NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was scanned as received.

ii

This reproduction is the best copy available.

UMI®

3D Parallel Computations of Turbofan Noise Propagation Using a Spectral Element Method

Farzad Taghaddosi

Department of Mechanical Engineering

McGill University

Montréal, Québec

January 2006

A Thesis Submitted to McGill University

in Partial Fulfillment of the Requirements of the Degree of

Doctor of Philosophy

Your file Votre référence
ISBN: 978-0-494-27845-1
Our file Notre référence
ISBN: 978-0-494-27845-1

Dedication

TO MY BELOVED PARENTS,

♡ AHMAD & BEHJAT ♡

FOR THEIR ENDLESS LOVE

AND

COUNTLESS SACRIFICES.

Acknowledgements

I would first like to express my gratitude to my thesis advisor, Professor Wagdi G. Habashi, for his help, support, and encouragement throughout my studies. It has been a valuable experience for me to have the opportunity to work with him on cutting-edge research at the CFD Laboratory. I am also grateful to him for providing the working environment and the computing facilities that made the large-scale computations in this work possible.

My appreciation also extends to Dr. Grant G. Guevrèmont, formerly an adjunct assistant professor at the Mechanical Engineering department, for acting as a co-supervisor in initial stages of this research and for his help and guidance.

My colleagues at the CFD Laboratory of McGill University (both former and present) deserve many thanks, especially Drs. Lakhdar Remaki and Claude Lepage for sharing their opinion and offering valuable and constructive suggestions on many subjects related to my thesis, and Dr. Amik St-Cyr for his help during the initial stages of code parallelization. I also wish to thank Prof. Abdelkader Baggag from the CLUMEQ Supercomputing Center for sharing insightful discussions about preconditioning methods.

My special thanks are due to Mr. Yves Simard, the very dynamic and helpful System Administrator of the CFD Lab., and Mr. Patrice Hamelin, the System Administrator for the CLUMEQ Supercomputing Center for providing technical computer support.

I also wish to thank Pratt & Whitney Canada for funding the very early stages of this research and the Natural Sciences and Engineering Research Council of Canada (NSERC) for their financial support in the remaining years. Their assistance is greatly appreciated.

Abstract

A three-dimensional code has been developed for the simulation of tone noise generated by turbofan engine inlets using computational aeroacoustics. The governing equations are the linearized Euler equations, which are further simplified to a set of equations in terms of acoustic potential, using the irrotational flow assumption, and subsequently solved in the frequency domain.

Due to the special nature of acoustic wave propagation, the spatial discretization is performed using a spectral element method, where a tensor product of the n^{th}-degree polynomials based on Chebyshev orthogonal functions is used to approximate variations within hexahedral elements. Non-reflecting boundary conditions are imposed at the far-field using a damping layer concept. This is done by augmenting the continuity equation with an additional term without modifying the governing equations as in PML methods.

Solution of the linear system of equations for the acoustic problem is based on the Schur complement method, which is a nonoverlapping domain decomposition technique. The Schur matrix is first solved using a matrix-free iterative method, whose convergence is accelerated with a novel local preconditioner. The solution in the entire domain is then obtained by finding solutions in smaller subdomains.

The 3D code also contains a mean flow solver based on the full potential equation in order to take into account the effects of flow variations around the nacelle on the scattering of the radiated sound field.

All aspects of numerical simulations, including building and assembling the coefficient matrices, implementation of the Schur complement method, and solution of the system of equations for both the acoustic and mean flow problems are performed on multiprocessors in parallel using the resources of the CLUMEQ Supercomputer Center. A large number of test cases are presented, ranging in size from 100 000-2 000 000 unknowns for which, depending on the size of the problem, between 8-48 CPU's are used.

The developed code is demonstrated to be robust and efficient in simulating acoustic propagation for a large number of problems, with an excellent parallel performance.

Résumé

Un code 3D pour la simulation du bruit généré par les entrées d'un réacteur à double flux a été développé. La méthodologie est basée sur un calcul aéro-acoustique où les équations d'Euler linéarisées sont résolues sur le domaine entier. Le modèle d'équations est simplifié à une équation à une variable, soit le potentiel acoustique, utilisant l'hypothèse d'un écoulement irrotationnel, squi par la suite est résolu dans le domaine fréquentiel.

Dû à la nature particulière de la propagation des ondes acoustiques, la discrétisation spatiale est réalisée par la méthode des éléments spectraux, où un produit tensoriel de polynômes de degré 'n' basés sur les fonction orthogonales de Chebyshev est utilisé pour l'approximation des variations à l'intérieur d'éléments héxahedraux. Pour le champ lointain, des conditions non réfléchissantes basées sur le concept d'une couche absorbante sont imposées. Ceci est réalisé en ajoutant un terme à l'équation de continuité, sans pour autant modifier le model physique.

La solution du système linéaire du problème acoustique est basée sur la méthode du complément de Schur, qui est une technique de décomposition de domaine sans recouvrement. La matrice de Schur est résolue en premier en utilisant une méthode itérative sans stokage de matrice et dont la convergence est accélérée par un nouveau pré-conditionneur local. La solution globale est ensuite obtenue en récoltant les différentes solutions obtenues sur les différents sous-domaines.

Le code 3D résoud également l'écoulement moyen par voie de l'équation du potentiel afin de tenir compte des effets de variation de l'écoulement autour de la nacelle sur la diffusion acoustique des champs de radiation sonore.

Tous les aspects de la résolution numérique comprenant la construction et l'assemblage des coefficients de la matrice, l'implémentation de la méthode du complément de Schur, ainsi que la résolution du système d'équations pour l'acoustique et l'écoulement moyen sont réalisés sur multi-processeurs et en parallèle. Un grand nombre de cas testes sont présentés dont les tailles varient entre 100 000 à 2 000 000 d'inconnues et pour lesquels entre 8 et 48 CPUs sont utilisés sur le superordinateur du CLUMEQ.

L'approche développée a dèmontré une grande robustesse et efficacité pour la simulation de la propagation acoustique pour un nombre important et varié de problèmes, avec une excellente performance de la parallélisation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Historical Background

Since the celebrated treatise of Lord Rayleigh more than a century ago [80], the subject of acoustics has expanded from its original context as a phenomenon closely related to fluid mechanics, to cover a wide range of topics from speech communication and phonetics to ultrasonic, signal processing, and so on.

The advancement of aeronautical and aerospace sciences during the early decades of the twentieth century, however, renewed interest in the subject of acoustics as a branch of fluid mechanics, where problems such as exhaust jet noise of newly developed engines had to be addressed.

In the early 1950's, a new theory emerged that revolutionized the field and brought new hope in finding solutions to very complicated problems. Introduced by Sir James Lighthill [60], the essence of the "*Acoustic Analogy*" theory is: "if the real fluid, in which highly nonlinear motions may occur, is replaced by a fictitious acoustic medium, in which only small amplitude linear motions occur, and if the medium is acted upon by an external stress system $T_{ij}(\vec{x}, t)$, then exactly the same density fluctuations will be produced."

That is, instead of solving the fully nonlinear equations, one can solve the linearized

but inhomogeneous wave equation

$$\left(\frac{\partial^2}{\partial t^2} + c_0^2 \nabla^2\right)\rho = \frac{\partial^2 T_{ij}}{\partial x_i \partial x_j}, \qquad T_{ij} = \rho u_i u_j + p_{ij} - c_0^2 \rho \, \delta_{ij}, \tag{1.1}$$

where ρ is the density and T_{ij} is the Lighthill stress tensor. A direct byproduct of Lighthill's theory is the famous 'U^8-law' for acoustic power of exhaust jets, derived by a relatively simple order of magnitude analysis based on far-field integral solution of equation (1.1), which found widespread use in jet noise prediction.

Although Lighthill's inhomogeneous wave equation is only valid for sound fields generated by stationary turbulent eddies, the concept behind this theory can be used to derive equations for other situations of interest. This fact is what makes Lighthill's acoustic analogy theory so significant.

A powerful example is the case of noise generated by moving rigid bodies, such as propellers and helicopter rotors. Unsteadiness created by the motion of rigid bodies contributes to the sound field. This contribution can be represented by adding extra terms to the right-hand side of equation (1.1) in the form of monopoles and dipoles. A general theory consisting of the case of moving solid bodies in arbitrary motion is presented by Ffowcs Williams and Hawkings, commonly known as the FW-H equation, which is the fundamental equation used in acoustic analysis of propellers and rotors [26]. Several integral solutions of the FW-H equation in the time domain, developed by Farassat [23], are currently the predominant formulations used in propeller/rotor noise analysis.

Aeroacoustics, a name initially coined by Lighthill, which studies the noise generated by aerodynamic forces, encompasses several fields. In addition to jet and propeller/rotor noise, study of sound field generated by rotors/fans of turbofan/turbojet engines has been another significant area of research since the 1960's. In the early years of jet-powered flight, turbojet engines were the dominant form of power plant used for aircraft. The main source of noise for this type of engine was the turbulence associated with the exhaust jet of hot gases, which creates significantly higher noise levels compared to forward-propagating rotor noise.

The advent of turbofan engines, however, created a shift in the way in which engine

noise was studied. This was mainly due to the partial unmasking of rotor/fan noise as a direct consequence of reduced jet noise caused by mixing of exhaust jet with the cooler bypass stream. As a result, studies of fan noise gained momentum, with research focusing on understanding the generating mechanisms and on methods for its prediction and analysis.

1.2 Fan Noise Problem

Research shows that as the bypass ratio of a turbofan engine increases beyond 10, fan noise dominates the engine noise signature, exceeding that generated by the exhaust jet [64]. This is more pronounced during take-off and landing, as depicted in Figure 1.1 [73].

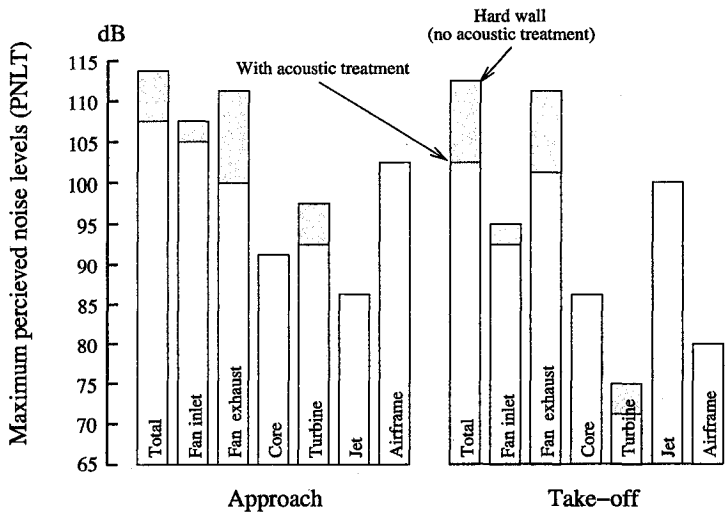

Figure 1.1: Flyover noise levels for a typical turbofan engine on a component basis.

At subsonic speeds, fan noise is characterized by tone noise (blade-passage frequency (BPF) and its harmonics), superimposed over broadband noise (which is mainly due to turbulence in the atmosphere and distortions in the flow as it enters the compressor inlet).

As the rotor tip speed approaches supersonic values, shocks will be formed on the blades and the fan noise will be dominated by multiple pure tone (MPT) noise, also known as buzz-saw noise [39, 70]. As the name implies, MPT noise is characterized by spikes

across all frequencies within its spectrum. It starts at speeds where an inlet shock forms upstream of the rotor, and ends when the shock completely moves inside the passage between the blades, where pressure disturbances can no longer propagate upstream.

For tonal noise, however, the generating mechanism is very different. As the blade rotates, its periodic velocity field induces pressure fluctuations on stator blades, and vice versa. In addition, the impingement of wakes emanating from trailing edges of rotor blades onto stator blades creates a similar effect. So, the sound generated is caused by the rotor-stator interaction propagating mostly upstream of the rotor and eventually into the far-field.

The main goal of fan noise prediction is therefore to be able to accurately predict the sound field as it propagates away from the engine, taking into account inlet geometry variations and changes in the flow field. The alternate goal is to be able to predict changes in the sound field as a function of geometry and flow field variations. Materialization of these goals, even partly, would translate into reduced noise emissions.

Given the noise generation mechanism, one can reduce the noise by adjusting rotor-stator blade count to achieve cut-off of tone noise due to impingement or rotor wakes, increase rotor-stator spacing to weaken the wakes, improve inlet design to minimize flow distortion which contributes to broadband noise, shield downward propagation of sound by using scarf inlets, and use blowing at the trailing edge of rotor blades to weaken the wakes [21]. Aside from suppressing fan noise by source control, one can also use acoustic liners in the duct, or employ active noise control systems by installing actuator rings in the inlet/exhaust duct or even on the rotor blades itself [95]. Several review articles summarize the status of fan noise prediction and the challenges ahead [21, 22, 44].

The research on noise simulation is mostly motivated by the desire to reduce noise pollution around the airports and to improve passenger comfort during flight. This is further driven by stricter noise standards put in place, such as ICAO Chapter 4, which came into effect on January 1, 2006. Also, there is the element of business competitiveness that motivates engine manufacturers to build quieter engines.

Based on a program initiated by NASA in 1992 in partnership with the Federal Aviation

Administration (FAA) and U.S. aerospace industries, a goal was set to develop technology to reduce aircraft noise by 10dB by 2007 and by 20dB by 2022, relative to 1992 levels. This means that fan noise reduction promises to remain an important component of overall engine noise concerns for the years to come.

1.3 Simulation of Fan Noise: Methods and Approaches

The problem of fan noise is comprised of three main components: 1) sound generation at the rotor/fan, 2) propagation of sound in the inlet duct, and 3) its radiation into the far-field. Problem of sound generation is usually considered independent of the latter two factors and is studied separately. Based on the theory of Tyler and Sofrin [105], pressure field generated due to rotor-stator interaction is a superposition of an infinite number of m-lobe pressure patterns, where $m = nB \pm kV$, each rotating at the speed $nB\Omega/m$. Here, n are the harmonics of the blade-passage frequency, B and V are the number of rotor and stator blades, respectively, k is a positive integer, and Ω is the rotor's angular speed. Since its inception, this theory has been widely used to characterize noise generation at the source.

