-state solution). Solid line - simulations; crosses - experiments. ................................................................. 40

3.38 Vertical velocity distribution above the deck ($X = 170$ m and $Y = 0$), for 0° yaw (steady-state solution). Solid line - simulations; crosses - experiments. ................................................................. 40

4.1 Overall view of the 280,000 cell mesh. ................................................................. 45

4.2 Partitioning of the FW-H surface across 8 processors. ................................................................. 47

4.3 Instantaneous vorticity contours. ................................................................. 48

4.4 Average streamlines over one shedding cycle. ................................................................. 49
4.5 Comparison of the averaged streamwise velocity with experiments for the
flow solver with and without LES. ........................................ 50

4.6 Comparison of the Cp coefficient with experiment for the flow solver with
and without LES. .......................................................... 51

4.7 Comparison of averaged streamwise perturbation velocity with experiments
with and without LES. ....................................................... 53

4.8 Validation of the FW-H code against the analytical solution for a stationary
monopole in a uniform mean flow. ....................................... 57

4.9 Comparison of the FW-H prediction using unstructured surface grid against
the analytical solution. ...................................................... 59

4.10 Comparison of pressure fluctuation, $p - p_\infty$ as predicted by PUMA and
FW-H code at various locations listed in Table 4.1 ................... 62

4.11 Directivity of the noise in the azimuthal direction behind the base of the
cone ($x = -0.1$). .......................................................... 63

4.12 Directivity of the noise in the longitudinal direction in the plane $z = 0$. ... 64

4.13 Polar plot of sound directivity in $z = 0$ plane at a few radial locations. ... 65

B.1 Algorithm to run multiple serial jobs in parallel ....................... 78

B.2 Megaflops versus number of processors ............................. 79

B.3 Total time versus number of processors .............................. 79
NOMENCLATURE

\( C_p \)       \( C_{p} \) coefficient of pressure
\( C_s \)       \( C_s \) sub-grid scale constant in Smagorinsky model
\( c \)         \( c \) sound speed in quiescent medium
\( d \)         \( d \) base diameter of the cone
\( f_s \)       \( f_s \) vortex shedding frequency (Hz)
\( H(f) \)      \( H(f) \) Heaviside function, \( H(f) = 0 \) for \( f < 0 \) and \( H(f) = 1 \) for \( f > 0 \)
\( L_i \)        \( L_i \) refer
\( L_M \)       \( L_M \) \( \dot{L}_{i} \) \( M_{i} \)
\( L_r \)       \( L_r \) \( \dot{L}_{r} \) \( r_{i} \)
\( \dot{L}_r \)   \( \dot{L}_r \) \( \dot{r}_i \)
\( \textbf{M} \)  \( \textbf{M} \) local Mach number vector of the source
\( M \)         \( |\textbf{M}| \)
\( M_r \)       \( M_r \) \( n_{i} \)
\( M_0 \)       \( U_0/c \)
\( M_r \)       \( M_r \) \( r_{i} \)
\( \dot{M}_r \)   \( \dot{M}_r \) \( \dot{r}_i \)
\( \hat{n} \)    \( \hat{n} \) unit normal vector to the surface, \( n_{i} \)
\( P_{ij} \)     \( P_{ij} \) compressive stress tensor with \( p_o \delta_{ij} \) subtracted
$p$ pressure

$p_0$ freestream pressure

$p'$ acoustic pressure, $p - p_o$

$p'_{\text{rms}}$ root mean squared pressure perturbation

$\rho_{\text{rms}}$ root mean squared density perturbation

$ret$ retarded time

$L_i$ Lighthill stress tensor

$t$ observer time

$\theta$ angular location of the observer

$U$ averaged streamwise velocity

$U_{0,\infty}$ freestream velocity

$U_i$ refer Eq. 4.2

$U_n$ $U_i n_i$

$\dot{U}_n$ $\dot{U}_i n_i$

$\ddot{U}_n$ $\ddot{U}_i n_i$

$u_i$ components of local fluid velocity

$u'_{\text{avg}}$ averaged streamwise perturbation velocity

$u_n$ $u_i n_i$

$v_n$ local normal velocity of the source surface

$\delta(f)$ Dirac delta function $\delta(f) = 1$ for $f = 0$, otherwise $\delta(f) = 0$

$\delta_{ij}$ Kronecker delta function, $\delta_{ij} = 1$ for $i = j$, otherwise $\delta_{ij} = 0$
\( \rho \)  
---  
density of the fluid

\( \rho_0 \)  
---  
freestream density of the fluid

\( \rho' \)  
---  
density perturbation, \( \rho - \rho_0 \)

\( \Omega \)  
---  
vorticity \( (s^{-1}) \)

\( \omega \)  
---  
angular frequency of the monopole source

\( \Box^2 \)  
---  
wave operator, \( \Box^2 \sim \left( \frac{1}{c^2} \frac{\partial^2}{\partial t^2} - \nabla^2 \right) \)
ACKNOWLEDGMENTS

I am thankful to Dr Lyle N. Long for his expert advice and guidance throughout this project. I appreciate the state-of-the-art computing facilities extended to me which greatly helped in finishing the project on time. I also express my gratitude towards Dr Philip J. Morris for his useful, critical comments and suggestions.

Thanks are due to all the members of the Airframe Noise group which includes my colleagues Frederic Souliez, Anurag Agarwal and Preetham Rao. I would like to specifically thank Frederic for providing me with time-accurate data for the ‘aerodynamic noise prediction’ problem. I am grateful to Anirudh Modi and Dr Robert Hansen for helping me learn the flow solver and the grid generation software.

I would like to express my gratitude to the U.S. Navy and the University of New Orleans for sponsoring the ship project. I would also like to acknowledge the Naval Surface Warfare Center (Carderock division) for providing us with the experimental data, and Martin Guillot, University of New Orleans, for providing the IGES file of the ship geometry and many helpful suggestions.
Chapter 1

Introduction

This thesis is focused on two problems - (1) the simulation of an airwake over a newly constructed ship - the LPD17, and (2) aerodynamic noise prediction from a cone in a uniform mean flow. Both problems involve unsteady, separated flows.

On a ship, the unsteady fluctuations over the deck present a lot of challenges in the operation of aircraft. Firstly, the take-off and landing operations become extremely difficult and dangerous to perform in gusty winds, and secondly, during the start-up and shut-down, the blades of a helicopter may undergo large-amplitude oscillations leading to catastrophes such as “tunnel strike”. For years, people have relied on experiments to provide them with the unsteady flow data over the ship decks. Such experiments are performed over real ships and are very expensive. Numerical simulation of the airwake offers an inexpensive alternative to achieve the same goal. This thesis presents the numerical simulations performed on a LPD17 ship.

The second problem addressed in this thesis belongs to the class of Computational Aeroacoustics (CAA). The purpose of these computations is to test if unstructured grids can be used for Aeroacoustics applications which require high accuracy. The flow over a cone was chosen as the model problem. The ultimate aim of this research is to predict
the aerodynamic noise from complicated geometries such as landing gear. Unstructured grids are much easier to generate over such complicated shapes and provide much better control over the cell distribution, and hence it is a better choice. The Ffowcs-Williams and Hawkings equation which is based on Lighthill’s acoustic analogy is used for far-field noise prediction.

This thesis is divided into five chapters. The following chapter describes the machines used for the computations and the unstructured flow solver. The ship airwake problem is discussed in the third chapter and the fourth chapter examines the aerodynamic-noise problem. Since the two problems discussed in this thesis are each somewhat distinct, the third and fourth chapters include detailed introductions and conclusions to each topic. The final chapter presents some possible future work.
Chapter 2

Parallel Computers and the Flow Solver

2.1 Parallel Computers

The two problems discussed in this thesis are very computationally intensive. Even with very powerful machines, such jobs may require days, or even months to give results. Parallel computing using Beowulf clusters offers an inexpensive way to handle such time-consuming simulations in a reasonable amount of time.