Sound propagation in the duct and its radiation into the far-field, however, pose a bigger challenge. In the compressor inlet area, the cross-sectional area is generally not constant but varies along the duct. As a result, the assumption of a perfectly cylindrical shape (axisymmetrical flow) is a simplification that could be unrealistic. A bigger challenge is due to the fact that perturbations leaving the duct are radiated into an infinite domain. Obviously, for computational purposes a truncated domain has to be used, which brings up issues such as nonreflecting boundary conditions. Moreover, unlike many purely fluid mechanics problems, acoustic problems require an accurate solution even in the regions far from the source. This means a coarse grid, which will reduce computational load, can not be used in the far-field region.

These computational complexities have led researchers to adopt different approaches in analyzing fan noise problem. In general, these methods can be divided into two categories:

Full Flow-field CAA Analysis

Here, the computational domain covers the fan face, inlet duct, and a truncated region in front of the duct's exit, representing the far-field. Noise generation at the rotor is modeled using either the Tyler & Sofrin theory [105], or by directly calculating pressure variations using first principles, i.e., by setting up the computational domain to include both rotor and stator, and then solving time-accurate Navier-Stokes equations with proper turbulence models. The work of Rumsey *et al.* [85] is among the first in this field, followed by Biedron [7], who used the same approach as in [85], but studied higher harmonics of BPF. Tsuchiya *et al.* [104], using unsteady CFD, performed similar analysis to investigate the characteristics of fan tone noise source.

In any CFD analysis, the very fact that the governing equations are replaced by their discrete counterparts naturally introduces some errors into the calculations. The issue of accuracy becomes even more important for aeroacoustic problems since the amplitude and energy levels of acoustic variables are generally several orders of magnitude smaller than those of the flow variables [66]. This large disparity imposes stringent requirements on the accuracy of the discretization scheme. This is further complicated by the fact that fan noise problems are usually associated with very high frequencies, i.e., more grid points are needed for accurate resolution of the waves. Therefore, the use of high-order schemes for aeroacoustic problems becomes an absolute necessity.

In the finite difference context, Lele [55] presented a series of schemes in which, instead of simply expanding the stencil, coefficients of the difference equations were chosen such that the scheme would represent more accurately wave components over a wide range of wave numbers. The result was a set of schemes whose discrete dispersion relation very accurately followed the dispersion relation of the acoustic waves. The dispersion relation, of the form $\omega = W(\vec{k})$, contains all the propagation and dispersion characteristics of the associated wave phenomenon.

Comparable to the compact schemes of Lele in terms of computational cost and accuracy, the Dispersion-Relation-Preserving (DRP) scheme of Tam and Webb [99] is another

widely used high-order method. This method, however, has the added advantage of preserving dispersion-relation and therefore guaranteeing that waves computed by this method will all propagate with the same speed. The work of Lan *et al.* [54] using linearized Euler equations in frequency domain, and Özyörük and Long [75, 76] using the Euler equations in the time-domain are recent examples of the use of DRP schemes in the simulation of turbofan radiated noise. As pointed out in [99], DPR schemes can provide accurate results when at least 4 to 5 mesh points are used to cover a wavelength. This is already a very significant improvement over classical second-order finite difference methods, where the required number of mesh points per wavelength for adequate accuracy is around 18–24.

In the community of finite element users, research on fan noise is mostly due to Eversman and coworkers [19, 20, 62, 84], who tackled 2D and axisymmetrical geometries. Their approach was based on solving the convected wave equation in terms of velocity potential, using quadratic or cubic finite elements in the near-field and wave envelope/infinite elements in the far-field. Wave envelopes are elements whose shape functions mimic the decaying behavior of outgoing waves of simple sources [2, 38]. Although Eversman's method gives overall good results for the sound pressure amplitudes in the far-field, it falls short of providing accurate phase information in the far-field unless a very fine grid is used

As it has been noted so far, any numerical scheme used for simulation of aeroacoustic problems should be nondispersive. An added advantage would be a scheme that would require a smaller number of grid points per wavelength for accurate resolution of the waves. These two very important properties are embedded in the spectral methods [36]. Traditionally, the use of spectral methods was limited to flows in simple domains with simple boundary conditions. This, however, changed with the introduction of spectral multidomain method [50] and spectral element method [77], allowing problems in more complex geometries to be tackled.

In spectral multidomain methods, as the name implies, the computational domain is divided into smaller subdomains where the flow variables are discretized using high-order orthogonal polynomials. Appropriate patching techniques are then used to ensure conti-

nuity of flow variables and their derivatives across the interface boundaries of subdomains [51], with ideas similar to those used in the finite volume methods. Stanescu *et al.* [91] solved unsteady nonlinear Euler equations using this method to simulate noise radiation from ducted fans, including both axisymmetric and fully three-dimensional geomtries.

The spectral element method (SEM) [77], however, is conceptually very similar to the finite element method. The difference is that instead of classical Lagrangian-based polynomials, spectral approximations based on orthogonal functions are used within each element. Recently, Stanescu *et al.* [93] used this method for the solution of fan noise problem for axisymmetric nacelles. Their approach was based on the weak formulation of the wave equation written in terms of velocity potential and solved in the frequency domain. More recently, Xu *et al.* [108] applied a similar technique to obtain far-field radiation patterns of a generic aft-mounted nacelle-body configuration.

Hybrid Methods

In a typical fan noise problem, the flow at the near-field, where the sound is generated, is highly nonlinear. These nonlinearities, however, diminish as the sound propagates along the duct and away from the source and radiates into the far-field. So, it seems natural to divide the computational domain into near-field, where nonlinear equations such as the Navier-Stokes ought to be used, and far-field, where linear acoustic theory could be applied. This segregation of computational domain and the use of two or more different forms of analysis in each subdomain is the essence of hybrid methods.

While the choice of the method for the near-field (encompassing the acoustic source and duct interior) is usually limited to the full set of Navier-Stokes equations, a variety of methods could be used for the far-field analysis. Acoustic analogy is one such approach, where the governing Navier-Stokes equations are rearranged in the form of an inhomogeneous wave equation, with the far-field sound pressure given in terms of surface and volume integrals using Green's function formulation [8, 40].

Assuming that the flow at the nonlinear near-field is well resolved, a linear Kirchhoff

formulation evaluated on an arbitrary control surface surrounding the nonlinear field and sound sources can also be used to calculate far-field radiation. Since, only surface integrals and first derivatives are included in the Kirchhoff formulation, it offers an advantage over the acoustic analogy approach, which may require evaluation of volume integrals.

An early application of the use of integral methods is due to Horowitz *et al.* [42]. To match the solutions between the near- and far-fields, they used an iterative technique so that a continuous description of the acoustic field could be obtained. More recent applications include the work of Lidoine *et al.* [59], Özyörük [74], and the hybrid method of Zhang *et al.* [110].

The classical Kirchhoff formulation is limited to a stationary surface. Farassat and Myers [24] derived and extended the formulation for a moving, deformable surface. This has found applications in propeller/rotor noise prediction [30]. In hybrid methods, the far-field can also be modeled using the linearized Euler equations. However, to ensure accurate solutions in the far-field, issues such as mesh spacing and dissipation and dispersion errors must be properly addressed [56, 101].

1.4 Proposed Approach

The work presented in this thesis is based on a full flow-field CAA analysis. It is in fact an extension of the work of Stanescu *et al.* [93] to three dimensions with several modifications [98]. The main objective of this work is to present an efficient yet cost-effective approach for simulation of radiated fan noise.

The governing equations are based on the linearized Euler equations, with the assumption of a mean flow in the x-direction, written in terms of the velocity potential. The assumption of isentropic flow, which allows the use of a velocity potential is valid if there are no significant nonlinearities near the duct inlet, such as those due to flow distortions. For the purpose of determining the forward-radiating acoustic field during take-off and landing, this assumption is reasonable. Also, using a velocity potential has the added advantage that

it reduces the number of dependent variables to only one. This translates into lower computational cost in terms of both memory usage and CPU time, which could be significant given the fact that three-dimensional problems are being solved.

To ensure adequate resolution of acoustic waves and to minimize the point-per-wavelength requirement, a spectral approximation is used in the context of the spectral element method by extending the 2D formulation of Patera [77] to three dimensions.

Using a time-harmonic assumption, the governing equations are solved in the frequency domain. This approach requires that a system of equations to be solved in order to obtain the solution. The alternative is a time-marching (explicit) method, which requires much less memory but the computation time will be comparable since time-steps are restricted by the very small grid size of the spectral mesh near the boundaries. Moreover, a low-dispersion scheme has to be used for time integration, which is more expensive than ordinary time marching techniques.

Due to the large size of typical three-dimensional problems, a parallel algorithm based on a nonoverlapping domain decomposition method is used to solve the system of linear equations. The algorithm is based on the Schur complement method. It reduces the solution of the original set of equations to a system (Schur matrix) that is at least one order of magnitude smaller and therefore much easier to solve. The parallel algorithm is also very suitable for distributed memory machines, allowing the use of a cluster or a network of computers for the solution of very large problems. Parallelization has been implemented using the MPI (Message Passing interface) protocol, with the extensive use of the PETSc libraries [4]. Also, there exists the option of using a parallel direct solver [1], which is only suitable for small problems (with 10^5 dof or less) due to memory restrictions.

At the noise source, i.e., fan face, the wave equation reduces to an eigenvalue problem in terms of the duct acoustic modes. Solution of this problem gives acoustic perturbations at the inlet in the form of m-lobe spinning acoustic potential patterns, in accordance with the Tyler & Sofrin theory [105]. Given the number of radial and circumferential (spinning) modes, boundary conditions at the fan face are then prescribed in terms of acoustic

potential. Cases of circular ducts (with or without a centerbody), rectangular sources, and semi-infinite parallel plates are considered.

The boundary conditions at the far-field are imposed using a buffer zone near the exit boundary in which outgoing waves are exponentially damped [28, 53]. This is achieved by adding a term to the governing equations to ensure that outdoing waves are not reflected back into the domain.

The effect of the mean flow on acoustic scattering is also taken into account by solving the convected wave equation. This is accomplished by first solving the mean flow problem separately on the same (spectral) grid. Once the mean flow field is known, the acoustic problem is solved taking into account the variations of mean flow properties. Solution of the mean flow problem is based on the full potential equation. Therefore, the results will be accurate for subsonic flows where nonlinearities are negligible and the assumption of irrotational flow remains valid. This is a reasonable assumption during take-off and landing, where Mach numbers are within the subsonic range and flow distortions near the inlet are not significant.

1.5 Thesis Originality and Contribution

The work presented addresses a difficult engineering problem in the area of computational aeroacoustics, i.e., simulation of noise propagation from turbofan engines. Currently, most industrial analyses are based on 2D simulation software which are only applicable to realtively simple or axisymmetric geometries. Thus, lack of a modern simulation tool that could handle realistic geometries within a reasonable execution time is a handicap. This research is, therefore, aimed at filling this gap by presenting an original piece of work, reflected in the objectives outlined below:

- Migrating from 2D to 3D technology,
- Achieving practical solution times: hours instead of weeks,

- Finding solution for complex geometries with industrial application,
- Creating an accurate and reliable design tool, and
- Exploiting massively parallel computers.

The multi-disciplinary nature of the research encompasses the three areas of Engineering Science, Mathematics, and Computer Science. The work presented addresses scientific issues within each discipline and provides a practical solution. Thus, the contributions are:

- Engineering Contribution:
  - Development of a 3D aeroacoustics code
  - Ability to simulate turbofan noise and other ducted acoustic problems
  - Accounting for mean flow effects
  - Incorporating symmetry formulation to reduce computational cost
  - Rapid solution times through extensive use of parallel computations
- Mathematical Contribution:
  - Delivering accurate results through the use of spectral methods
  - Using an iterative solution technique based on the Schur complement method
  - Introducing a novel preconditioner to reduce computation time
- Computational Contribution:
  - Adopting algorithms and memory management techniques to achieve an excellent parallel efficiency

1.6 Thesis Outline

After a comprehensive review of fan noise problem in this Chapter and discussion of different solution methods and literature review, we proceed to Chapter 2, where we present a detailed analysis of the acoustic problem by first deriving the governing equations, followed by the formulation of the spectral element method in three dimensions and treatment of the boundary conditions. Solution of the mean flow problem, including derivation of the weak form, discretization of equations and linearization are then followed. The chapter concludes with the mathematical formulation for implementation of symmteric boundary conditions and with details regarding generation of the spectral grid.

A brief discussion about parallelization, overlapping and nonoverlapping domain decomposition methods, and issues related to parallel efficiency and presented in Chapter 3.

Chapter 4 is dedicated entirely to the subject of the solution method. Overlapping and nonoverlapping methods used for solution of the mean flow and acoustic problems, respectively, are discussed in detail and the algorithms are presented. The chapter also includes several sections covering the important subject of preconditioning for the Schur matrix. Different preconditioners are discussed and a new preconditioner is proposed. The efficiency and suitability of the proposed preconditioner is then demonstrated using numerical examples and eigenvalue analysis.

Chapter 5 contains a variety of numerical tests to demonstrate and validate the numerical method presented in this thesis for solution of a range of acoustic radiation problems. All tests involve three-dimensional geometries and the results are compared to the analytical or other numerical data. Problems varying in size from 100 000 to 2 million unknowns are solved on multi-processors using 8-48 CPU's.

Finally, a summary is presented in Chapter 6 and several recommendations are made regarding the direction of the future research.

Chapter 2

Mathematical Formulation

As mentioned earlier in Chapter 1, the mathematical formulation used for simulating the fan noise problem is based on the linearized form of the Euler equations, which are further simplified by the introduction of velocity potential using an irrotational flow assumption. The Governing equations are non-dimensionalized using the freestream density ρ_∞, speed of sound c_∞, and characteristic length of the engine inlet (usually fan radius) R, as the reference variables. Therefore, non-dimensional variables, denoted by "$*$", are

$$p^* = \frac{p}{\rho_\infty c_\infty^2}, \qquad \rho^* = \frac{\rho}{\rho_\infty}, \qquad l^* = \frac{l}{R}, \qquad \nabla^* = R\nabla, \tag{2.1}$$

$$V^* = \frac{V}{c_\infty}, \qquad \Phi^* = \frac{\Phi}{c_\infty R}, \qquad t^* = \frac{t}{R/c_\infty}, \tag{2.2}$$

where p, l, V, Φ, and t are the pressure, length scale, velocity, velocity potential, and time, respectively.