Three facilities offering parallel computational power at Penn State have been used for the computations - COst effective COnputing Array (COCOA) [1], COCOA2 and LionX [2]. Figures 2.1 and 2.2 are photographs of the COCOA and COCOA2 clusters respectively, located in the Aerospace Engineering Department of the Pennsylvania State University. COCOA is a Beowulf cluster with 25 machines each having dual 400 MHz Pentium II processors. The machines are connected via a fast-Ethernet network which can support up to 100 Mbps bandwidth. A single Baynetworks 24-port fast-Ethernet switch with a backplane bandwidth of 2.5 Gbps is used for the networking. All the processors are dedicated to run parallel jobs. The operating system is Red Hat Linux. Message Passing Interface (MPI) is used for parallel programming and the Gnu C compiler is used for compiling the flow solver. Details regarding setting up and benchmarking of COCOA may be
COCOA was primarily set up to make parallel computing facility readily available to the Computational Fluid Dynamics (CFD) group of the Aerospace Engineering Department at the Pennsylvania State University. The total cost of the cluster was just $80,000 in the year 1998, when it was set up. Since then this facility has been used intensively for various CFD simulations [4]. COCOA2 is a newly assembled Beowulf cluster at Penn State. It has 21 nodes each having dual 800 MHz Pentium III processors and 1 GB RAM each. The cluster has dual fast-Ethernet per node and all the nodes are connected using two HP2524 switches with channel bonding.
Figure 2.2: Rack-mounted machines - COCOA2

Figure 2.3 plots the parallel speedup for COCOA and COCOA2 for the aeroacoustic problem considered in Chapter 4 solved using the flow solver to be described in the next section. (1 Mflop = one million floating point operations per second). Fairly good performance is obtained considering the small size of the problem. Figure 2.3 shows the reduction in the flop rate per processor as the grid points are distributed over a larger number of processors. This trend is typical of Beowulf clusters as the ratio of computation over communication decreases.

LionX is also a Beowulf cluster with 32 machines (each having dual 400 MHz Intel Xeon processors). These machines are connected via Myricom Myrinet with wire speed
Figure 2.3: Parallel speed up for COCOA and COCOA2

1.28 Gbps. LionX also uses Linux with MPI for parallel programming. Performance comparison and benchmarking results for LionX can be obtained from its website [2].

2.2 Flow Solver

Parallel Unstructured Maritime Aerodynamics (PUMA) is a computer program, written in C, for the analysis of internal and external non-reacting compressible flows over arbitrary complex geometries. PUMA uses the Message Passing Interface (MPI) to run the code in parallel. It can be run on an arbitrary number of processors with very good scaling performance. Several papers [1, 5] detail the benchmarking of the performance and validation of PUMA.
PUMA is based on finite volume methods and supports mixed topology unstructured grids composed of tetrahedra, wedges, pyramids and hexahedra. The code may be run to preserve time accuracy for unsteady problems, or may be run using a pseudo-unsteady formulation to enhance the convergence to the steady state. The second and fourth order accurate Runge-Kutta scheme is implemented for time accurate computations. Other schemes such as Jacobi, Gauss-Seidel and Symmetric Successive Over Relaxation (SSOR) are available for steady state simulations. The flux computations are performed using the Roe or Vanleer schemes. The Courant-Friedrichs-Levy (CFL) number and slope for time marching can be explicitly specified in the input file. Primitive flow quantities are computed at the cell centers. The code can be restarted from any point of time at which the solution is available from previous computations. All flow variables are stored with double precision, but may be optionally stored as single precision to save memory and communication time at the cost of reduced precision. More information about PUMA may be obtained from its manual [6].

2.2.1 Governing Equations

In both the problems discussed in this thesis, the Euler equations are solved by neglecting the body forces and using no turbulence modeling. The Euler equations, Eqs. 2.1, 2.2 and 2.3 are written in the integral form and solved in PUMA using a finite volume
formulation.

\[ \frac{\partial}{\partial \tau} \int \rho dV + \int_S (\rho \ddot{\mathbf{u}}).d\mathbf{S} = 0 \]  \hspace{1cm} (2.1)

\[ \frac{\partial}{\partial \tau} \int \rho \mathbf{u} dV + \int_S (\rho \ddot{\mathbf{u}}).\mathbf{u} d\mathbf{S} + \int_S p d\mathbf{S} = 0 \]  \hspace{1cm} (2.2)

\[ \frac{\partial}{\partial \tau} \int \rho E dV + \int_S (\rho E + p) \ddot{\mathbf{u}}.d\mathbf{S} = 0 \]  \hspace{1cm} (2.3)

The system is closed by the equation of state for an ideal gas.

\[ p = \rho RT \]  \hspace{1cm} (2.4)

In the above equations, \( S \) denotes the surface of the control volume \( V \). An overhead vector, \( \ddot{\mathbf{u}} \) is used to represent vector quantities. \( p \) is pressure, \( \rho \) is fluid density, \( T \) is temperature, \( \mathbf{u} \) is velocity, \( E \) is the energy per unit mass and \( R \) is the universal gas constant.

Since no turbulence modeling is used, the smallest turbulent scales that are captured in these simulations are of the order of the grid size. Boundary layer effects are ignored by making an inviscid flow approximation.
Chapter 3

LPD 17 Ship Airwake Simulations

This chapter describes the results of an airwake simulation over the LPD 17 ship. The simulations are aimed at providing an idea of the general flow pattern and the dominant shedding frequencies. This information is crucial for the safety of the helicopters that operate from the ship.

3.1 Introduction

The increasing use of helicopters in conjunction with ships presents the need to accurately predict the abnormal behavior of a helicopter when operated in proximity to a ship. The ship-helicopter dynamic interface offers a multitude of challenges [7]. “Blade Sailing” is one of such problems of paramount importance. The blade-sailing phenomenon is commonly observed when operating a helicopter from a ship deck. In severe conditions blade sailing may result in a catastrophic “tunnel-strike” which can severely damage the rotor blades and also cost lives. These problems occur during the start-up and shut down of the rotor, when the centrifugal forces are small. The weak centrifugal force cannot offer enough resistance to the impulsive lift/drag force which can cause large-amplitude oscillations of the blades in the longitudinal/lateral direction. The problem is aggravated when operating from a ship deck because of the flow separation from the sharp edges of the mast,
the deck and the hangar. The shedding of vortices from the edges characterizes the time variation of the flow over the deck. There is a chance that the shedding frequency will match the angular frequency of the rotor and excite the blades in resonance. This may amplify the deflections to such a level that the rotor strikes the tail boom - “tunnel strike”. The problem of blade sailing is particularly addressed by Newman [8].

Keller and Smith [9] and Keller [10] attempted to model the velocity distribution over the deck using simple linear models. They concluded that such simplified models cannot accurately predict the severe blade sailing observed in practice. This identifies the need to obtain the ‘real’ airwake data either experimentally or numerically.

Another problem which is more commonly observed and is perhaps more important than the “tunnel strike” problem is that of take-off and landing from a ship deck. The increased pilot workload during these phases of the flight of a helicopter cannot be ignored. Until recently, preventive experimental measures have been used to study safety issues. The idea is to locate the safest regions on the ship deck where landing/take-off operations should be performed. Thorough experimental investigations are performed to identify Ship Helicopter Operating Limits (SHOLs) for each ship-helicopter combination. A helicopter can consistently operate safely from within a region called the safe-operating envelope, which is bounded by these SHOLs. The process of determining these envelopes is both slow and expensive. The determination of these envelopes for each ship-helicopter combination may cost between $75,000 - $150,000. Besides, the SHOLs significantly change with the conditions at sea, and hence the ‘safe’ envelopes may not necessarily be safe in all
conditions. There are various other problems associated with obtaining experimental data in adverse conditions - when the sea is rough, or when it’s very windy. Above all, the data obtained in such conditions cannot be accepted with confidence since there are so many variables.

An alternative to this laborious process is to numerically obtain the ship airwake and use that with some dynamic model of a rotor to predict the blade response. This idea is appealing as it significantly reduces the cost and time. It is time-efficient because once the time-accurate data for the ship airwake is available, it can be used with any helicopter model to obtain the response in minutes. The challenge however is to simulate the flow accurately in time. This has motivated the CFD community to simulate the ship airwake using parallel computers. The first difficulty is to deal with the complicated geometry of the ship. There is no definite length-scale which can be used for such geometries, which further complicates the problem. The biggest challenge that still remains is the lack of computational power. Even with today’s powerful parallel computers, computationally intensive jobs like these take days or even months to complete.

A number of attempts have been made to numerically simulate the helicopter/ship dynamics. Healy [11][12] and Johns and Healy [13] addressed the problems with shipboard rotors and looked into the prospects of simulating the ship-helicopter dynamic interface. Healy concluded that the simulation of the airwake is the most challenging and computationally intensive task in this problem. Healy [11] also highlighted the need for more accurate experimental data on real ships. Liu and Long [14], Tattersall, Albone and Allen [15]
and Tai [16] have independently approached the problem of simulating ship airwake. These references suggest different numerical techniques for such simulations.

The present study reports the results of simulations of the LPD 17 ship airwake, and comparison with the wind tunnel experiments. The simulations are performed for two yaw cases. Both steady state and time accurate results are compared. The frequency spectra are compared with the experiments to estimate the Strouhal number of the flow. An extensive comparison is made for the 0° yaw case and the results from the simulations of the 30° yaw case are presented too. This work has been presented at the 15th AIAA CFD conference held in Anaheim, California in June 2001 [17].

3.2 The Unstructured Grid

The San Antonio LPD 17 (Figs. 3.1 and 3.2) is the US Navy’s newest class of ship. It is a warfare capable ship and its primary mission is amphibious warfare. An amphibious operation is an attack launched from the sea by naval and landing forces embarked in helicopters, landing craft, and amphibious vehicles on a hostile shore. The total length of the ship is 200 m (waterline) and 208 m (full). The extreme beam (width) is 32 m. The ship can sustain a speed of 22+ knots. The ship deck can land 2 CH-53Es or 4 CH-46s or 2 MV-22s. The hangar can accommodate 1 CH-53E or 2 CH-46s or 1MV-22 or 3 VH/AH-1. The prominent features which mark the geometry of LPD 17 are the two masts one behind the other separated by a distance of about 90 m.

For a complicated geometry like the LPD 17, an unstructured grid is the obvious
Figure 3.1: LPD 17 - a schematic showing the various compartments on LPD17.

Figure 3.2: The port out-board profile of LPD 17.
choice. The grid used for the simulations was generated using Gridgen \[18\]. Some intricate details of the geometry such as the fences were ignored for simplicity. This simplification may cause a loss of some small scale fluctuations which, fortunately, are unimportant for the present study. Figure \[3.3\] shows the grid point distribution over the surface of the ship. Clustering is done all around the ship with increasing cell size towards the boundaries of the bounding box. The computational domain extends to a distance of \(2L\) both in front and behind the ship, where \(L = 200\) m is the ship length. The side outer boundaries are at a distance of \(7 \times W\) from the center plane, where \(W = 30\) m is the width of the ship deck. The waterline is taken to be the bottom surface of the bounding box. The upper outer boundary is at a distance of \(6 \times H\) from the waterline, where \(H = 50\) m is the maximum height of the ship. Figures \[3.4\] and \[3.5\] show the size of the bounding box in relation to the size of the ship.

All the computations are performed for inviscid flow. Hence, the only boundary condition imposed at the ship surface is that the velocity normal to the surface is zero. The waterline which is the bottom face of the bounding box is also assigned the zero-normal-velocity condition. All the other faces of the bounding box are assigned the Riemann boundary condition to avoid any reflections into the domain.

Since the time step for time-accurate computations (using an explicit scheme) is determined by the smallest cell in the volume grid, a fairly uniform surface grid is chosen (Fig. \[3.3\]). The total number of tetrahedra in the volume grid is around 0.2 million. The minimum cell length is about 7 mm which gives a time step of \(6 \times 10^{-5}\) seconds for a
Figure 3.3: The surface grid on the LPD 17 generated using GridGen.

Figure 3.4: A slice of the volume grid at $X = 108$ m.
Figure 3.5: A slice of the volume grid at the center plane ($Y = 0$).

Courant-Friedrichs-Lewy (CFL) number of 0.8.

3.3 Simulations and Comparison with the Experiments

Two cases have been studied: $0^\circ$ head wind, and a cross wind at a yaw angle of $30^\circ$ (from port side) to the ship. The yaw angle is defined as the angle the velocity vector makes with the center plane of the ship. Since the ship is aligned with the X axis in the simulations, the yaw angle is just the angle between the X axis and the velocity vector. The positive X axis is chosen to point towards the rear from the bow of the ship. The positive Z axis points vertically up, and the Y axis forms a right-handed coordinate system. The origin is located at the point where the bow of the ship intersects the waterline.

For both cases the magnitude of the freestream velocity is 30 kts. (15.44 m/s). In order to achieve a reasonable (i.e. computationally efficient) Mach number, the freestream pressure value is reduced to 5.2 KPa. With reduced pressure, and freestream air density of 1.226 kg/m³, the speed of sound is 77 m/s. The Mach number is thus scaled by a factor of
It is well known that low Mach number flows are fairly insensitive to Mach number.

All the domain cells are initialized with freestream values. Once the initial conditions are specified, the boundary conditions and the geometry of the ship govern the flow. Flows over bluff bodies like a ship never reach a steady state. The flow pattern keeps changing with time. This change may be periodic or aperiodic, depending on the Reynolds number of the flow. For very high Reynolds number flows, as is the case here, the flow is turbulent and hence aperiodic. The vortices are shed from the bow, the masts, the hangar and other objects protruding from the surface of the ship. It is nearly impossible to analytically estimate the frequencies which would dominate the time variation of the flow over the deck. The aim of these computations is to accurately predict these frequencies and to predict the large scale flow structures.

The experiments were conducted by the Naval Surface Warfare Center (Carderock division) [19]. The experimental model was scaled down by a factor of 94 to accommodate the wind tunnel. Time accurate data was collected at a few points above the deck for two yaw values: 0° and 30°. The sampling rate was 0.02 seconds. Figure 3.6 shows the location of the points at which the experimental data was collected for the 0° yaw case.

### 3.3.1 Pseudo Steady State

Local time stepping (pseudo time marching) is used with the Successive Symmetric Over Relaxation (SSOR) scheme to accelerate the convergence to a physically realistic steady flow. Since there is no fixed steady-state, a physically realistic solution at any point
Figure 3.6: The points at which experimental data is collected for the $0^\circ$ yaw case.
of time is referred to as a “pseudo steady state”. It requires about 600 iterations, which take 6 hours of wall-clock time on 8 processors of COCOA, to reach a “pseudo steady state”. A “pseudo steady state” is a good representative of the time averaged solution. The analysis of a “pseudo steady state” solution can reveal important information about the ‘static’ flow field.

Figures 3.7 and 3.8 show the streamlines for the “pseudo steady state” solution for the 0° yaw case behind the hangar and between the two masts, respectively. The figures clearly show the recirculation. The flow over the hangar is very similar to the flow over a backward facing step. It exhibits the separation of the shear layer and its re-attachment downstream (Fig. 3.7).
Figure 3.8: Recirculation in between the two masts.

Figures 3.9 and 3.10 show the contour plots of the velocity magnitude and the coefficient of pressure ($C_p$) for the 0° yaw case. The plots are drawn at the center plane ($Y=0$). Figures 3.11 and 3.12 show similar plots for the 30° yaw case. Since the stagnation points do not lie on the $Y = 0$ plane for a non-zero yaw case, the $C_p$ is low in front of the masts (Fig. 3.12). A similar trend is observed in the velocity contour plot (Fig. 3.11): high velocity in front of the masts due to the curvature.

The velocity contours are plotted for both yaw cases on a $Y-Z$ plane at $X = 153$ m (Figs. 3.13 and 3.14) to study the effect of yaw on the flow pattern in the lateral direction. This plane is located 15 m behind the hangar. The velocity contours clearly show the drift of the wake in the direction of velocity. For the 30° yaw case, the ship deck behaves like
Figure 3.9: Velocity contours at the center plane (Y=0) for the 0° yaw case (pseudo steady-state).