2.1 Governing Equations

2.1.1 Equation of State

For air, under the conditions of the acoustic problem, the ideal gas assumption is valid. So, the equation of state is

$$p = \rho \mathcal{R} T, \tag{2.3}$$

in which $\mathcal{R}$ is the universal gas constant, and T is the absolute temperature. Since the flow is both inviscid and irrotational, i.e., isentropic, the following relation also exists between the pressure and density

$$\frac{p}{\rho^\gamma} = \text{const.}, \tag{2.4}$$

where γ is the ratio of specific heats. Given the above, one can readily obtain the equation for the speed of sound in the air, c,

$$c^2 = \left(\frac{\partial p}{\partial \rho}\right)_s = \sqrt{\gamma \mathcal{R} T} = \sqrt{\frac{\gamma p}{\rho}}. \tag{2.5}$$

The equation of state and the speed of sound after non-dimensionalization take the form

$$p^* = \frac{1}{\gamma}(\rho^*)^\gamma, \tag{2.6}$$

and

$$(c^*)^2 = (\rho^*)^{\gamma-1}, \tag{2.7}$$

respectively.

2.1.2 Continuity and Momentum Equations

The non-dimensional form of the Euler equations in vector form is

$$\frac{\partial \vec{V}^*}{\partial t^*} + \vec{V}^* \cdot \nabla \vec{V}^* = -\frac{\nabla p^*}{\rho^*}. \tag{2.8}$$

Assuming an irrotational flow, a potential can be defined, Φ^*, such that $\vec{V}^* = \nabla \Phi^*$. Using this definition, the momentum equation can be recast in terms of density

$$\rho^* = \left[1 - (\gamma - 1)\left(\frac{\partial \Phi^*}{\partial t^*} + \frac{\nabla \Phi^* \cdot \nabla \Phi^* - M_\infty^2}{2}\right)\right]^{1/(\gamma-1)}, \tag{2.9}$$

in which M_∞ is the freestream Mach number. Details of this derivation are given in Appendix A. Similarly, the continuity equation

$$\frac{\partial \rho^*}{\partial t^*} + \nabla \cdot (\rho^* \vec{V}^*) = 0, \tag{2.10}$$

can be written in terms of the velocity potential

$$\frac{\partial \rho^*}{\partial t^*} + \nabla \cdot (\rho^* \nabla \Phi^*) = 0. \tag{2.11}$$

Equations (2.9) and (2.11) are then linearized using the small-perturbation assumption. This is a valid assumption since acoustic variables are generally several orders of magnitude smaller than their flow counterparts. Under this assumption, flow variables are split into a mean value (denoted by '0' subscript) plus an acoustic perturbation. That is,

$$\rho^* = \rho_0 + \rho, \qquad \text{and} \qquad \Phi^* = \Phi_0 + \Phi. \tag{2.12}$$

Substitution of small-perturbation assumption into (2.9) and (2.11) results in two sets of equations for the continuity and momentum equations. One set represents the acoustic field

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho_0 \nabla \Phi + \rho \nabla \Phi_0) = 0, \tag{2.13}$$

$$\rho = -\frac{\rho_0}{c_0^2}\left[\frac{\partial \Phi}{\partial t} + (\nabla \Phi_0 \cdot \nabla \Phi)\right], \tag{2.14}$$

while the other set

$$\nabla \cdot (\rho_0 \nabla \Phi_0) = 0, \tag{2.15}$$

$$\rho_0 = \left[1 - (\frac{\gamma - 1}{2})\left(\nabla \Phi_0 \cdot \nabla \Phi_0 - M_\infty^2\right)\right]^{1/(\gamma-1)}, \tag{2.16}$$

represents the mean flow. In deriving these equations, the following approximate binomial expansion is used

$$(\rho^*)^{\gamma-1} = (\rho_0 + \rho)^{\gamma-1} \approx \rho_0^{\gamma-1} + (\gamma - 1)\rho_0^{\gamma-2}\rho, \tag{2.17}$$

in which higher-order terms are neglected.

The momentum equation (2.14) can also be written in terms of the acoustic pressure. Expanding the pressure about the mean value using a Taylor series and ignoring higher-order terms, we have

$$p^* \approx p_0 + \left(\frac{\partial p}{\partial \rho}\right)_{s=s_0} (\rho^* - \rho_0), \tag{2.18}$$

or

$$(p^* - p_0) = c_0^2 (\rho^* - \rho_0). \tag{2.19}$$

So, the relation between the acoustic pressure and the density is

$$p = c_0^2 \rho. \tag{2.20}$$

Using this, the momentum equation becomes

$$p = -\rho_0 \left[\frac{\partial \Phi}{\partial t} + (\nabla \Phi_0 \cdot \nabla \Phi) \right]. \tag{2.21}$$

2.2 Acoustic Problem

2.2.1 Weak Form

The weak form of the acoustic problem is obtained by first multiplying the continuity equation (2.13) by the test function, Ψ, and integrating the weighted residual over the computational domain, Ω,

$$\int_\Omega \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho_0 \nabla \Phi + \rho \nabla \Phi_0) \right] \Psi \, d\Omega = 0. \tag{2.22}$$

Using the Divergence theorem and the vector identity

$$\nabla \cdot (a\vec{A}) = a \nabla \cdot \vec{A} + \nabla a \cdot \vec{A}, \tag{2.23}$$

the above integral can be written in the following weak form

$$\int_\Omega \left[\Psi \frac{\partial \rho}{\partial t} - \nabla \Psi \cdot (\rho_0 \nabla \Phi + \rho \nabla \Phi_0) \right] d\Omega = - \int_\Gamma \Psi (\rho_0 \nabla \Phi + \rho \nabla \Phi_0) \cdot \vec{n} \, d\Gamma, \tag{2.24}$$

in which Γ is the boundary of the domain, Ω, and $\vec{n}$ is the outward normal vector to the boundary.

Given the time-harmonic nature of the acoustic problem, the acoustic potential can be written as

$$\Phi = \phi(x, y, z) \, e^{-i\bar{\omega}t}. \tag{2.25}$$

Here, i is the imaginary unit, $i = \sqrt{-1}$, and $\bar{\omega}$ is the reduced (non-dimensional) frequency, $\bar{\omega} = \omega R/c_\infty$, with ω being the angular frequency of the acoustic source. Similarly, the test function is defined as

$$\Psi = \psi(x, y, z)\, e^{i\bar{\omega}t}, \tag{2.26}$$

compatible with the definition of the dot product in the complex Hilbert space. The mean potential has no time dependence. So, we have

$$\Phi_0 = \phi_0(x, y, z). \tag{2.27}$$

Using equations (2.14) and (2.25)-(2.27) and substituting them in (2.24), yields the weak form of the acoustic problem as a function of acoustic potential, ϕ,

$$\begin{aligned}\int_\Omega \frac{\rho_0}{c_0^2} \big[\bar{\omega}^2 \phi\psi + (u_0^2 - c_0^2)\, \phi_x\psi_x + (v_0^2 - c_0^2)\, \phi_y\psi_y + (w_0^2 - c_0^2)\, \phi_z\psi_z + \\ u_0v_0\, (\phi_x\psi_y + \phi_y\psi_x) + u_0w_0\, (\phi_x\psi_z + \phi_z\psi_x) + v_0w_0\, (\phi_y\psi_z + \phi_z\psi_y) + \\ i\bar{\omega}u_0\, (\phi\psi_x - \phi_x\psi) + i\bar{\omega}v_0\, (\phi\psi_y - \phi_y\psi) + i\bar{\omega}w_0\, (\phi\psi_z - \phi_z\psi)\big]\, d\Omega = \\ -\int_\Gamma \Psi(\rho_0 \nabla\Phi + \rho\nabla\Phi_0) \cdot \vec{n}\, d\Gamma,\end{aligned} \tag{2.28}$$

in which $\nabla = (\partial_x, \partial_y, \partial_z)$, and $\nabla\Phi_0 = (u_0, v_0, w_0)$.

2.2.2 Spatial Discretization

Following the discussion in Chapter 1 about accuracy requirements for simulation of wave-like phenomena, the spectral element method (SEM) [52, 77] is used for spatial discretization. By definition, spectral methods are based on representing a given smooth function $u(x)$ as a truncated series of orthogonal functions

$$u(x) \approx \sum_i a_i f_i(x). \tag{2.29}$$

The function $f_n(x)$ is called orthogonal with respect to the weight $w(x)$ within the interval $[a, b]$, if

$$\int_a^b f_m(x)\, f_n(x)\, w(x)\, dx = \begin{cases} 0 & m \neq n, \\ k^2 & m = n, \;\; k \in \mathbb{R}. \end{cases} \tag{2.30}$$

It can be easily shown that any approximation based on orthogonal functions minimizes the mean square error [5]. such a property, however, does not exist for ordinary functions. In other words, approximations based on orthogonal functions provide the best fit in the sense of mean square error.

Another very important property of spectral methods lies in their rate of convergence as the order of approximation increases. For a smooth function $u(x)$ approximated by an N^{th} order polynomial $u_N(x)$, the error in the approximation is given by

$$\| u - u_N \| \leq \frac{c}{N^\alpha}, \tag{2.31}$$

in which c is a positive constant and α is a parameter related to the number of continuous derivatives of $u(x)$. The relation shows that for infinitely differentiable functions (i.e., smooth), the error is smaller than any power of $1/N$, hence exponential or infinite accuracy. In simple terms, exponential or '*spectral accuracy*' means that if the order of approximation N is doubled, the error in the numerical solution will drop by at least two orders of magnitude. This is significantly different from finite difference, finite volume, or finite element methods, where the error drops by a fixed factor, i.e., behaves as $1/N^p$, where p depends on the scheme and is essentially finite. Exponential decay of the error as a function of order of approximation is shown in Figure 2.1 for the solution of the Helmholtz equation, $u_{xx} + \lambda^2 u = 0$, using the spectral element method. The error is obtained by comparing the numerical results to the exact solution, $u(x) = \cos(\lambda x)$, in $[0, 1]$ using $\lambda = 2\pi$.

A familiar example of a spectral approximation is the Fourier series, in which the sine and cosine functions (orthogonal in the interval $[0, \pi]$) are used. Due to periodicity of trigonometric functions, discrete Fourier series are best suited for representation of periodic functions. For this reason, when applied to nonperiodic functions, they show slow convergence as the number of terms in the series increases.

For nonperiodic problems, therefore, it is more appropriate to use nonperiodic polynomials. A class of such polynomials, called Jacobi polynomials, $P_n^{\alpha,\beta}(x)$, are available as eigenfunctions of the singular Sturm-Liouville problem in $[-1, +1]$. Using the Rodrigues

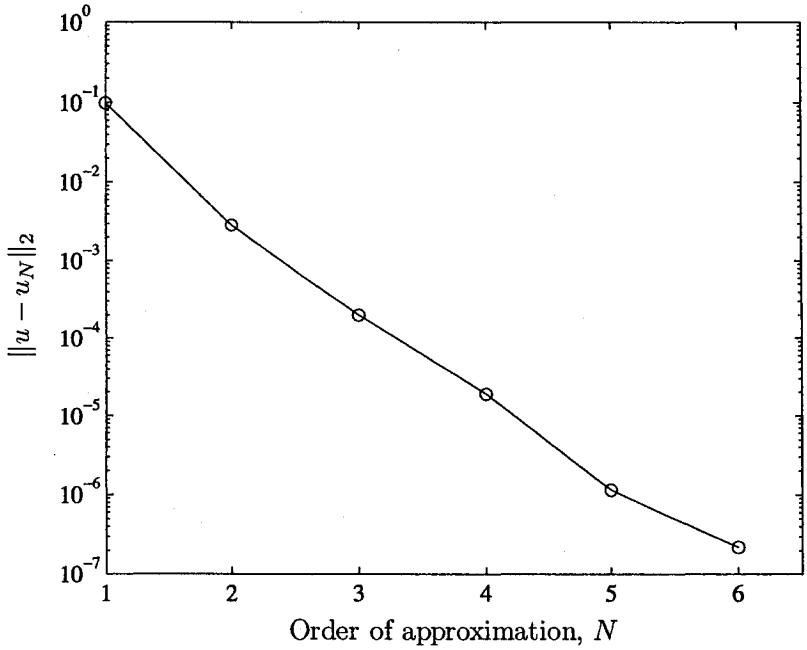

Figure 2.1: Exponential convergence of the spectral method.

formula, the n^{th} degree Jacobi polynomials, $P_n^{\alpha,\beta}(x)$, are

$$P_n^{\alpha,\beta}(x) = \frac{(-1)^n}{2^n n!}(1-x)^{-\alpha}(1+x)^{-\beta}\frac{d^n}{dx^n}[(1-x)^{\alpha+n}(1+x)^{\beta+n}] \qquad \alpha,\beta > -1. \quad (2.32)$$

A sub-class of Jacobi polynomials for which $\alpha = \beta$, called ultra-spherical polynomials, are often used in spectral approximation of nonperiodic functions. Well-known examples are Legendre ($\alpha = \beta = 0$), and Chebyshev ($\alpha = \beta = -\frac{1}{2}$) polynomials. Reference [29] provides a detailed discussion on orthogonal polynomials and their properties.

In the present formulation of the spectral element method, Chebyshev polynomials are used as basis functions although Legendre polynomials could also be used [61, 83, 87]. Chebyshev polynomials of order n are defined as

$$T_n(x) = \cos(n \cos^{-1} x). \quad (2.33)$$

These functions are orthogonal in $[-1,+1]$ with respect to the weight $w(x) = (1-x^2)^{-1/2}$,

$$\int_{-1}^{+1} \frac{T_m(x)\,T_n(x)}{\sqrt{1-x^2}}\,dx = \begin{cases} 0 & m \neq n, \\ c_n \dfrac{\pi}{2} & m = n, \end{cases} \tag{2.34}$$

where $c_0 = 2$, and $c_n = 1$ for $n > 0$. Using the transformation, $x = \cos\theta$, Chebyshev polynomials can also be written as

$$T_n(\cos\theta) = \cos(n\theta). \tag{2.35}$$

The first few Chebyshev polynomials can be easily derived with the help of trigonometric identities. They are

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_2(x) = 2x^2 - 1, \quad T_3(x) = 4x^3 - 3x. \tag{2.36}$$

Higher-order polynomials can be obtained using the recursion relation

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \qquad n > 1. \tag{2.37}$$

2.2.3 Spectral Element Method

The spectral element method (SEM) is in fact a p-version finite element, where spectral approximation is used within each element. This method, first proposed by Patera [77], has since been widely used in many applications because it combines the geometric flexibility of finite element method with the spatial accuracy associated with spectral methods.

Spatial discretization proceeds by first subdividing the computational domain, Ω, into E elements, each of the size Ω^e, such that, $\Omega = \bigcup_{i=1}^{E} \Omega_i^e$, and $\bigcap_{i=1}^{E} \Omega_i^e = \varnothing$. In the present formulation, Ω^e represents a hexahedral element with curved edges (surfaces), in general. Each element is mapped into the reference element, the bi-unit cuboid $\mathcal{Q} = [-1,+1]^3$, using the affine transformation

$$\mathbf{x}(\xi,\eta,\zeta) = \sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{k=0}^{N} h_i(\xi)\,h_j(\eta)\,h_k(\zeta)\,(x,y,z)_{ijk}^e, \tag{2.38}$$

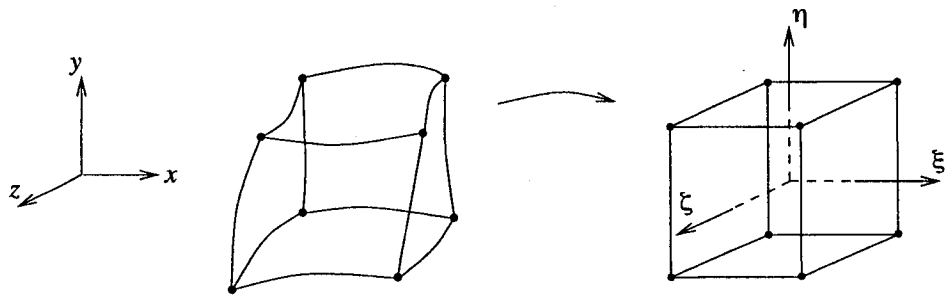

Figure 2.2: Coordinate transformation from physical domain to reference element.

where (x, y, z) are coordinates in the physical space and $\mathbf{x}(\xi, \eta, \zeta)$ the corresponding values in the transformed space (Fig. 2.2).

The spatial approximation is obtained using a tensor product of the N^{th} order interpolation functions, $h(\sigma)$, which are Lagrangian interpolants collocated at the coordinates of the Gauss-Chebyshev-Lobatto (GCL) points, σ_i. That is,

$$h_m(\sigma_n) = \delta_{mn}, \qquad m, n = 0, \ldots, N, \tag{2.39}$$

where δ_{mn} is the Kronecker delta

$$\delta_{ij} = \begin{cases} 1 & i = j, \\ 0 & i \neq j, \end{cases} \tag{2.40}$$

and σ_i are defined as

$$\sigma_i = -\cos\left(\frac{\pi i}{N}\right), \qquad i = 0, \ldots, N. \tag{2.41}$$

Interpolating functions, $h(\sigma)$, are built using a truncated series of the Chebyshev polynomials

$$h(\sigma) = \sum_{n=0}^{N}{}'' a_n T_n(\sigma), \tag{2.42}$$

where $''$ indicates that the first and the last term of the series is to be halved. Introducing the coefficient, c_n,

$$c_n = \begin{cases} 2 & n = 0, N, \\ 1 & \text{otherwise}, \end{cases} \tag{2.43}$$

one can write

$$h(\sigma) = \sum_{n=0}^{N} \frac{1}{c_n} a_n T_n(\sigma), \tag{2.44}$$

in which the only unknowns are the coefficients, a_n. In order to obtain these coefficients, the discrete orthogonal property of Chebyshev polynomials is exploited [18], giving

$$a_n = \frac{2}{N} \sum_{m=0}^{N} \frac{1}{c_m} h(\sigma_m)\, T_n(\sigma_m). \tag{2.45}$$

At any collocation point m, this relation simplifies to

$$a_n = \frac{2}{N} \frac{1}{c_m} T_n(\sigma_m), \tag{2.46}$$

due to the fact that $h_m(\sigma_n) = \delta_{mn}$. This leads to

$$h_m(\sigma) = \frac{2}{N c_m} \sum_{n=0}^{N} \frac{1}{c_n} T_n(\sigma_m)\, T_n(\sigma), \tag{2.47}$$

which is the N^{th} degree interpolating polynomial in terms of the Chebyshev functions, and is used as the basis function to build spatial approximation within elements.

For the sake of clarity in formulations, from this point on the following abbreviated notation

$$\sum_{ijk} = \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=0}^{N}, \tag{2.48}$$

will be used to indicate the triple sum. Also, summation over repeated indices is assumed, whenever meaningful. Therefore, equation (2.47) can be written in the following form

$$h_m(\sigma) = \frac{2}{N c_m c_n} T_n(\sigma_m)\, T_n(\sigma). \tag{2.49}$$

To approximate the acoustic potential, an isoparametric formulation is used, i.e., the polynomial basis for the solution and the geometry are the same. So,

$$\phi = \sum_{ijk} h_i(\xi)\, h_j(\eta)\, h_k(\zeta)\, \phi^e_{ijk}. \tag{2.50}$$

The test functions are defined as the tensor product of the 1-D interpolation functions, $h_m(\sigma)$,

$$\psi = h_l(\xi)\, h_m(\eta)\, h_n(\zeta). \tag{2.51}$$

Element coefficient matrices for the weak formulation, Eq. (2.28), are then obtained using the transformation defined by (2.38) and the equations (2.47)-(2.51) to give

$$[A^e_{ijk,lmn}]\,\{\phi^e_{lmn}\} = \{f^e_{ijk}\}, \tag{2.52}$$

in which

$$f^e_{ijk} = -\int_{\Gamma^e} \Psi(\rho_0 \nabla\Phi + \rho\nabla\Phi_0)\cdot\vec{n}\,d\Gamma, \tag{2.53}$$

and

$$\begin{aligned} A^e_{ijk,lmn} = {} & \bar{\omega}^2 \mathcal{A}_{ijk,lmn} + \mathcal{A}^{xx}_{ijk,lmn} + \mathcal{A}^{yy}_{ijk,lmn} + \mathcal{A}^{zz}_{ijk,lmn} + \mathcal{A}^{xy}_{ijk,lmn} + \mathcal{A}^{yx}_{ijk,lmn} \\ & + \mathcal{A}^{xz}_{ijk,lmn} + \mathcal{A}^{zx}_{ijk,lmn} + \mathcal{A}^{yz}_{ijk,lmn} + \mathcal{A}^{zy}_{ijk,lmn} + i\bar{\omega}[(\mathcal{A}^{\star x}_{ijk,lmn} - \mathcal{A}^{x\star}_{ijk,lmn}) \\ & + (\mathcal{A}^{\star y}_{ijk,lmn} - \mathcal{A}^{y\star}_{ijk,lmn}) + (\mathcal{A}^{\star z}_{ijk,lmn} - \mathcal{A}^{z\star}_{ijk,lmn})]. \end{aligned} \tag{2.54}$$

The submatrices are given by

$$\mathcal{A}_{ijk,lmn} = \int_{\Omega^e} (\rho_0/c_0^2)\, h_i(\xi)\, h_j(\eta)\, h_k(\zeta)\, h_l(\xi)\, h_m(\eta)\, h_n(\zeta)\, |J|\, d\xi\, d\eta\, d\zeta, \tag{2.55}$$

$$\mathcal{A}^{xx}_{ijk,lmn} = \int_{\Omega^e} [\rho_0(u_0^2 - c_0^2)/c_0^2]\, \mathcal{D}^x_{ijk}\, \mathcal{D}^x_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta, \tag{2.56}$$

$$\mathcal{A}^{yy}_{ijk,lmn} = \int_{\Omega^e} [\rho_0(v_0^2 - c_0^2)/c_0^2]\, \mathcal{D}^y_{ijk}\, \mathcal{D}^y_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta, \tag{2.57}$$

$$\mathcal{A}^{zz}_{ijk,lmn} = \int_{\Omega^e} [\rho_0(w_0^2 - c_0^2)/c_0^2]\, \mathcal{D}^z_{ijk}\, \mathcal{D}^z_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta, \tag{2.58}$$

$$\mathcal{A}^{xy}_{ijk,lmn} = \int_{\Omega^e} (\rho_0 u_0 v_0/c_0^2)\, \mathcal{D}^x_{ijk}\, \mathcal{D}^y_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta, \tag{2.59}$$

$$\mathcal{A}^{xz}_{ijk,lmn} = \int_{\Omega^e} (\rho_0 u_0 w_0/c_0^2)\, \mathcal{D}^x_{ijk}\, \mathcal{D}^z_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta, \tag{2.60}$$

$$\mathcal{A}^{yz}_{ijk,lmn} = \int_{\Omega^e} (\rho_0 v_0 w_0/c_0^2)\, \mathcal{D}^y_{ijk}\, \mathcal{D}^z_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta, \tag{2.61}$$

$$\mathcal{A}^{\star x}_{ijk,lmn} = \int_{\Omega^e} (\rho_0 u_0/c_0^2)\, h_i(\xi)\, h_j(\eta)\, h_k(\zeta)\, \mathcal{D}^x_{lmn}\, d\xi\, d\eta\, d\zeta, \tag{2.62}$$

$$\mathcal{A}^{\star y}_{ijk,lmn} = \int_{\Omega^e} (\rho_0 v_0/c_0^2)\, h_i(\xi)\, h_j(\eta)\, h_k(\zeta)\, \mathcal{D}^y_{lmn}\, d\xi\, d\eta\, d\zeta, \tag{2.63}$$

$$\mathcal{A}^{\star z}_{ijk,lmn} = \int_{\Omega^e} (\rho_0 w_0/c_0^2)\, h_i(\xi)\, h_j(\eta)\, h_k(\zeta)\, \mathcal{D}^z_{lmn}\, d\xi\, d\eta\, d\zeta, \tag{2.64}$$

$$\mathcal{A}^{x\star}_{ijk,lmn} = \int_{\Omega^e} (\rho_0 u_0/c_0^2)\, \mathcal{D}^x_{ijk}\, h_l(\xi)\, h_m(\eta)\, h_n(\zeta)\, d\xi\, d\eta\, d\zeta, \tag{2.65}$$

$$\mathcal{A}^{y\star}_{ijk,lmn} = \int_{\Omega^e} (\rho_0 v_0/c_0^2)\, \mathcal{D}^y_{ijk}\, h_l(\xi)\, h_m(\eta)\, h_n(\zeta)\, d\xi\, d\eta\, d\zeta, \tag{2.66}$$

$$\mathcal{A}^{z\star}_{ijk,lmn} = \int_{\Omega^e} (\rho_0 w_0/c_0^2)\, \mathcal{D}^z_{ijk}\, h_l(\xi)\, h_m(\eta)\, h_n(\zeta)\, d\xi\, d\eta\, d\zeta, \tag{2.67}$$

where due to symmetry

$$\mathcal{A}^{yx}_{ijk,lmn} = (\mathcal{A}^{xy}_{ijk,lmn})^T, \tag{2.68}$$

$$\mathcal{A}^{zx}_{ijk,lmn} = (\mathcal{A}^{xz}_{ijk,lmn})^T, \tag{2.69}$$

$$\mathcal{A}^{zy}_{ijk,lmn} = (\mathcal{A}^{yz}_{ijk,lmn})^T. \tag{2.70}$$

In the above equations, $|J|$ is the determinant of the Jacobian of transformation

$$J = \frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)}, \tag{2.71}$$

and $\mathcal{D}$ are differential operators in the transformed space

$$\begin{Bmatrix} \mathcal{D}^x \\ \mathcal{D}^y \\ \mathcal{D}^z \end{Bmatrix} = \begin{bmatrix} (y_\eta z_\zeta - y_\zeta z_\eta) & (y_\zeta z_\xi - y_\xi z_\zeta) & (y_\xi z_\eta - y_\eta z_\xi) \\ (x_\zeta z_\eta - x_\eta z_\zeta) & (x_\xi z_\zeta - x_\zeta z_\xi) & (x_\eta z_\xi - x_\xi z_\eta) \\ (x_\eta y_\zeta - x_\zeta y_\eta) & (x_\zeta y_\xi - x_\xi y_\zeta) & (x_\xi y_\eta - x_\eta y_\xi) \end{bmatrix} \begin{Bmatrix} \partial/\partial\xi \\ \partial/\partial\eta \\ \partial/\partial\zeta \end{Bmatrix} = [\mathcal{J}] \begin{Bmatrix} \partial_\xi \\ \partial_\eta \\ \partial_\zeta \end{Bmatrix}, \tag{2.72}$$

with, for example,

$$\mathcal{D}^x_{ijk} = \mathcal{D}^x\, [h_i(\xi)\, h_j(\eta)\, h_k(\zeta)]. \tag{2.73}$$

Metric quantities can be easily evaluated starting with equation (2.38). For example,

$$\left(\frac{\partial x}{\partial \xi}\right) = \sum_{ijk} \frac{\partial h_i(\xi)}{\partial \xi}\, h_j(\eta)\, h_k(\zeta)\, x^e_{ijk}. \tag{2.74}$$

At any point (ξ_p, η_q, ζ_r) in the reference element, $\mathcal{Q}$, this becomes

$$\left(\frac{\partial x}{\partial \xi}\right)_{pqr} = \sum_{ijk} \frac{\partial h_i(\xi_p)}{\partial \xi}\, h_j(\eta_q)\, h_k(\zeta_r)\, x^e_{ijk} = \sum_{ijk} D_{ip}\, \delta_{jq}\, \delta_{kr}\, x^e_{ijk}, \tag{2.75}$$

in which

$$D_{ij} = \frac{\partial h_i(\sigma_j)}{\partial \sigma}. \tag{2.76}$$

Given the definition of the Kronecker delta and using contraction, one can write

$$\left(\frac{\partial x}{\partial \xi}\right)_{pqr} = D_{ip}\, x^e_{iqr}. \tag{2.77}$$

That is, differentiation reduces to a matrix-vector operation. Similarly, one obtains

$$\left(\frac{\partial x}{\partial \eta}\right)_{pqr} = D_{jq}\, x^e_{pjr}, \qquad \left(\frac{\partial x}{\partial \zeta}\right)_{pqr} = D_{kr}\, x^e_{pqk}, \qquad \ldots. \tag{2.78}$$

Matrix D, the so-called Chebyshev differentiation matrix, is a $(N+1)\times(N+1)$ skew-centrosymmetric matrix whose elements can be evaluated explicitly using a discrete Chebyshev transformation [71]

$$\begin{cases} D_{jk} = \dfrac{c_j}{c_k}\dfrac{(-1)^{j+k}}{(\sigma_j - \sigma_k)}, & j \neq k, \\ D_{jk} = \dfrac{-\sigma_j}{2(1-\sigma_j^2)}, & j \neq 0, N, \\ D_{00} = -D_{NN} = -(2N^2+1)/6, \end{cases} \tag{2.79}$$

where σ_i are the Gauss-Chebyshev-Lobatto points given by (2.41), with coefficients c_n defined in (2.43). Once N is known, entries of D are calculated only once and stored for subsequent calculations. Performing differentiation operation using matrix-vector product could be costly if N is high. In that case, it would be more efficient to use fast transform methods [90].