Figure 3.10: $C_p$ contours at the center plane (Y=0) for the 0° yaw case (pseudo steady-state).
Figure 3.11: Velocity contours at the center plane (Y=0) for the 30° yaw case (pseudo steady-state).

Figure 3.12: $C_p$ contours at the center plane (Y=0) for the 30° yaw case (pseudo steady-state).
Figure 3.13: Velocity contours at $X = 153$ m for the $0^\circ$ yaw case (pseudo steady-state).

a forward facing step and a backward facing step in a tandem arrangement for the lateral velocity component.

3.3.2 Time Accurate Simulations

While the pseudo steady state solution provides a qualitative representation of the flow-field, it does not contain any time dependent information, for example vortex shedding frequency. Time accurate simulations are essential to capture the dynamics of the flow over the deck.

An unstructured grid is not a perfect choice for time-accurate simulations, as it generates a few very small cells which drastically reduce the time step of integration. For explicit schemes, the time step cannot be arbitrarily increased by increasing the CFL because of stability concerns. The wall-clock time for simulating 1 second of real flow increases with
decreasing cell size. Hence the computations become extremely time consuming. Further clustering of the grid in order to accurately resolve the flow increases the time cost in two ways: (1) the number of cells increases i.e. wall-clock time per iteration increases, and (2) the smallest cell size reduces i.e. the time step reduces. The cell-size variation is therefore as important as the number of cells in the grid.

Although the unstructured grid does present some problems, the alternative of using a structured grid over an extremely complicated geometry such as LPD 17 is not feasible. Hence, the same unstructured grid (Fig. 3.3) is used for the time accurate computations. The CFL is kept constant at 0.8 which gives a timestep of $6 \times 10^{-5}$ seconds. The time accurate computations are started from a “pseudo steady state” solution. It takes around 17,000 iterations to simulate 1 second of real flow, and requires about 2 days on 8 processors of COCOA.

Figure 3.14: Velocity contours at $X = 153$ m for the 30° yaw case (pseudo steady-state).
In order to compare the numerical results with the experiments, the primitive flow variables are nondimensionalized as follows:

\[
\tilde{u} = \frac{u}{V_\infty}; \quad \tilde{v} = \frac{v}{V_\infty}; \quad \tilde{w} = \frac{w}{V_\infty}
\]

\[
\tilde{p} = \frac{p}{p_\infty}; \quad \tilde{t} = \frac{t}{T_s}; \quad St = fT_s
\]

where the (\^) quantities are the nondimensional variables, \( \infty \) represents free-stream values, \( V \) is the magnitude of the velocity, \( p \) represents pressure, \( t \) is time and \( T_s \) is the time scale. \( T_s \) is calculated by taking the ratio of the length scale (chosen to be the ship length) and the velocity scale, \( V_\infty \). The Strouhal number (\( St \)) is the nondimensional frequency. \( u, \ v \) and \( w \) are the components of velocity in the \( X, Y \) and \( Z \) directions respectively.

The freestream velocity is identical in the simulations and the experiments, but the geometry is scaled down by a factor of 94 for the experiments. Using similarity principle, the time scale in the experiments should be 94 times smaller than the time scale in the simulations.

The experimental data is available for 40 seconds of real time which is equivalent to 290 units of nondimensional time. The simulations are performed for 10 seconds which is equivalent to 1 unit of nondimensional time. It is known from the experiments that the typical Strouhal number for the problem is about unity. Therefore, the simulations performed for a unit nondimensional time may capture the flow dynamics. A simulation of 1 unit of nondimensional time takes approximately a month of computations on 8 processors.
of COCOA. With this speed, it would take years to simulate 290 units of nondimensional time on 8 processors. This quantifies the magnitude of the problem.

Since the experimental data has been collected over a considerably longer period of time than the simulated data, it resolves the low-frequency components much more accurately. The sampling rate for the simulations is $7 \times 10^{-5}$, whereas for the experiments it is 0.15, in nondimensional time units. Though the simulations should resolve the high-frequency components better than the experiments, it is not necessarily the case here as the small scale turbulence is not captured by the simulations.

Figures 3.15, 3.16 and 3.17 show the frequency spectra of $v$, $w$ and $|V|^2$ respectively, at a point on the center-plane behind the hangar, for the 0° yaw case. The magnitude of the power spectrum is scaled by the peak value between $St = 0.2-25$. Figures 3.18, 3.19 and 3.20 plot the frequency spectra of the same variables at a different point, for the 30° yaw case. Although the simulations do not capture the low-frequency components accurately, they clearly indicate that the dominant frequencies lie between $St = 1 - 3$.

### 3.4 Time-Averaged Solution

It has been argued that a “pseudo steady state” solution does not match well with the time-averaged experimental results [20], while the data obtained on time averaging the time accurate simulations matches fairly well. A comparison between a “pseudo steady state” solution and the time averaged solution is made to examine if this is true for the case in the present study. The time accurate data is sampled for a unit nondimensional time.
Figure 3.15: Frequency Spectrum of ‘v’ (0° yaw; location - X = 170 m, Y = 0, Z = 17 m).

Figure 3.16: Frequency Spectrum of ‘w’ (0° yaw; location - X = 170 m, Y = 0, Z = 17 m).
Figure 3.17: Frequency Spectrum of $|V|^2$ (0° yaw; location - $X = 170$ m, $Y = 0$, $Z = 17$ m).

Figure 3.18: Frequency Spectrum of ‘v’ (30° yaw; location - $X = 170$ m, $Y = 0$, $Z = 25$ m).
Figure 3.19: Frequency Spectrum of ‘w’ (30° yaw; location - X = 170 m, Y = 0, Z = 25 m).

Figure 3.20: Frequency Spectrum for |V|^2 (30° yaw; location - X = 170 m, Y = 0, Z = 25 m).
Figures 3.21 and 3.22 show the velocity contours of the time averaged solution at the center plane for 0° yaw and 30° yaw, respectively. Comparison of Figs. 3.9 and 3.21, and 3.11 and 3.22 shows that a “pseudo steady state” solution represents the time averaged solution fairly well. This might be due to our use of an explicit scheme. Previous studies [20] have used implicit methods. Implicit schemes permit larger time steps to be used (since they are more stable), but this often means that the small cells will have very large local CFL numbers. There may be a loss of accuracy due to this which may account for their difference between pseudo-time and time-averaged solutions.

Figures 3.23 and 3.24 show the velocity contours of the time averaged solution for 0° yaw and 30° yaw, respectively, at a few X locations on the deck. The asymmetry in the wake, even for the 0° yaw case, is apparent in the figures. This is expected as the ship is not symmetric about the Y = 0 plane.

In the time averaged solution for the 30° yaw case (Fig. 3.24), a vortex structure can
Figure 3.22: Velocity contours at the center plane \((Y = 0)\) for the 30° yaw case (simulated time-averaged solution).

Figure 3.23: Velocity contours at \(X = 150, 165, 180, 195\) and 210 m for the 0° yaw case (simulated time-averaged solution).
Figure 3.24: Velocity contours at $X = 150, 165, 180, 195$ and $210$ m, for the $30^\circ$ yaw case (simulated time-averaged solution).

be seen detached from the deck aligned in the direction of the wind. Figure 3.25 shows the surface-streamlines plot on the $X = 153$ m plane. The vortex structure is clearly seen in the figure. These structures are continuously generated and shed in time.

Figures 3.26 and 3.27 compare the time averaged data obtained from simulations with the experiments. The simulations are agree fairly well with the experiments in the wake of the hangar, although there is more asymmetry in the simulated result. The simulations also do not clearly show the wake of the second mast as observed in the experiments. This may be due to two effects: first the experimental data is taken only at 36 points on the plane, and hence cannot capture the details, and second, the simulations are done for just 1-2 cycles and hence may contain the effect of shed vortices. Figures 3.28 and 3.29 plot the time averaged solution at a few $X$ locations from the simulations and from the experiments,
Figure 3.25: Surface streamlines on $X = 153$ m plane, for the 30° yaw case (simulated time-averaged solution).

respectively.