Now that all components are in place, integrals (2.55)-(2.67), which form the element coefficient matrices, can be easily evaluated. Details of the evaluation for sample contributing matrices are presented in Appendix C.

2.2.4 Boundary Conditions

Once the global matrix and the right-hand side vector are built by assembly of element contributions, boundary conditions must be imposed to uniquely define the problem. In the

acoustic problem, three different types of boundaries exist: solid surfaces, acoustic source, and the far-field.

Solid Surfaces

For a typical turbofan engine, solid surfaces include the inner duct, possibly a centerbody, and the nacelle. Since the flow is considered inviscid, only a no penetration condition is applied on all solid surfaces. That is,

$$\begin{cases} \nabla\Phi \cdot \vec{n} = 0 \\ \nabla\Phi_0 \cdot \vec{n} = 0 \end{cases} \quad \text{on } \Gamma_w, \tag{2.80}$$

where $\vec{n}$ is the outward normal vector to the solid boundary, Γ_w. This boundary condition is applied by simply discarding the contour integral on the right-hand side of equation (2.28) because it will be identically zero at such boundaries.

Acoustic Source

Before deriving equations for the boundary conditions at the source, it is assumed that the source is located at a section of the duct with a uniform cross section, whether circular or rectangular, and that freestream mean flow is along the x-axis perpendicular to the plane of the source (yz-plane), as shown in Figure 2.3.

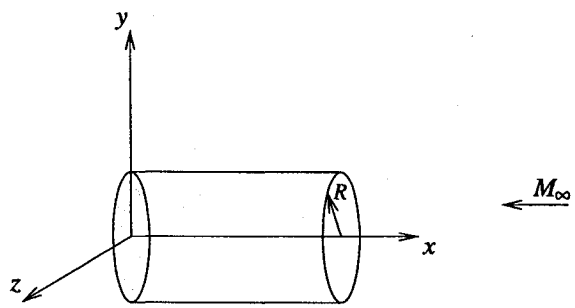

Figure 2.3: Acoustic source in a duct with a circular cross section.

Under these assumptions, the continuity equation (2.13) takes the form

$$\frac{\partial \rho}{\partial t} + \rho_0 \nabla^2 \Phi + M \frac{\partial \rho}{\partial x} = 0. \tag{2.81}$$

Now, using the momentum equation we have

$$\frac{\partial \rho}{\partial t} = -\frac{\rho_0}{c_0^2}\left(\frac{\partial^2 \Phi}{\partial t^2} + M \frac{\partial^2 \Phi}{\partial t\, \partial x}\right), \tag{2.82}$$

and

$$\frac{\partial \rho}{\partial x} = -\frac{\rho_0}{c_0^2}\left(\frac{\partial^2 \Phi}{\partial x\, \partial t} + M \frac{\partial^2 \Phi}{\partial x^2}\right). \tag{2.83}$$

Substituting (2.82) and (2.83) into (2.81) yields

$$\frac{1}{c_0^2}\left(\frac{\partial}{\partial t} + M \frac{\partial}{\partial x}\right)^2 \Phi = \nabla^2 \Phi, \tag{2.84}$$

which is the convected wave equation at the acoustic source[1].

Two different common shapes for the acoustic source are considered: 1) circular and 2) rectangular. While the source shape for fan noise analysis is clearly circular/annular, rectangular shapes are also incorporated to allow a wider use of the code.

In all experimental tests, the amplitude of acoustic perturbations at the source is given in terms of acoustic pressure, p, rather than of the acoustic potential, Φ, which is considered below. Therefore, to perform the tests one needs to know the relationship between the two. This relationship can be easily obtained using the momentum equation (2.21) rewritten here

$$p = -\rho_0 \left[\frac{\partial \Phi}{\partial t} + (\nabla \Phi_0 \cdot \nabla \Phi)\right]. \tag{2.85}$$

At the acoustic source, we have

$$\frac{\partial \Phi}{\partial t} = -i\omega_\ell \Phi, \qquad \nabla \Phi = \frac{\partial \Phi}{\partial x}\vec{\imath} = ik_x \Phi \vec{\imath}, \qquad \text{and} \qquad \nabla \Phi_0 = M_\ell \vec{\imath}. \tag{2.86}$$

So,

$$\nabla \Phi_0 \cdot \nabla \Phi = ik_x M_\ell \Phi, \tag{2.87}$$

[1]Given the non-dimensionalization introduced at the beginning of the chapter, $c_0 = 1$ in this equation.

and the momentum equation at the source reduces to

$$p = -\rho_\ell \left(-i\omega_\ell \Phi + ik_x M_\ell \Phi\right), \tag{2.88}$$

or in terms of the acoustic pressure

$$\Phi = \frac{ip}{\rho_\ell \left(k_x M_\ell - \omega_\ell\right)}. \tag{2.89}$$

Note that both p and Φ in the above equation are complex variables with a time-harmonic component. So, the acoustic potential corresponding to a known pressure amplitude, p_{amp}, is

$$\Phi_{\text{amp}} = \frac{ip_{\text{amp}}}{\rho_\ell \left(k_x M_\ell - \omega_\ell\right)}. \tag{2.90}$$

This is the value A_s or A_{mn} used in equations (2.97) and (2.116), respectively.

Circular Cross Section Due to periodicity in the circumferential direction, the acoustic potential at a circular source can be represented by

$$\Phi(x, r, \theta, t) = \varphi(r)\, e^{i(k_x x + m\theta - \omega_\ell t)}, \tag{2.91}$$

in which k_x is the x-component of the wave number, and m an integer representing the circumferential (azimuthal) mode number. Subscript ℓ indicates local values at acoustic source, and ω_ℓ is the source local frequency[2]

$$\omega_\ell = \frac{\omega R}{c_\ell} = \frac{\omega R/c_\infty}{c_\ell/c_\infty} = \frac{\bar{\omega}}{c_\ell^*}. \tag{2.92}$$

Substituting (2.91) into the wave equation (2.84), leads to an eigenvalue problem for φ at the acoustic source (fan face) in the form of the Bessel equation

$$\frac{d^2\varphi}{dr^2} + \frac{1}{r}\frac{d\varphi}{dr} + \left(k_{ms}^2 - \frac{m^2}{r^2}\right)\varphi = 0, \tag{2.93}$$

with eigenvalues k_{ms} given by

$$k_{ms}^2 = (\omega_\ell - k_x M_\ell)^2 - k_x^2. \tag{2.94}$$

[2]Note that ω_ℓ would be different from $\bar{\omega}$ only if $c_\ell^* \neq c_\infty$. This could only happen, when mean flow is not uniform at the fan face.

Here s represents the radial mode number. Numerical solution of equation (2.93) for a known acoustic mode (m, s) will result in k_{ms} as its eigenvalues and radial distribution of acoustic potential, $\varphi(r)$, as its eigenvectors. Once these values are known, one can use equation (2.91) to evaluate acoustic potential at the source and impose it as a Dirichlet boundary condition.

Boundary conditions associated with equation (2.93) are based on the zero normal velocity at the walls and depend on the shape of the duct at the source. For a circular duct without a centerbody, these conditions are

$$\begin{cases} d\varphi/dr = 0 & r = R, \\ \varphi = \varphi_0 & r = 0, \end{cases} \tag{2.95}$$

where φ_0 is given and is finite, and for a duct with a centerbody of radius r_i,

$$\begin{cases} d\varphi/dr = 0 & r = R, \\ d\varphi/dr = 0 & r = r_i. \end{cases} \tag{2.96}$$

Alternatively, one can find $\varphi(r)$ and k_{ms} given the fact that equation (2.93) has an analytical solution [102]

$$\varphi_{ms}(r) = \sum_s A_s J_m(k_{ms}r) + Y_m(k_{ms}r), \tag{2.97}$$

where A_s is in general a function of the radial mode number, s, and J_m and Y_m are the Bessel functions of the first and the second kind and of order m, respectively. Applying the boundary conditions (2.95) and (2.96) to (2.97) leads to

$$J'_m(k_{ms}R) = 0, \tag{2.98}$$

for a duct without a centerbody, and to the system

$$\begin{cases} A_s J'_m(k_{ms}r_i) + Y'_m(k_{ms}r_i) = 0, \\ A_s J'_m(k_{ms}R) + Y'_m(k_{ms}R) = 0, \end{cases} \tag{2.99}$$

for a duct with a centerbody. Solution of (2.98) or (2.99) will give the eigenvalue k_{ms}, which can subsequently be used to evaluate $\varphi(r)$ using (2.97).

In the present work, a mixed approach has been adopted by first solving the eigenvalue problem (2.93) numerically to find k_{ms}, and then using equation (2.97) to evaluate radial acoustic distribution, $\varphi(r)$. This approach has the advantage that no assumptions are made about the distribution of the grid points on the fan face, e.g., being radial. That is, the grid can be completely unstructured and no interpolation will be needed, since an analytical relation is used to evaluate $\varphi(r)$.

To solve eigenvalue problem (2.93) numerically, we proceed by first building the weak form assuming w are the test functions

$$\int_{\Sigma} \left[\frac{d^2\varphi}{dr^2} + \frac{1}{r}\frac{d\varphi}{dr} + \left(k_{ms}^2 - \frac{m^2}{r^2} \right) \varphi \right] w \, d\Sigma = 0. \tag{2.100}$$

Here, Σ is the one-dimensional radial region. After integration by parts, the weak form becomes

$$\int_{\Sigma} \frac{d\varphi}{dr}\frac{dw}{dr} r \, dr - \int_{\Sigma} (k_{ms}^2 - \frac{m^2}{r^2}) \, \varphi w r \, dr = \left[wr \frac{d\varphi}{dr} \right]_{\partial\Sigma}. \tag{2.101}$$

Using the boundary conditions (2.95) and (2.96), the weak form reduces to

$$\int_{\Sigma} \left(\frac{d\varphi}{dr}\frac{dw}{dr} + \frac{m^2}{r^2} \varphi w \right) r \, dr - k_{ms}^2 \int_{\Sigma} \varphi w r \, dr = 0. \tag{2.102}$$

Equation (2.102) is discretized using a 1-D spectral element formulation by dividing the radial domain, Σ, into 40 elements of equal length. The order of interpolation, N, in each element is taken to be the same as for the 3D hexahedral elements. Details of discretization for equation (2.102) are given in Appendix D. After discretization, the resulting eigenvalue system

$$([K] - k_{ms}^2[M])\{\varphi\} = 0 \tag{2.103}$$

is solved to obtain eigenvalues k_{ms}, using routines from the LAPACK package[3].

It is worth mentioning that for any given propagation mode, (m, s), there are only certain harmonics of blade-passage frequency that will propagate in the duct, eventually radiating into the far-field. These are waves with a real wave number, k_x. Using equation(2.94)

[3]LAPACK routines for a wide range of linear algebra applications are available freely from Netlib repository: `www.netlib.org`.

and solving for the axial wave number gives

$$k_x^{\pm} = \omega_\ell \left[\frac{-M_\ell \pm \sqrt{1-(\beta k_{ms}/\omega_\ell)^2}}{\beta^2} \right], \qquad \beta = \sqrt{1-M_\ell^2}, \tag{2.104}$$

where $\pm$ indicates right- and left-propagating waves, respectively. Here, only right-propagating waves are considered, which correspond to the "+" sign. For k_x to be real, the term under the square root must be positive, i.e.,

$$1 - \left(\frac{\beta k_{ms}}{\omega_\ell} \right)^2 > 0. \tag{2.105}$$

Frequencies that satisfy this relation are called *cut-on*. That is, waves associated with these frequencies will propagate along the duct undamped. To determine whether a certain mode will propagate (cut-on) or will be damped (cut-off), a parameter called cut-off ratio is defined

$$\xi_{ms} = \frac{\bar{\omega}}{\beta k_{ms}}. \tag{2.106}$$

So, the mode (m, s) at a given frequency $\bar{\omega}$ will propagate if $\xi_{ms} > 1$. Otherwise, the corresponding acoustic waves will be damped in the duct and will not contribute to the radiated acoustic energy.

Rectangular Cross Section For a source with a rectangular cross section, (Figure 2.4), there is no circumferential symmetry and therefore the acoustic potential takes the form

$$\Phi = \varphi(y, z)\, e^{i(k_x x - \omega_\ell t)}. \tag{2.107}$$

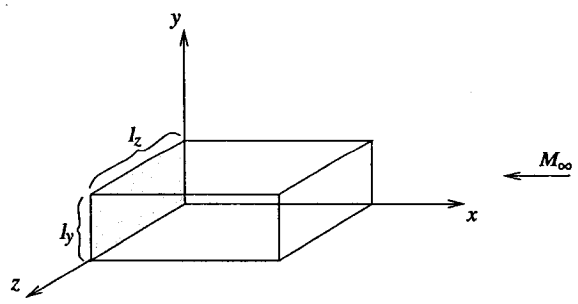

Figure 2.4: Acoustic source in a duct with a rectangular cross section.

Substituting this into the convected wave equation (2.84), with the Laplacian ∇^2 written in Cartesian coordinates, yields the Helmholtz equation

$$\frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} + k_{mn}^2 \varphi = 0, \tag{2.108}$$

with eigenvalues k_{mn} given by

$$k_{mn}^2 = (\omega_\ell - k_x M_\ell)^2 - k_x^2. \tag{2.109}$$

In these equations, subscripts m and n correspond to the duct cross modes in the y- and z-directions, respectively. Equation (2.108) can be solved analytically using the method of separation of variables. Assuming

$$\varphi(y, z) = F(y)\, G(z), \tag{2.110}$$

and substituting into the Helmholtz equation gives two ordinary differential equations with boundary conditions imposed assuming zero normal velocity at the walls

$$\begin{cases} F'' + k_m^2 F = 0, \\ F' = 0, \quad \text{at } y = 0,\ \ell_y, \end{cases} \tag{2.111}$$

$$\begin{cases} G'' + k_n^2 G = 0, \\ G' = 0, \quad \text{at } z = 0,\ \ell_z, \end{cases} \tag{2.112}$$

where $'$ indicates differentiation, and k_m and k_n are components of the wave number in the y- and z-directions, respectively, which are related to the eigenvalue k_{mn} by

$$k_{mn}^2 = k_m^2 + k_n^2. \tag{2.113}$$

Solving (2.111) and (2.112) gives

$$F(y) = A_m \cos(k_m y), \qquad k_m = \frac{m\pi}{\ell_y}, \tag{2.114}$$

and

$$G(z) = A_n \cos(k_n z), \qquad k_n = \frac{n\pi}{\ell_z}. \tag{2.115}$$

Therefore, the general solution of the acoustic potential at the source is

$$\Phi = \sum_{m,n} A_{mn} \left(\cos k_m y\right)\left(\cos k_n z\right) e^{i(k_x x - \omega_\ell t)}, \tag{2.116}$$

in which A_{mn} is the amplitude of the (m, n) mode, and the axial wave number k_x is given by

$$k_x = \omega_\ell \left[\frac{-M_\ell + \sqrt{1 - (\beta k_{mn}/\omega_\ell)^2}}{\beta^2}\right], \qquad \beta = \sqrt{1 - M_\ell^2}. \tag{2.117}$$

The propagating modes correspond to real values of k_x, i.e. when the cut-off ratio, defined as

$$\xi_{mn} = \frac{\bar{\omega}}{\beta k_{mn}}, \tag{2.118}$$

is greater then one.

Equation (2.116) was derived assuming the coordinate axes y and z are aligned with the bottom of the rectangular section, as shown in Fig. 2.4. However, in our program we assume the origin of the coordinate system to be at the center of the duct. Therefore, to calculate acoustic potential at the source, coordinates have to be shifted by the half duct length in the y and z directions. This yields

$$\Phi = \sum_{m,n} A_{mn} \left[\cos k_m (y + \frac{\ell_y}{2})\right]\left[\cos k_n (z + \frac{\ell_z}{2})\right] e^{i(k_x x - \omega_\ell t)}. \tag{2.119}$$

The case of a semi-infinite duct can be treated in a similar way. Assuming that the duct has an infinite length in the z-direction, acoustic potential becomes

$$\Phi = \varphi(y)\, e^{i(k_x x - \omega_\ell t)}, \tag{2.120}$$

which leads to a solution of the form

$$\Phi = \sum_{m} A_m \left(\cos k_m y\right) e^{i(k_x x - \omega_\ell t)}, \qquad k_m = \frac{m\pi}{\ell_y}, \tag{2.121}$$

with k_x defined as before. If the origin of the coordinates system is assumed to be at the center of the duct, then acoustic potential becomes

$$\Phi = \sum_{m} A_m \left[\cos k_m (y + \frac{\ell_y}{2})\right] e^{i(k_x x - \omega_\ell t)}. \tag{2.122}$$

Far-field

Wave-like exterior problems that are allowed to radiate into an infinite domain in the far-field will naturally be attenuated as they travel away from the source or the scatterer. For computational purposes, however, the far-field domain must be truncated. Therefore, exterior boundaries must be such that they appear transparent to the outgoing waves. Otherwise, waves reflected from the boundary will travel back into the domain and contaminate the interior solution, with the situation worsening as the source frequency increases. So, it is clear that having boundary conditions that can simulate a reflectionless effect (less than 1%) are of vital importance [33].

The simplest such boundary conditions, called characteristic boundary condition, requires the incoming characteristic variables to vanish at the boundary [109]. These methods, however, are only optimal when the exterior boundary is perpendicular to the incident waves. Moreover, they usually require the outer boundary to be about 10-20 wavelengths away from the scatterer, unless meshing is done in such a way that the exterior boundary is made almost perpendicular to the outgoing waves [75].

Radiation boundary conditions are another family of methods, based on asymptotic solution of the exterior flow problem with the assumption that the governing equations have a linear behavior far from the source [17, 32, 100].

A broader set of methods, which can be generally called zonal techniques, are based on forming a layer around the exterior boundary in which the outgoing disturbances are attenuated. Terms such as "sponge layers" [78], "buffer zones" [96], "exit zones" [28], and "absorbing layers" [81] are commonly used in the literature when referring to various implementations of this technique.

The most well-known of these methods is the perfectly matched layer (PML), first proposed by Berenger [6] for the solution of exterior electromagnetic problems. The idea behind PML is to attribute to the layers surrounding the computational domain material properties that modify the original field equations, so that the incoming waves will decay within the layer irrespective of their angle of incidence or frequency, thus creating a

reflectionless boundary.

PML methods are generally implemented in rectilinear coordinates system, assuming the absorbing layer to form a rectangular box. However, they can also be implemented in curvilinear coordinate systems, as shown in [9] and [109]. While originally introduced for the solution of problems dealing with electromagnetic wave scattering, PML methods have also been widely used in the solution of aeroacoustic problems [14, 43].

In the present work, however, another type of absorbing layer boundary condition, which is easier to implement, is used. It is based on simply adding a damping term to the governing equations; unlike the PML methods which require modifying the governing equations in the matched layer. This is consistent with our approach in building an efficient yet simple method for fan noise prediction. This concept was first introduced by Israeli and Orszag [45], and subsequently used by Kosloff and Kosloff [53] and others [28, 93, 109]. As in PML, in this method frequency components are equally damped because the damping factor is independent of the frequency.

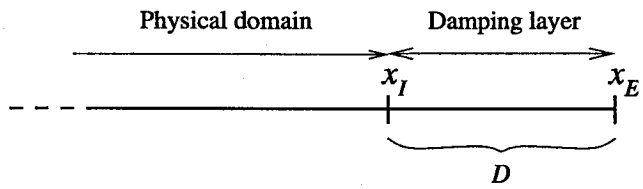

Figure 2.5: Schematic of damping layer.

As shown in Figure 2.5, an absorbing layer is built by extending the physical domain in the far-field direction beyond the outer boundary edge, x_I. The thickness of the layer D is usually taken to be about 2-3 wavelengths. Within the layer, the continuity equation (2.13) is modified by adding a term to the right-hand side

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho_0 \nabla \Phi + \rho \nabla \Phi_0) = -\nu(x)\rho, \tag{2.123}$$

which simulates exponential damping. The damping coefficient, ν, is often taken to be a

power function

$$\nu(x) = \nu_0 \left| \frac{x - x_I}{D} \right|^n , \tag{2.124}$$

or an exponential function

$$\nu(x) = \nu_0 \left[e^{\delta |(x-x_I)/D|} - 1 \right] . \tag{2.125}$$

Here, ν_0 is the damping amplitude, and n and δ parameters that determine damping intensity. These parameters are determined using numerical experiments. Some guidelines about choosing proper values for these parameters are given in Chapter 5.

In general, any function $\nu(x)$ used in (2.123) must satisfy the following two conditions

$$\nu(x) = 0 \qquad x \leq x_I, \qquad \text{and} \qquad \frac{d\nu}{dx} = 0 \qquad x = x_I. \tag{2.126}$$

The first condition lets disturbances enter the damping layer at their original value, and the second ensures continuity, and hence smooth transition, of the waves between the physical domain and the damping zone.

As it is apparent from equation (2.123), damping is applied only in one dimensions. So, in multi-dimensional problems overall damping is taken to be a superposition of one-dimensional damping in the x-, y-, and z-directions. This is done by modifying the weak form in the damping layer, where integrals of the form

$$-\int_\Omega (\nu\rho)\Psi \, d\Omega = \int_\Omega \frac{\nu\rho_0}{c_0^2} \left(i\bar{\omega}\phi\psi + u_0\phi_x\psi + v_0\phi_y\psi + w_0\phi_z\psi \right) d\Omega \tag{2.127}$$

are evaluated along each coordinate direction and their contribution are added to the element matrices.

It should also be pointed out that due to strong damping effects at the far-field boundaries, the contour integral in equation (2.28) will have negligible contribution to the solution at the far-field and therefore is neglected in all computations.

2.3 Mean Flow Problem

The presence of the mean flow in the acoustic field affects the radiation pattern of the sound as it propagates into the far-field. Also, strong variations of flow near the engine

inlet - when the mean flow is present - affects the sound scattering off the inlet. Therefore, for a realistic prediction of the acoustic field, it is necessary to solve for the mean flow problem first, since the two sets of equations are uncoupled.

Given the subsonic nature of the external flow and the range of Mach numbers for which irrotational condition and thus the potential flow assumption is valid, the full potential equation is solved, as given by equations (2.15) and (2.16). This also allows taking into account the compressibility effects, which are an important factor in some of the test cases being considered.

2.3.1 Weak Form and Linearization

We start by deriving the weak form for the mean flow problem based on the continuity equation for a steady, compressible flow

$$\nabla \cdot (\rho_0 \nabla \Phi_0) = 0. \tag{2.128}$$

Using the method of weighted residuals, and assuming Ψ_0 to be the weight function, we set

$$\int_\Omega \Big[\nabla \cdot (\rho_0 \nabla \Phi_0)\Big] \Psi_0 \, d\Omega = 0. \tag{2.129}$$

With the help of the vector identity

$$\nabla \cdot (a\vec{A}) = \nabla a \cdot \vec{A} + a \nabla \cdot \vec{A}, \tag{2.130}$$

and then applying the Divergence theorem, the integral can be written in the following weak form

$$\int_\Omega \nabla \Psi_0 \cdot (\rho_0 \nabla \Phi_0) \, d\Omega = \int_\Gamma \Psi_0 \, (\rho_0 \nabla \Phi_0 \cdot \vec{n}) \, d\Gamma, \tag{2.131}$$

in which Γ is the boundary of the domain, and $\vec{n}$ is the outward normal vector to the boundary. Equation (2.131) is a nonlinear equation in terms of Φ_0, because ρ_0 depends on the solution Φ_0 through (2.16), which is rewritten below

$$\rho_0 = \left[1 - (\frac{\gamma - 1}{2}) \left(\nabla \Phi_0 \cdot \nabla \Phi_0 - M_\infty^2\right)\right]^{1/(\gamma-1)}. \tag{2.132}$$

Solution of (2.131), therefore, requires an iterative procedure. The simplest such method is the Jacobi iteration in which (2.131) is linearized using values of ρ_0 from the previous iteration, ρ_0^k. That is, we solve

$$\int_\Omega \nabla\Psi_0 \cdot (\rho_0^k \nabla\Phi_0)\, d\Omega = \int_\Gamma \Psi_0 \,(\rho_0^k \nabla\Phi_0 \cdot \vec{n})\, d\Gamma. \tag{2.133}$$

The Jacobi method, although easy to implement, has only a linear convergence rate. A more efficient approach, with a quadratic convergence, is based on the Newton-Raphson method, which is subsequently presented.

In the Newton-Raphson method, the solution of the nonlinear equation

$$\mathcal{I}(\Phi_0, \rho_0(\Phi_0)) = 0, \tag{2.134}$$

is achieved by the linear expansion of $\mathcal{I}$ about a mean value, $\bar{\Phi}$, using the Taylor series

$$\mathcal{I}(\Phi_0, \rho_0(\Phi_0)) = \mathcal{I}(\bar{\Phi}) + \left(\frac{\partial \mathcal{I}}{\partial \Phi_0}\right)_{\bar{\Phi}} \delta\Phi_0 = 0, \tag{2.135}$$

in which $\delta\Phi_0 = \Phi_0 - \bar{\Phi}$. Rearranging the terms leads to the following linear system

$$\left(\frac{\partial \mathcal{I}}{\partial \Phi_0}\right)_{\bar{\Phi}} \delta\Phi_0 = -\mathcal{I}(\bar{\Phi}), \tag{2.136}$$

which is solved for $\delta\Phi_0$ in each iteration, with updates of the form $\Phi_0 = \bar{\Phi} + \delta\Phi_0$, until the residual, i.e, the right-hand side vector, becomes negligible.

For the weak form (2.131), the functional $\mathcal{I}(\Phi_0)$ is

$$\mathcal{I}(\Phi_0) = \int_\Omega \nabla\Psi_0 \cdot (\rho_0 \nabla\Phi_0)\, d\Omega - \int_\Gamma \Psi_0 \,(\rho_0 \nabla\Phi_0 \cdot \vec{n})\, d\Gamma. \tag{2.137}$$

As will be discussed later in the section related to the boundary conditions, linearization of the above functional only involves the volume integral. Therefore, linearizing with respect to the mean flow potential, Φ_0, we have

$$\begin{aligned} \frac{\partial \mathcal{I}}{\partial \Phi_0} &= \frac{\partial}{\partial \Phi_0} \int_\Omega \nabla\Psi_0 \cdot (\rho_0 \nabla\Phi_0)\, d\Omega, \\ &= \int_\Omega \left[\left(\frac{\partial \rho_0}{\partial \Phi_0}\right) (\nabla\Psi_0 \cdot \nabla\Phi_0) + \rho_0 \frac{\partial}{\partial \Phi_0} (\nabla\Psi_0 \cdot \nabla\Phi_0) \right] d\Omega. \end{aligned} \tag{2.138}$$

Using (2.132), gives

$$\frac{\partial \rho_0}{\partial \Phi_0} = -\frac{1}{2}\,\rho_0^{2-\gamma}\,\frac{\partial}{\partial \Phi_0}(\nabla\Phi_0 \cdot \nabla\Phi_0), \tag{2.139}$$

which is substituted back to give the linearized weak form

$$\int_\Omega \Big[\rho_0\,(\nabla\Psi_0 \cdot \nabla\delta\Phi_0) - \rho_0^{2-\gamma}(\nabla\Psi_0 \cdot \nabla\Phi_0)(\nabla\Phi_0 \cdot \nabla\delta\Phi_0)\Big]\,d\Omega = \\ -\int_\Omega \nabla\Psi_0 \cdot (\rho_0\nabla\Phi_0)\,d\Omega + \int_\Gamma \Psi_0\,(\rho_0\nabla\Phi_0 \cdot \vec{n})\,d\Gamma. \tag{2.140}$$

This can be written in the matrix form

$$[B]^k\{\delta\Phi_0\} = -\{\mathcal{I}\}^k, \tag{2.141}$$

in which k is the iteration number. Starting with an initial guess, $\Phi_0^k = \bar{\Phi}$, mean flow density ρ_0^k is calculated from (2.132) and the above linear system is solved for $\delta\Phi_0$. The iteration then continues by updating Φ_0

$$\Phi_0^{k+1} = \Phi_0^k + \delta\Phi_0, \tag{2.142}$$

until the norm of the residual, $\|\mathcal{I}^k\|$, drops below a certain tolerance.