Figures 3.30-3.38 compare the steady state velocity distribution over the deck with the experiments. The simulations follow the velocity pattern in the experiments but under-predict the wake, especially, behind the hangar (Fig. 3.30), and near the surface of the ship deck (Fig. 3.36). This may be because of the inviscid flow assumption, poor grid resolution or because of inadequate sampling for obtaining the steady state. It should be noted that the experimental data is taken at very low flow velocities and could be erroneous.
Figure 3.26: Velocity contour plot at $X = 165$ m for the $0^\circ$ yaw case (simulated time-averaged solution).

Figure 3.27: Velocity contour plot at $X = 165$ m for the $0^\circ$ yaw case (experimental time-averaged solution).
3.5 Conclusions

The airwake over the LPD 17 has been numerically simulated using parallel machines. Two cases have been simulated: 0° and 30° yaw. The results for both time accurate simulations and the steady state are presented for the two cases. Although there is no steady state for the problem, a “pseudo steady state” is found to represent the general flow pattern over the ship. The time averaged data compares very well with a “pseudo steady state” solution, and fairly well with the experiments. The simulations slightly under-predict the wake which may be because of the inviscid flow assumption.

One unit of nondimensional time is simulated accurately in time. Frequency spectra plots from the simulations are compared with the experiments. The experiments strongly
Figure 3.29: Velocity contour plots at $X = 165, 180, 195, 210$ and $225$ m for the $0^\circ$ yaw case (experimental time-averaged solution).

Figure 3.30: Longitudinal velocity distribution along the center line ($Y = 0$ and $Z = 22$ m), for $0^\circ$ yaw (steady-state solution). Solid line - simulations; crosses - experiments.
Figure 3.31: Transverse velocity distribution along the center line \((Y = 0 \text{ and } Z = 22 \text{ m})\), for 0° yaw (steady-state solution). Solid line - simulations; crosses - experiments.

Figure 3.32: Vertical velocity distribution along the center line \((Y = 0 \text{ and } Z = 22 \text{ m})\), for 0° yaw (steady-state solution). Solid line - simulations; crosses - experiments.
Figure 3.33: Longitudinal velocity distribution across the deck ($X = 170$ m and $Z = 22$ m), for $0^\circ$ yaw (steady-state solution). Solid line - simulations; crosses - experiments.

Figure 3.34: Transverse velocity distribution across the deck ($X = 170$ m and $Z = 22$ m), for $0^\circ$ yaw (steady-state solution). Solid line - simulations; crosses - experiments.
Figure 3.35: Vertical velocity distribution across the deck \((X = 170 \text{ m} \text{ and } Z = 22 \text{ m})\), for \(0^\circ\) yaw (steady-state solution). Solid line - simulations; crosses - experiments.

Figure 3.36: Longitudinal velocity distribution above the deck \((X = 170 \text{ m} \text{ and } Y = 0)\), for \(0^\circ\) yaw (steady-state solution). Solid line - simulations; crosses - experiments.
Figure 3.37: Transverse velocity distribution above the deck \((X = 170 \text{ m and } Y = 0)\), for \(0^\circ\) yaw (steady-state solution). Solid line - simulations; crosses - experiments.

Figure 3.38: Vertical velocity distribution above the deck \((X = 170 \text{ m and } Y = 0)\), for \(0^\circ\) yaw (steady-state solution). Solid line - simulations; crosses - experiments.
indicate that the dominant Strouhal number is of the order unity; this is supported by the simulations. The need to sample more time-accurate data to accurately resolve the low Strouhal numbers is highlighted. With such limited data it is difficult to pinpoint the Strouhal number in the wake, but the simulations correctly estimate the order of magnitude of the Strouhal number and the basic flow features are predicted properly. The next chapter discusses the problem of aerodynamic noise prediction from a cone.
Aerodynamic Noise Prediction using Unstructured Grids

This chapter addresses the second problem - computation of aerodynamic noise from bluff bodies. The aim is to be able to predict aerodynamic noise from complicated geometries like landing gear due to the separated flow in the wake of such bluff bodies. The cone case was chosen as a model problem to validate the codes and to justify the use of unstructured grids with a low-order accurate flow solver for aeroacoustic predictions.

4.1 Introduction

Recently, the Ffowcs Williams-Hawkings (FW-H) equation has been used with permeable surfaces for predicting aerodynamic noise [21]. The application of FW-H in this manner effectively allows for the inclusion of the quadrupole source terms inside the surface without performing volume integrations. This has significantly improved the accuracy of noise prediction for cases where the contribution from nonlinear interactions in the flow cannot be ignored. This is typical of highly turbulent flows, for example, high Reynolds number jets and wakes.

The solution of FW-H equation requires time accurate data on, and in the volume outside the permeable surface. This data is usually obtained by solving the Euler/Navier-Stokes equations accurately in time. Since the volume integration can be ignored for low
Mach number flows, the outer grid can be made coarse without much loss of accuracy. Unstructured grids provide great flexibility in distributing the grid in the domain, and hence can be used to cluster the cells inside the FW-H surface. This feature can be exploited to significantly increase the computation speed while keeping almost the same accuracy in predicting aerodynamic noise. This will also permit the modeling of complex geometries such as helicopter fuselages, landing gear, and flaps.

The goal here is to test the combination of unstructured grids with the FW-H equation in predicting the aerodynamic noise. The test case is chosen to be the flow over a cone. A cone has sharp edges which fixes the separation point. This makes the flow fairly Reynold’s number independent.

We use the Parallel Unstructured Maritime Aerodynamics (PUMA) \[22\] code for generating the time-accurate flow data. PUMA has been validated for time-accurate computations \[1, 17, 5\]. The ultimate aim is to predict the airframe noise from complex geometries such as landing gear, slats, and flaps. This cone case may be considered as a benchmark problem.

### 4.2 The Grid

The grid used for the simulation of the flow over a cone of vertex angle 60° was generated using Gridgen. Figure 4.1 shows an overall view of the mesh consisting of approximately 280,000 tetrahedra. Clustering was performed around the cone and in the
wake region with increasing cell size towards the outer boundaries of the computational domain. The reason for using Gridgen comes from one interesting feature of this commercial software: arbitrary surfaces can be created around the cone (one within the CFD domain boundaries and the other being the CFD domain boundary) and are sources for the meshing algorithm. It is possible to export separately any of these closed surfaces in a separate file, providing a means to extract flow data on the surface using a FW-H module that is added to the unstructured solver. The smallest cylinder is used as a porous FW-H surface. At the bounding faces of the CFD domain, Riemann boundary conditions are assigned at each face center, hence minimizing reflections from the boundaries into the computational domain. The large cells in the far-field also help dissipate any reflections. A no-slip condition is used at the solid surface, even though the boundary layer is not resolved due to computer limitations.

By using a set of faces that are actually used by the flow solver during the computation, there is no additional work required to extract the data needed for the far-field noise. This type of FW-H surface also reflects the true mesh clustering present where the flow variables are being computed locally: there is no loss in accuracy due to the interpolation onto a surface whose refinement might not be that of the computational grid. Since only the surface terms are evaluated during the acoustic prediction procedure, one does not have to take into account any phenomenon occurring outside the integration surface. The surface can also cross regions dominated by nonlinear effects.

During a run, the faces (triangles in this case) are identified and flagged on each CPU,
Figure 4.1: Overall view of the 280,000 cell mesh.
so that face data is output at a prescribed sampling rate (around 50 kHz in the present case): the sampling is done in such a manner that one had at least 20 data points per wavelength, the shortest wavelength being 10 times that of the simulated shedding frequency. To avoid any redundant data, faces shared between two adjacent CPUs has to be identified at the beginning of each run, so that the number of faces whose data are output is identical to the number of triangles on the actual FW-H surface. The grid partitioning is done dynamically each time a run is initialized, the global cell indexing changes from run to run, making it necessary to run the above flagging procedure any time the program is restarted. This makes the routine independent of the number of CPUs being used. Figure 4.2 illustrates the regions on the surface shared between 8 processors using the Gibbs-Poole-Stockmeyer reordering algorithm [23]. As expected, each region is a neighbor to at most two other partitions, minimizing the amount of inter-processor communication.