2.3.2 Spatial Discretization

Since, to solve the acoustic problem the mean flow variables (ρ_0, u_0, v_0, w_0) are needed at the Gauss-Chebyshev-Lobatto points, we seek the solution of the mean flow problem on the same grid, i.e., the spectral grid. However, such a high resolution would not be necessary if only the solution of the mean flow problem was sought.

We start the spatial discretization of (2.140) by first representing the variation in the mean potential, $\delta\Phi_0$, in discrete form in terms of the Lagrange interpolation polynomials collocated at the Gauss-Chebyshev-Lobatto points, as defined in (2.47). So, we have

$$\delta\Phi_0 = \sum_{ijk} h_i(\xi)\,h_j(\eta)\,h_k(\zeta)\,\delta\phi^e_{0,ijk}. \tag{2.143}$$

with the weight function, Ψ_0, defined as the tensor product of the one-dimensional interpolation functions

$$\Psi_0 = h_l(\xi)\, h_m(\eta)\, h_n(\zeta). \tag{2.144}$$

Substituting for $\delta\Phi_0$ and Ψ_0 from the above equations into (2.140) and noting that $\nabla\Phi_0 = (u_0, v_0, w_0)$, the linearized weak form integrated over an element, Ω^e, can be written as

$$[B^e_{ijk,lmn}]\,\{\delta\phi^e_{0,lmn}\} = \{g^e_{ijk}\}, \tag{2.145}$$

in which the element coefficient matrix $[B^e]$ is

$$\begin{aligned} B^e_{ijk,lmn} &= \mathcal{B}^{xx}_{ijk,lmn} + \mathcal{B}^{yy}_{ijk,lmn} + \mathcal{B}^{zz}_{ijk,lmn} + \mathcal{B}^{xy}_{ijk,lmn} + \mathcal{B}^{yx}_{ijk,lmn} \\ &\quad + \mathcal{B}^{xz}_{ijk,lmn} + \mathcal{B}^{zx}_{ijk,lmn} + \mathcal{B}^{yz}_{ijk,lmn} + \mathcal{B}^{zy}_{ijk,lmn}, \end{aligned} \tag{2.146}$$

with the submatrices defined as follows

$$\mathcal{B}^{xx}_{ijk,lmn} = \int_{\Omega^e} (\rho_0 - \rho_0^{2-\gamma} u_0^2)^k\, \mathcal{D}^x_{ijk}\, \mathcal{D}^x_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta, \tag{2.147}$$

$$\mathcal{B}^{yy}_{ijk,lmn} = \int_{\Omega^e} (\rho_0 - \rho_0^{2-\gamma} v_0^2)^k\, \mathcal{D}^y_{ijk}\, \mathcal{D}^y_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta, \tag{2.148}$$

$$\mathcal{B}^{zz}_{ijk,lmn} = \int_{\Omega^e} (\rho_0 - \rho_0^{2-\gamma} w_0^2)^k\, \mathcal{D}^z_{ijk}\, \mathcal{D}^z_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta, \tag{2.149}$$

$$\mathcal{B}^{xy}_{ijk,lmn} = \int_{\Omega^e} -(\rho_0^{2-\gamma} u_0 v_0)^k\, \mathcal{D}^x_{ijk}\, \mathcal{D}^y_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta, \tag{2.150}$$

$$\mathcal{B}^{xz}_{ijk,lmn} = \int_{\Omega^e} -(\rho_0^{2-\gamma} u_0 w_0)^k\, \mathcal{D}^x_{ijk}\, \mathcal{D}^z_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta, \tag{2.151}$$

$$\mathcal{B}^{yz}_{ijk,lmn} = \int_{\Omega^e} -(\rho_0^{2-\gamma} v_0 w_0)^k\, \mathcal{D}^y_{ijk}\, \mathcal{D}^z_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta. \tag{2.152}$$

Note that, due to the symmetry, we have

$$\mathcal{B}^{yx}_{ijk,lmn} = (\mathcal{B}^{xy}_{ijk,lmn})^T, \tag{2.153}$$

$$\mathcal{B}^{zx}_{ijk,lmn} = (\mathcal{B}^{xz}_{ijk,lmn})^T, \tag{2.154}$$

$$\mathcal{B}^{zy}_{ijk,lmn} = (\mathcal{B}^{yz}_{ijk,lmn})^T. \tag{2.155}$$

Similarly, the right-hand side vector, $\{g^e_{ijk}\}$, is given by

$$g^e_{ijk} = [C^e_{ijk,lmn}]\,\{\Phi^{e,k}_{0,lmn}\}, \tag{2.156}$$

where

$$C^e_{ijk,lmn} = C^{xx}_{ijk,lmn} + C^{yy}_{ijk,lmn} + C^{zz}_{ijk,lmn}, \tag{2.157}$$

and the matrices are defined as

$$C^{xx}_{ijk,lmn} = \int_{\Omega^e} -\rho_0^k\, \mathcal{D}^x_{ijk}\, \mathcal{D}^x_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta, \tag{2.158}$$

$$C^{yy}_{ijk,lmn} = \int_{\Omega^e} -\rho_0^k\, \mathcal{D}^y_{ijk}\, \mathcal{D}^y_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta, \tag{2.159}$$

$$C^{zz}_{ijk,lmn} = \int_{\Omega^e} -\rho_0^k\, \mathcal{D}^z_{ijk}\, \mathcal{D}^z_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta. \tag{2.160}$$

It should be noted that the contribution from the surface integral over the engine inlet, Γ_f, should be properly added to the right-hand side vector in (2.156). In all the above equations, the differential operator $\mathcal{D}$ is given by (2.72), and the superscript k refers to the values from the previous iteration. Evaluation of the integrals in the above equations is done in a manner very similar to those presented for the acoustic problem in Appendix C.

2.3.3 Boundary Conditions

In (2.140), the boundary Γ is composed of the solid surfaces, far-field, and the engine inlet, $\Gamma = \Gamma_w \cup \Gamma_\infty \cup \Gamma_f$. Since the flow is inviscid, solid surfaces are considered to be impermeable to the flow, and therefore the no-penetration boundary condition

$$\nabla\Phi_0 \cdot \vec{n} = 0 \qquad \text{on } \Gamma_w, \tag{2.161}$$

is applied by simply neglecting the contour integral in (2.140) over such surfaces.

On the far-field boundary, Γ_∞, which is considered to be at a sufficiently large distance from the engine inlet, the flow can be considered uniform with

$$\nabla\Phi_0 = (M_\infty, 0, 0). \tag{2.162}$$

On this boundary, a Dirichlet boundary condition is imposed for the mean flow potential Φ_0 by simply integrating (2.162) in the x-direction. This gives

$$\Phi_0 = M_\infty x \qquad \text{on } \Gamma_\infty. \tag{2.163}$$

The last portion of the boundary corresponds to the engine inlet, Γ_f, for which the mass flow rate ($\dot{m}$) is known. Given the area of the engine inlet, A_f, one can write

$$\rho_0 \nabla\Phi_0 \cdot \vec{n} = \dot{m}/A_f, \tag{2.164}$$

and the boundary integral becomes

$$\begin{aligned}\int_\Gamma \Psi_0 \left(\rho_0 \nabla\Phi_0 \cdot \vec{n}\right) d\Gamma &= \int_{\Gamma_f} \Psi_0 \left(\rho_0 \nabla\Phi_0 \cdot \vec{n}\right) d\Gamma, \\ &= \frac{\dot{m}}{A_f} \int_{\Gamma_f} \Psi_0 \, d\Gamma,\end{aligned} \tag{2.165}$$

which is evaluated numerically using a proper weight function on the two-dimensional surface representing the inlet.

Given the fact that $\Psi_0 = 0$ on the far-field boundary, where Dirichlet boundary condition is applied, and that boundary integral vanishes on Γ_w, the contour integral needs to be evaluated on Γ_f only, in which case it is written in terms of the mass flow rate and therefore is not dependent on Φ_0. That is the reason why only the volume integral in (2.131) is considered, when linearizing the weak form using the Newton-Raphson method.

2.4 Geometric Symmetry

It is not very uncommon for the inlet duct to have a geometric symmetry, whether with respect to the xy-plane, xz-plane, or both (axisymmetry). So, one can properly take advantage of the symmetry conditions and perform calculations for only half or quarter of the domain[4]. This means a reduction of computational cost associated with building coeffi-

[4]In the case of axisymmetry, mathematically one can reduce the domain to even less than a quarter. However, since in our code the requirement for the exit boundaries is to be perpendicular to one of the coordinate axes, we are restricted to a minimum size of a quarter.

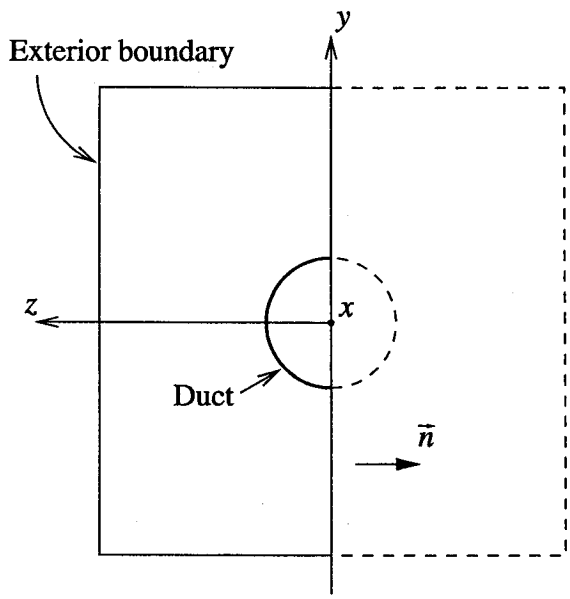

Figure 2.6: Front view of a duct in a symmetric domain.

cient matrices by nearly a factor of 2 or 4, respectively. The cost associated with solving the system of equations, however, will be further reduced because operation counts are non-linear (exponential) functions of the number of equations, whether a direct or an iterative solver is used. Therefore, it is important to take advantage of the symmetry of a problem to substantially reduce the cost.

Removing part of the computational domain due to symmetry will expose the plane of symmetry as an exterior boundary, along which one should evaluate the boundary integral

$$-\int_\Gamma \Psi(\rho_0 \nabla\Phi + \rho\nabla\Phi_0)\cdot\vec{n}\,d\Gamma, \tag{2.166}$$

for the acoustic problem and

$$\int_\Gamma \Psi_0\,(\rho_0\nabla\Phi_0\cdot\vec{n})\,d\Gamma, \tag{2.167}$$

for the mean flow problem, respectively, according to the weak forms (2.24) and (2.131). Figure 2.6 shows the front view of a circular duct symmetric with respect to the xy-plane, in which half the computational domain, e.g. $z < 0$, has been removed. Referring to this figure, we now discuss details of implementation for the mean flow and acoustic problems.

Generalization to other cases of geometrical symmetry, e.g., with respect to the xz-plane, is straightforward.

2.4.1 Mean Flow Problem

A closer look at the trimmed domain, Fig. 2.6, shows that in a mean flow problem the symmetry boundary (xy-plane) acts as a wall and therefore the normal component of acoustic potential, $\nabla\Phi_0 \cdot \vec{n}$, will be zero on this surface. As a result, the integral in (2.167) will be zero and will not have any contributions to the coefficient matrix. So, no changes need to be made to the mean flow equations when geometric symmetry is present.

2.4.2 Acoustic Problem

For the acoustic problem, one should in general evaluate the integral in (2.166). Given that on a symmetry boundary the normal component of mean flow velocity is zero, as explained above, the boundary intergral in (2.166) reduces to

$$I_{sym} = -\int_{\Gamma} \Psi \rho_0 (\nabla\Phi \cdot \vec{n})\, d\Gamma. \tag{2.168}$$

In what follows, we develop the formulation for evaluation of the above integral when symmetry exists along the xy-plane. Similar formulations could be easily developed for other cases.

We first separate the integral in (2.168) into two, one covering the area above the x-axis and the other below it

$$I_{sym} = I_{sym})_{y>0} + I_{sym})_{y<0} = I^{+}_{sym} + I^{-}_{sym}. \tag{2.169}$$

We treat the case of $y = 0$ last, because it may not always be present. Also, to derive the equations we use the cylindrical form of the gradient operator along the plane of symmetry

$$\nabla = \frac{\partial}{\partial r}\vec{e}_r + \frac{1}{r}\frac{\partial}{\partial \theta}\vec{e}_\theta + \frac{\partial}{\partial x}\vec{e}_x, \tag{2.170}$$

where the polar angle θ is measured counter clockwise and relative to the y-axis[5].

▶ **Case:** $y > 0$

Along the positive side of the xy-plane, we have $\vec{n} = -\vec{e}_\theta$. Therefore,

$$\nabla\Phi \cdot \vec{n} = \Big(\frac{\partial\Phi}{\partial r}\,\vec{e}_r + \frac{1}{r}\frac{\partial\Phi}{\partial\theta}\,\vec{e}_\theta + \frac{\partial\Phi}{\partial x}\,\vec{e}_x\Big) \cdot \vec{n} = -\frac{1}{r}\frac{\partial\Phi}{\partial\theta}. \tag{2.171}$$

Using the cylindrical form of acoustic potential along the plane of symmetry

$$\Phi = \phi(x,r)\, e^{im\theta} e^{-i\omega t}, \tag{2.172}$$

we get

$$\nabla\Phi \cdot \vec{n} = -\frac{im}{r}\,\phi(x,r)\, e^{im\theta} e^{-i\omega t}. \tag{2.173}$$

Note that on the plane of symmetry and for $y > 0$,

$$\theta = 0, \qquad \text{and} \qquad r = y.$$

So,

$$\nabla\Phi \cdot \vec{n} = -\frac{im}{y}\,\phi(x,y)\, e^{-i\omega t}. \tag{2.174}$$

Similarly, defining the test function along the plane of symmetry to be

$$\Psi = \psi(x,y)\, e^{i\omega t}, \tag{2.175}$$

the boundary integral (2.168) becomes

$$I^{+}_{sym} = im \int_{\Gamma^+} \Big(\frac{\rho_0}{y}\Big)\phi\psi \, d\Gamma. \tag{2.176}$$

To evaluate this integral, we subdivide the domain Γ^+ into rectangular elements, which are indeed faces of the elements located on the symmetry plane.