The time step needed for a time-accurate solution is determined by the smallest cell characteristic length. This is estimated to be one third of the cell volume divided by the maximum face area. For the grid described above, this yields a time step of 9.45E-08 seconds at a CFL number of 0.9. The shedding frequency found during the experimental investigation of the flow is 36 Hz, for a Strouhal number equal to 0.171. The Strouhal number is defined based on the cone diameter as \( St = \frac{f_s d}{U_\infty} \). The numerical simulation is performed at Mach 0.2 at standard atmospheric pressure and temperature conditions, with an increased viscosity to match the experiment’s Reynolds number (50,000). Scaling the Strouhal number to the simulation’s Mach number yields a shedding frequency of 230 Hz.
The computation of a complete shedding cycle requires roughly 46,000 iterations.

4.3 CFD Results

After initializing all variables to the freestream values, local time stepping is used to accelerate the convergence towards a physically realistic flow. This is done by assigning to each cell the maximum allowable time step for a given CFL number based on each cell’s characteristic cell length. Global time stepping is then turned on for several cycles before data are sampled, to ensure that the data on the FW-H surface follow the equations of motion. Figure 4.3 illustrates the vorticity patterns in the wake of the cone, showing strong recirculation phenomena. The noise from this recirculation is predicted by the FW-H
module. Figure 4.4 is the averaged streamline contour over one shedding period, illustrating the axisymmetric bubble that was observed during Calvert’s experimental study [24].

In order to validate the solution, multiple comparisons were made between the simulation and the experimental measurements. A basic Smagorinsky sub-grid scale turbulence model [25] was added to the flow solver in order to improve the predictions, since a large-eddy simulation should yield better turbulent quantities.

Figure 4.5 shows the averaged streamwise velocity profiles computed by the original flow solver and those computed by the same solver combined with an LES. In all three cases the magnitude of the reverse flow velocity is under predicted when compared with experimental measurements. The predictions agree fairly well with Calvert’s data in terms
of the length of the recirculation zone. Past the stagnation point, the results including LES modeling follow the experimental curve more closely than those computed without any turbulence model.

Figure 4.4 illustrates the variation of the pressure coefficient $C_p$ along the wake centerline. In this case, the LES having the largest sub-grid scale constant $C_s$ greatly overcorrects the pressure drop in the near wake of the cone. The LES using a Smagorinsky constant of 0.10 matches the measured pressure data very well until the stagnation point is reached. These results are consistent with those found in other related investigations, using either the $k$-$\varepsilon$ turbulence model [26] or the $k$-$\varepsilon$-$v^2$ model [27]. These simulations were compared against a set of experiments [28] at a lower Reynolds number (42,000). Madabhushi
Figure 4.5: Comparison of the averaged streamwise velocity with experiments for the flow solver with and without LES.
Figure 4.6: Comparison of the Cp coefficient with experiment for the flow solver with and without LES.
[29] also used an LES with as many as 850,000 mesh points, but completely over-predicted the length of the recirculation zone.

Figure 4.7 shows that the averaged streamwise perturbation velocity is not well predicted using any of the sub-grid scale constants. With the grid coarsening in the far wake, the fluctuating velocities are damped very rapidly as one goes away from the cone base. It is the flow solver without any turbulence model that yields unsteady velocity values that are closest to the experimental data. The solution without LES was selected to try to predict the far-field noise. It also leads to the conclusion that a more advanced turbulence model (dynamic LES, Detached Eddy Simulation) is needed to simulate such separated flows, as found by Strelets [30].

4.4 Far-Field Noise Prediction

The two commonly used methods for far-field aerodynamic noise predictions use the “moving surface” Kirchhoff equation or the Ffowcs Williams-Hawkings (FW-H) equation. While the governing equation in the moving surface Kirchhoff formulation [31] is a convective wave equation, the FW-H equation is an exact rearrangement of the continuity and the momentum equations into the form of an inhomogeneous wave equation. Therein lies the strength of the FW-H equation over the Kirchhoff formulation. The FW-H equation gives accurate results even if the surface of integration lies in the nonlinear flow region. This is typically the case in jets and wakes when the nonlinear region extends to large distances downstream.
Figure 4.7: Comparison of averaged streamwise perturbation velocity with experiments with and without LES.
In the Kirchhoff formulation the source terms are assumed to be distributed over a fictitious surface in the flow. The nonlinear effects (nonlinear wave propagation and steepening; variations in the local sound speed; and noise generated by shocks, vorticity, and turbulence in the flow field) happening within the Kirchhoff surface are captured by the surface integration terms, but the Kirchhoff formulation requires the integration surface to be placed in a linear flow region (i.e. far away from the body). This is difficult to achieve as most computational grids are generated with the concern of minimizing computations. Usually, a fine quality mesh is used near the body with increasing cell size towards the outer boundaries. Therefore, the quality of the solution available in the linear flow region is generally bad. The FW-H equation, on the other hand, works well even if the integration surface is in the nonlinear flow region. A detailed comparison of the Kirchhoff and FW-H formulations is provided by Brentner and Farassat [32].

The solution of the full FW-H equation requires the evaluation of two surface integrals and one volume integral. The surface integrations correspond to the “thickness” noise (monopole) and the “loading” noise (dipole). The volume integration corresponds to the quadrupole term which accounts for the nonlinearity in the flow [33, 31]. Evaluation of the volume integral can be extremely computationally intensive and difficult to implement. Fortunately, the quadrupole term can be safely ignored for most subsonic flows as is the case in the present study.

Only recently has the FW-H equation been used on a fictitious (i.e. not the same as the
body) permeable integration surface [21] - exactly like the Kirchhoff approach. di Francescantonio [21] demonstrated that when the FW-H approach is applied on a Kirchhoff-type surface, the quadrupole sources enclosed within the surface are accounted for by the surface sources. It should be noted that the “thickn ess” noise and the “loading” noise as obtained from solving the FW-H equation do not have any physical significance if the surface of integration is chosen to be permeable (fictitious). However, when the integration surface coincides with the body, these terms provide a physical insight into the source of sound generation.

The FW-H equation is written in the standard differential form as

\[ \square^2 p'(x,t) = \frac{\partial^2}{\partial x_i \partial x_j} [T_{ij}H(f)] - \frac{\partial}{\partial x_i} [L_i \delta(f)] + \frac{\partial}{\partial t} ([\rho_o U_n] \delta(f)) \]

where \( L_i \) and \( U_n \) are defined as

\[ U_n = U_i \hat{n}_i \quad U_i = (1 - \frac{\rho}{\rho_o}) v_i + \frac{\rho u_i}{\rho_o} \]

\[ L_i = P_{ij} \hat{n}_j + \rho u_i (u_n - v_n) \]

and \( T_{ij} \) is the Lighthill stress tensor. The FW-H equation can be solved using the formulation in Brentner and Farassat [32], and the solution can be written in an integral form.
The quadrupole term is ignored in the present formulation. The integrations are performed on the FW-H surface at the retarded time. Since the FW-H surface is fixed relative to the body (the cone) for this study, and the flow Mach number is constant, the following terms in the above integrals are zero: $U_n = M_r = 0$. The standard time binning technique discussed by Özyörük and Long [34] is used for obtaining pressure at the observer locations.

### 4.4.1 The FW-H code and its Validation

The FW-H code is written in Fortran 90. The code was tested for a model problem - a stationary monopole in a uniform mean flow. The FW-H surface is chosen to be a box made up of rectangular panels. The analytical solution to the model problem is evaluated at the center of each panel to obtain the time history of the primitive variables on the FW-H surface. The prediction from the FW-H code (using the analytical data on the surface as input) is then compared with the analytical pressure perturbation at a point outside the surface. Figure [4.8] compares at an arbitrary point (300 m, 0, 0) the pressure perturbation predicted by the FW-H code and that obtained analytically for a stationary monopole source
Figure 4.8: Validation of the FW-H code against the analytical solution for a stationary monopole in a uniform mean flow.

with an amplitude of 0.01 Pascals and a frequency of 2.267 Hz placed in a uniform mean flow of 0.3 Mach number. The analytical solution to this problem is:

$$
\phi(x,t) = \frac{\epsilon \exp(i \omega \tau_*)}{4\pi \left[ (x + U_0(\tau_* - t))^2 + y^2 + z^2 \right]^{1/2}} \times \frac{1}{1 + \frac{M_0(x + U_0(\tau_* - t))}{\left[ (x + U_0(\tau_* - t))^2 + y^2 + z^2 \right]^{1/2}}} \quad (4.4)
$$

where $\tau_*$ is given by

$$
\tau_* = t + \frac{M_0x - \left[ (x^2 + (1 - M_0^2)(y^2 + z^2) \right]}{c(1 - M_0^2)} \quad (4.5)
$$

The derivation of the above is available in Appendix A. The unstructured grid over
the cone is created such that there is an unstructured cylindrical surface enclosed in the computational domain (Fig. 4.1). This surface is chosen to be the permeable FW-H surface. The elements of the surface are faces of the tetrahedra, and therefore, triangles. Since these triangles are chosen from the unstructured mesh, the area and normal varies from element to element. This, however, is not a problem because the FW-H equation only requires information on a closed surface; it does not depend on the structure of the elements constituting the surface. Clustering of the surface elements is desired to increase the resolution of the sources. The FW-H surface used for the present computation is the inner cylinder in Fig. 4.1. This grid was used with the model problem of stationary monopole in a uniform mean flow to test if the unstructured grid poses any problems. A perfect match is observed between the FW-H prediction and the analytical solution (Fig. 4.9). The comparison is made at an arbitrary point (300 m, 0, 0). This confirms that an unstructured-mesh surface can be used as a FW-H surface without any loss of accuracy. Note that the first few seconds where the FW-H prediction does not match the analytical solution is the time it takes for the sound to reach the observer. This delay is more in Fig. 4.9 than in Fig. 4.8 because the unstructured FW-H surface is very small and hence, farther away from the observer point than the structured surface used for Fig. 4.8.
Figure 4.9: Comparison of the FW-H prediction using unstructured surface grid against the analytical solution.

4.4.2 Results for the Cone

PUMA is used to obtain time accurate data (the primitive flow variables) on the FW-H surface. One complete shedding cycle of the simulation is used for far-field noise prediction. Pressure at a few points outside the FW-H surface (in the near field) was collected to compare with the predictions of the FW-H code. Four points distributed in the azimuthal direction near the base of the cone and very close to but outside the FW-H surface were chosen for comparison. The coordinates of the points are tabulated in Table 4.1. The cone has a base diameter of 0.02 m and a vertex angle of 60°. The center of the base of the cone is at the origin and the vertex points upstream (positive x). The FW-H surface is a cylinder of radius 0.05 m and length 0.175 m, centered at the origin.

Figure 4.10 compares the pressure fluctuations at the four points listed in Table 4.1.
Table 4.1: Coordinates of the observer locations for comparing FW-H predictions against PUMA.

<table>
<thead>
<tr>
<th>Point No.</th>
<th>x (m)</th>
<th>y (m)</th>
<th>z (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>-0.055</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>-0.025</td>
<td>0.055</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>-0.055</td>
</tr>
</tbody>
</table>

Note that the PUMA pressure predictions have been shifted up by 20 Pascals. This is relatively a very small amount, about 0.02% of the mean pressure. We believe that this under-prediction by PUMA may be due to the dissipation caused by inadequate clustering of grid cells. It may also be due to the small sample size. An ensemble averaging over a few shedding cycles should improve the results. Note that this error is of the order of magnitude of pressure perturbations predicted by the FW-H code at any point inside the FW-H surface, which should actually be zero. However, the prediction by the FW-H code agrees very well qualitatively with the PUMA solution.

4.4.3 Sound Directivity

The directivity of the noise from the cone is obtained by calculating the root mean squared (r.m.s.) pressure perturbation for one shedding cycle at different observer locations in the azimuthal and longitudinal directions. Since the calculation for one observer location is completely independent of any other location, it is a perfect problem to run in parallel. Long and Brentner [35] suggested some self-scheduling parallel methods for multiple serial
Figure 4.10: Comparison of pressure fluctuation, $p - p_{\infty}$ as predicted by PUMA and FW-H code at various locations listed in Table 4.1.
Figure 4.11: Directivity of the noise in the azimuthal direction behind the base of the cone ($x = -0.1$).

codes. A similar approach is used here to parallelise the FW-H code to obtain the directivity patterns. The MPI standard library is used to make the code parallel. The parallel algorithm used is described in Appendix B.

Figure 4.11 plots the directivity pattern in the azimuthal direction on the plane $x = -0.1$ m, which is right behind the base of the cone. The pattern in Fig. 4.11 is symmetric because of the symmetry of the cone about its axis. Since the FW-H equation cannot predict the pressure fluctuation inside the FW-H surface, we can compute the noise only outside the FW-H surface. Therefore, the directivity patterns are plotted in an annular region outside the FW-H surface.
Figure 4.12: Directivity of the noise in the longitudinal direction in the plane $z = 0$.

Figure 4.12 plots the directivity pattern in the longitudinal direction on the $z = 0$ plane. Since the noise is caused by both turbulence and fluctuating surface forces, the directivity shows several lobes.

A conventional polar directivity pattern in the longitudinal direction ($z = 0$ plane) is plotted in Fig. 4.13 for observers at 10 different radial locations ($r = 0.15 - 0.24$ m). In Fig. 4.13, the cone is pointing to the right; the radial distance from the origin is equal to the r.m.s. pressure and the angle (theta) illustrates the location of the observer point in the domain.
Figure 4.13: Polar plot of sound directivity in $z = 0$ plane at a few radial locations.

4.5 Conclusions

Aerodynamic noise from a cone has been studied as a model problem to test the possibility of using unstructured grids for noise prediction from complicated bodies such as landing gears. A finite volume flow solver, PUMA has been used to obtain time-accurate flow data on a permeable FW-H surface. The FW-H code was validated against a model problem of a monopole in a uniform mean flow. The predictions from the FW-H code have been compared at four observer locations in the near field with direct calculations from PUMA. Noise predictions are made for a period of one shedding cycle. The comparison is fairly accurate with only a small D.C shift error. The directivity patterns of the noise from the cone are plotted in azimuthal and longitudinal directions. The sound directivity pattern has been shown to be fairly complicated due to the complex physics inside the FW-H surface.
Chapter 5

Future Work

For the ship airwake problem, longer time accurate runs on finer grids are needed to accurately predict the Strouhal number in the wake and capture the separation bubble. This requires tremendous supercomputing power. Another possible extension of the work is to use the steady state data for different yaw angles and compute the SHOLs for some helicopters. A more involved analysis could incorporate the helicopter in the CFD simulation itself and study the combined effect. This may be useful because the presence of a powerful rotor can significantly alter the flowfield. This can be done in two ways - (1) simulating the effect of a rotor by inserting fictitious velocity sources, or (2) exactly simulating the physical situation by using moving grids for the rotor blades.

The airwake data collected from these simulations can be used for rotor start-up and shut-down simulations which can predict the deflections of the rotor blades. Such simulations can help identify the critical conditions when landing and take-off operations should not be performed. The airwake simulations can also be incorporated in a flight simulator to train pilots.

One of the major problems with using unstructured grids and explicit time marching scheme is that one loses all control over the time step. The time step is governed by the
smallest cell size in the computational grid which can be extremely small. This drastically increases the total computation time. This problem can be avoided by either using structured grids or by using an implicit time marching scheme. Structured grids are extremely difficult to create over complicated geometries but this difficulty can be avoided if a “stair-step” grid is used. A “stair-step” grid however does not represent the geometry exactly and this may not be acceptable in all cases. The use of implicit time marching scheme looks attractive but it wasn’t tried because of unavailability of any such software.

The aeroacoustics problem treated in this thesis is a benchmark problem for the FW-H code and the concept of using unstructured grids for aeroacoustic applications. The plan is to use this code to predict aerodynamic noise from complicated geometries such as a landing gear. My colleague, Mr Souliez is already running a time accurate case of flow over a landing gear geometry [36]. The data from this run will be used to predict far-field aerodynamic noise and directivity patterns. Since the FW-H code is independent the source of the time-accurate data, a comparison can be made between the data from a structured grid and from an unstructured grid. This would consolidate the approach of using unstructured grids for aeroacoustic applications.
Bibliography


Appendix A

Velocity Potential from a Monopole in a Uniform Mean Flow

Consider a stationary point source of mass addition (a monopole), \( Q(x,t) \) at \( x = \vec{0} \) of the form

\[
Q(x,t) = A \exp(i\omega t) \delta(x - \vec{0}) \tag{A.1}
\]

The linearised continuity equation is

\[
\frac{\partial \rho'}{\partial t} + u_0 \cdot \nabla \rho' + \rho_o \nabla u' = Q(x,t) \tag{A.2}
\]

and the linearised momentum equation is

\[
\rho_o \left( \frac{\partial u'}{\partial t} + u_o \cdot \nabla u' \right) = -\nabla p' \tag{A.3}
\]

where the primed variables are perturbation quantities and the subscript \((o)\) denotes mean flow quantities. Using \( u' = \nabla \phi \) (irrotational flow assumed), Eq. \([A.3]\) implies

\[
\rho_o \frac{D_0 \phi}{D_0 t} = -p' + \text{constant} \tag{A.4}
\]
Differentiate Eq. A.4 w.r.t time \( \left( \frac{D_o}{D_o t} \right) \) and use the relation \( p' = c^2 \rho' \) with Eq. A.2 to obtain

\[
\rho_o \frac{D^2_o \phi}{D_o t^2} = -c^2 \left[ Q(x,t) - \rho_o \nabla^2 \phi \right]
\] (A.5)

This gives the wave equation in \( \phi \) for a monopole source in a uniform mean flow.

\[
\frac{D^2_o \phi}{D_o t^2} - c^2 \nabla^2 \phi = -\frac{c^2}{\rho_0} Q(x,t)
\] (A.6)

Eq. A.6 can be solved using Green’s Functions for \( \phi(x,t) \).

### A.1 Particular case : \( U_0 = U_0 \hat{i} \)

Using the source definition in Eq. A.1, the wave equation (Eq. A.6) is written as

\[
\frac{1}{c^2} \left[ \frac{\partial}{\partial t} + U_0 \frac{\partial}{\partial x} \right]^2 \phi - \nabla^2 \phi = \varepsilon \exp(i\omega t) \delta(x - \bar{0})
\] (A.7)

The source is located at \( x = 0 \) and is stationary. The amplitudes \( \varepsilon \) and \( A \), are related as, \( \varepsilon = -(c^2/\rho_0)A \). We define a new reference frame, \( R^* \) moving with \( U_0 \) such that

\[
x^* = x - U_0 t, \quad y^* = y, \quad z^* = z, \quad t^* = t
\] (A.8)
The wave equation in the moving frame, $\mathcal{R}^*$ becomes

$$\frac{1}{c^2} \frac{\partial \phi}{\partial t^*} - \nabla^2 \phi = \epsilon \exp(i\omega t^*) \delta(x^* + U_0 t^* - \bar{0})$$  \hspace{1cm} (A.9)$$

Using Green’s Functions, the velocity potential, $\phi$ can be obtained as

$$\phi(x^*, t^*) = \int_{\Omega_T} \frac{\delta(t - \tau - \|x^* - \bar{x} + U_0 \tau\|/c)}{4\pi\|x^* - \bar{x} + U_0 \tau\|} \times \epsilon \exp(i\omega \tau) \delta(\xi_1) \delta(\xi_2) \delta(\xi_3) \, d^3\xi \, d\tau$$  \hspace{1cm} (A.10)$$

After integrating w.r.t. space variables, $\phi$ can be expressed as

$$\phi(x^*, t^*) = \int_T \frac{\delta(t - \tau - \|x^* - \bar{x} + U_0 \tau\|/c)}{4\pi\|x^* + U_0 \tau\|} \times \epsilon \exp(i\omega \tau) \, d\tau$$  \hspace{1cm} (A.11)$$

The following property of the Dirac delta function, $\delta$ is used to evaluate the integral in Eq. [A.11]

$$\int_{-\infty}^{+\infty} f(\theta) \delta[g(\theta)] \, d\theta = \left[ \frac{f(\theta)}{\|dg/d\theta\|} \right]_{\theta=\theta_*} g(\theta_*) = 0$$  \hspace{1cm} (A.12)$$

where $\theta_*$ are the roots of $g(\theta) = 0$. For the present case, $g = t - \tau - \|x^* + U_0 \tau\|/c$, and $\theta$ is $t$. $\tau_*$ is obtained as

$$\tau_* = t^* + \frac{M_0 x^*}{c} - \frac{1}{c} \left[ (x^* + U_0 t^*)^2 + (1 - M_0^2)(y^2 + z^2) \right] \frac{1 - M_0^2}{1 - M_0^2}$$  \hspace{1cm} (A.13)$$
\[ \frac{\partial g}{\partial \tau} \text{is solved at } \tau = \tau^* \text{ to give} \]

\[ \left. \frac{\partial g}{\partial \tau} \right|_{\tau = \tau^*} = -1 - M_0 \frac{x^* + U_0 \tau^*}{[(x^* + U_0 \tau^*)^2 + y^2 + z^2]^{1/2}} \]

(A.14)

The velocity potential, \( \phi \) in the moving frame, \( \mathcal{R}^* \) can now be written as

\[ \phi(x^*, t^*) = \frac{\varepsilon \exp(i\omega \tau^*)}{4\pi [(x^* + U_0 \tau^*)^2 + y^2 + z^2]^{1/2}} \times \frac{1}{1 + \frac{M_0(x^* + U_0 \tau^*)}{[(x^* + U_0 \tau^*)^2 + y^2 + z^2]^{1/2}}} \]

(A.15)

Writing Eq. (A.15) in the stationary frame coordinates \((x, y, z, t)\)

\[ \phi(x, t) = \frac{\varepsilon \exp(i\omega \tau^*)}{4\pi [(x + U_0(\tau^* - t))^2 + y^2 + z^2]^{1/2}} \times \frac{1}{1 + \frac{\frac{M_0(x + U_0(\tau^* - t))}{[(x + U_0(\tau^* - t))^2 + y^2 + z^2]^{1/2}}} \]

(A.16)

where \( \tau^* \) is given by

\[ \tau^* = t + \frac{M_0 x - [x^2 + (1 - M_0^2)(y^2 + z^2)]}{c(1 - M_0^2)} \]

(A.17)
Appendix B

Parallel Algorithm to Obtain Directivity Patterns

A Single Program Multiple Data (SPMD) approach is used to make the FW-H code parallel. The code can be run on any number of processors for any number of observer locations. The idea is to automatically schedule the processors to choose observer locations and obtain the r.m.s. pressure there. The easiest way to implement this is to predetermine which processor works on which observer locations. Suppose we work with N observer locations and M processors (assume N is greater than M). The first processor will work on observer locations with serial numbers 1, M+1, 2M+1 and so on. The second processor will work on 2, M+2, 2M+2 and so on. If M does not divide N exactly then the remaining observer locations are given to the first Q processors, where Q is the remainder of N/M. The approach is explained in the algorithm in Fig. B1.

Since the calculation of one observer location is completely independent of any other observer location, this parallel implementation scales almost perfectly. The performance of this code was tested on four different parallel machines - COCOA, COCOA2, Lionx and SGI Origin 2000. The description of the first three machines is available in Chapter 2. The SGI Origin 2000 is available at the National Center for Supercomputing Applications
Figure B.1: Algorithm to run multiple serial jobs in parallel (NCSA) at the University of Illinois at Urbana-Champaign. The performance measurements are plotted in Figs. B2 and B3. The low Megaflop rate is due to the large file input/output (File I/O) operations that this code performs.
Figure B.2: Megaflops versus number of processors

Figure B.3: Total time versus number of processors