Let us define an isoparametric transformation between rectangular element faces and reference element $\mathcal{Q} = [-1,+1]^2$ in the (ξ,η)-plane, and express the variation of acoustic

[5]Note that this convention must match the one used for variation of θ when boundary conditions are applied on the acoustic source.

potential and test function on these faces in terms of Lagrangian interpolants collocated at Gauss-Chebyshev-Lobatto points

$$\phi = \sum_{ij} h_i(\xi)\, h_j(\eta)\, \phi^f_{ij}, \qquad \text{and} \qquad \psi = h_l(\xi)\, h_m(\eta). \tag{2.177}$$

Using this approximation, the surface integral in (2.176) evaluated over an element face becomes

$$\tilde{I}^+_{sym} = \sum_{ij} \left[im \int_{\tilde{\Gamma}^+} \left(\frac{\rho_0}{y}\right) h_i(\xi)\, h_j(\eta)\, h_l(\xi)\, h_m(\eta)\, |J|\, d\xi d\eta \right] \phi^f_{ij}. \tag{2.178}$$

As explained in Appendix C, this integral can be evaluated exactly using a Gauss-Chebyshev quadrature. So, we have

$$\tilde{I}^+_{sym} = S^{xy}_{ij,lm}\, \phi^f_{ij}, \quad \text{where} \quad S^{xy}_{ij,lm} = im \sum_{pq} \left(\frac{\rho_0}{y}\right)_{pq} E_{ilp}\, E_{jmq}\, |J|_{pq}, \qquad y > 0, \tag{2.179}$$

and $E_{ijk} = \int_{-1}^{+1} h_i(\xi)\, h_j(\xi)\, h_k(\xi)\, d\xi$, as defined in Appendix C. Once the coefficient matrices $S^{xy}_{ij,lm}$ are evaluated, they should be properly added to the elements of $A^e_{ijk,lmn}$ in (2.54) for nodes on the symmetry plane.

► **Case:** $y < 0$

Using the same convention for θ as before, along the negative side of the xy-plane we have

$$\theta = \pi, \qquad \text{and} \qquad \vec{n} = \vec{e}_\theta.$$

Therefore,

$$\nabla\Phi \cdot \vec{n} = \frac{1}{r}\frac{\partial \Phi}{\partial \theta}. \tag{2.180}$$

We know that in polar coordinates, $(r, \pi) = (-r, 0)$. So, using the polar definition of acoustic potential, Eq. (2.172), we obtain

$$\nabla\Phi \cdot \vec{n} = -\frac{im}{y}\, \phi(x, y)\, e^{-i\omega t}, \tag{2.181}$$

and the boundary integral becomes

$$I^-_{sym} = im \int_{\Gamma^-} \left(\frac{\rho_0}{y}\right) \phi\psi\, d\Gamma, \tag{2.182}$$

which is the same as Eq. (2.176). So, following a similar procedure, the boundary integral for faces below the x-axis will be

$$\tilde{I}^{-}_{sym} = S^{xy}_{ij,lm}\,\phi^{f}_{ij}, \quad \text{where} \qquad S^{xy}_{ij,lm} = im \sum_{pq} \left(\frac{\rho_0}{y}\right)_{pq} E_{ilp}\,E_{jmq}\,|J|_{pq}, \qquad y < 0, \tag{2.183}$$

Equations (2.179) and (2.183) indicate that the surface integral (2.168) will be *zero* when there are no azimuthal modes, i.e. when $m = 0$, irrespective of the state of radial modes. This is to be expected, because radial modes cause perturbations that are axisymmetric with respect to the x-axis. Therefore, for such modes the normal component of acoustic velocity, $\nabla\Phi \cdot \vec{n}$, will be zero on any plane parallel to the x-axis, such as the xy-plane.

► **Case:** $y = 0$

In deriving equations (2.179) and (2.183), the assumption that $y \lessgtr 0$ was made. That is, the possibility of having any points on the x-axis was excluded. Obviously, such cases could happen, making the integrals in (2.176) and (2.182) singular. To resolve the singularity, numerical integration of boundary integrals involving such singular points are performed at the points of a staggered grid based on Gauss-Chebyshev (GC) points instead of Gauss-Chebyshev-Lobatto (GCL) points.

Define the set of N Gauss-Chebyshev points as[6]

$$\hat{\sigma}_i = -\cos\left[\frac{(2i+1)\pi}{2N}\right], \qquad i = 0, \ldots, N-1, \tag{2.184}$$

where N is the number of GCL points used in the spectral grid. The interpolation function based on Chebyshev polynomials, $T_n(\sigma)$, collocated at N Gauss-Chebyshev points is [18]

$$\hat{h}_m(\sigma) = \frac{2}{N} \sum_{n=0}^{N-1} \frac{1}{\hat{c}_n}\, T_n(\hat{\sigma}_m)\, T_n(\sigma), \tag{2.185}$$

[6]If the number of Gauss-Chebyshev points are taken to be the same as that of GCL points, i.e. $N + 1$, collocation points will be introduced on the x-axis, and therefore the problem of singularity will persist.

where $\hat{c}_0 = 2$ and $\hat{c}_n = 1$. Therefore, the boundary integral for singular points becomes

$$\tilde{I}^0_{sym} = \hat{S}^{xy}_{ij,lm}\,\phi^f_{ij}, \quad \text{where} \qquad \hat{S}^{xy}_{ij,lm} = im \sum_{\hat{p}\hat{q}} \left(\frac{\rho_0}{y}\right)_{\hat{p}\hat{q}} E_{il\hat{p}}\, E_{jm\hat{q}}\, |J|_{\hat{p}\hat{q}}, \tag{2.186}$$

and $E_{ij\hat{k}} = \int_{-1}^{+1} h_i(\xi)\, h_j(\xi)\, \hat{h}_k(\xi)\, d\xi$. Variables ρ_0 and y and the Jacobian $|J|$ are evaluated at the GC points $(\hat{p}, \hat{q})$ using a tensor-product approximation at the GCL points defined with interpolating functions in (2.47). It should be noted that any face on the plane of symmetry that intersects the x-axis may not necessarily have a GCL collocation point on the axis itself. So, the singularity of integral in (2.176) or (2.182) will depend on the shape of the grid on the plane of symmetry and the number of GCL points used.

2.5 Construction of the Spectral Grid

To build the spectral grid corresponding to the coordinates x^e_{ijk} in (2.38) from the 8-node hexahedral finite element grid, an algebraic method is used. Algebraic grid generation methods, which are based on interpolation, are generally preferred over methods based on solving partial differential equations for two main reasons: 1) they are much faster to compute, and 2) they allow greater control over the location of grid points inside the domain [49].

The general framework for algebraic grid generation is the *transfinite interpolation* (TFI). Introduced by Gordon and Hall in the early 1970's, TFI is based on interpolating a domain using the information about its primitive curves, i.e., the curves that form the domain [34, 35]. The term "transfinite" refers to the fact that interpolation matches the primitive curves over a nondenumerable (transfinite) number of points, rather than at only a finite number of distinct points, as in most classical interpolation methods.

The building block for the TFI method is the unidirectional interpolation along a curve. Consider Figure 2.7, which shows the region $\mathcal{D}$ bounded by four curves given parametrically as $\mathbf{F}(\xi_i, \eta_j)$, $i, j = 0, 1$, where $\mathbf{F}$ is a vector function, and ξ and η are the parameters. We now define the interpolation along the vertical curves

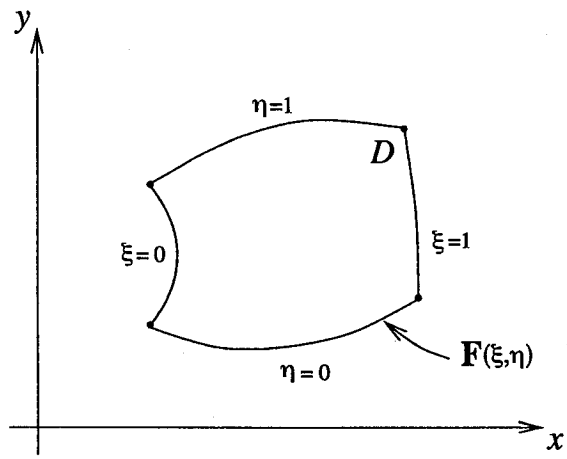

Figure 2.7: Curves representing a typical two-dimensional domain, $\mathcal{D}$.

$$\mathbf{x}(\xi,\eta) = \sum_{l=0}^{1} \psi_l(\xi)\, \mathbf{F}(\xi_l,\eta), \tag{2.187}$$

where $\mathbf{x}(\xi,\eta) = (x,y)$ are the coordinates in the physical plane, and $\psi(\xi)$ are the Lagrange interpolating functions satisfying the cardinality condition, $\psi_i(\xi_j) = \delta_{ij}$, with δ_{ij} being the Kronecker delta. Using this condition, it can be verified that (2.187) reduces to

$$\mathbf{x}(\xi_i,\eta) = \mathbf{F}(\xi_i,\eta), \tag{2.188}$$

along the vertical curves, $\xi_i = \text{const}$. That is, (2.187) interpolates each curve *entirely* and over a transfinite number of nodes. Interpolation along horizontal curves can be represented in a similar way

$$\mathbf{x}(\xi,\eta) = \sum_{m=0}^{1} \varphi_m(\eta)\, \mathbf{F}(\xi,\eta_m), \tag{2.189}$$

where $\varphi_m(\eta)$ are the Lagrangian interpolants, as before.

It is clear that (2.187) and (2.189) are in fact unidirectional interpolations because they interpolate $\mathbf{x}(\xi,\eta)$ along vertical or horizontal curves, only. Also, they can be interpreted as projection operators, $\mathcal{P}_\xi$ and $\mathcal{P}_\eta$, respectively,

$$\mathcal{P}_\xi(\mathbf{F}) = \sum_{l=0}^{1} \psi_l(\xi)\, \mathbf{F}(\xi_l,\eta), \tag{2.190}$$

$$\mathcal{P}_\eta(\mathbf{F}) = \sum_{m=0}^{1} \varphi_m(\eta)\, \mathbf{F}(\xi, \eta_m), \tag{2.191}$$

because they satisfy the so-called idempotent property, $\mathcal{P}^2 = \mathcal{P}$. Functions ψ and φ are called *blending functions* – the interpolation is built as a blend of primitive curves through these functions.

We now move to build a multi-directional or planar interpolation by introducing the tensor product operator

$$\mathbf{x}(\xi, \eta) = (\mathcal{P}_\xi \mathcal{P}_\eta)(\mathbf{F}) = \sum_{l=0}^{1} \sum_{m=0}^{1} \psi_l(\xi)\, \varphi_m(\eta)\, \mathbf{F}(\xi_l, \eta_m), \tag{2.192}$$

which interpolates $\mathbf{F}$ at the corners of the domain only, and not along the primitive curves, in general. To verify this, note that at any corner point (ξ_i, η_j), $i, j = 0, 1$, (2.192) becomes

$$\mathbf{x}(\xi_i, \eta_j) = \mathbf{F}(\xi_i, \eta_j), \tag{2.193}$$

where we have used the cardinality condition. This means that interpolation matches the curves at the points of intersection, satisfying the consistency condition. However, along a horizontal curve ($\eta_j = \text{const.}$), for example, (2.192) becomes

$$\mathbf{x}(\xi, \eta_j) = \sum_{l=0}^{1} \psi_l(\xi)\, \mathbf{F}(\xi_l, \eta_j), \tag{2.194}$$

which is an exact fit only at collocation points of ψ_l and not along the entire curve, in general, unless $\mathbf{F}(\xi, \eta_j)$ is linear.

To build a planar or transfinite interpolation that interpolates the entire region $\mathcal{D}$, the Boolean sum of unidirectional operators must be used, defined as

$$\mathbf{x}(\xi, \eta) = (\mathcal{P}_\xi \oplus \mathcal{P}_\eta)(\mathbf{F}) = (\mathcal{P}_\xi + \mathcal{P}_\eta - \mathcal{P}_\xi \mathcal{P}_\eta)(\mathbf{F}). \tag{2.195}$$

It is easy to verify that (2.195) matches $\mathbf{F}$ on the nondenumerable set of points comprising the boundary of $\mathcal{D}$. A generalization of (2.187) and (2.189) can be obtained when $\mathcal{D}$ is defined using M and L curves in the ξ- and η-directions, respectively. In that case, the

operators become

$$\mathcal{P}_\xi(\mathbf{F}) = \sum_{l=0}^{L-1} \psi_l(\xi)\, \mathbf{F}(\xi_l, \eta), \tag{2.196}$$

$$\mathcal{P}_\eta(\mathbf{F}) = \sum_{m=0}^{M-1} \varphi_m(\eta)\, \mathbf{F}(\xi, \eta_m), \tag{2.197}$$

where ψ and φ are the Lagrange interpolating polynomials of order $(L-1)$ and $(M-1)$, respectively. Using (2.195), an interpolation of the region $\mathcal{D}$ can be obtained, which is transfinite along $L+M$ curves.

Similarly, one can extend this concept to three dimensions assuming that M, L, and N curves are used to interpolate a hexahedral region in the ξ-, η-, and ζ-directions, respectively. Using $\mathbf{F}(\xi, \eta, \zeta)$ to represent the parametric form of the curves, the projectors then take the form

$$\mathcal{P}_\xi(\mathbf{F}) = \sum_{l=0}^{L-1} \psi_l(\xi)\, \mathbf{F}(\xi_l, \eta, \zeta), \tag{2.198}$$

$$\mathcal{P}_\eta(\mathbf{F}) = \sum_{m=0}^{M-1} \varphi_m(\eta)\, \mathbf{F}(\xi, \eta_m, \zeta), \tag{2.199}$$

$$\mathcal{P}_\zeta(\mathbf{F}) = \sum_{n=0}^{N-1} \vartheta_n(\zeta)\, \mathbf{F}(\xi, \eta, \zeta_n), \tag{2.200}$$

with the transfinite interpolation along $ML+MN+LN$ curves given by the triple Boolean sum

$$\begin{aligned} \mathbf{x}(\xi, \eta, \zeta) &= (\mathcal{P}_\xi \oplus \mathcal{P}_\eta \oplus \mathcal{P}_\zeta)(\mathbf{F}), \\ &= (\mathcal{P}_\xi + \mathcal{P}_\eta + \mathcal{P}_\zeta - \mathcal{P}_\xi\mathcal{P}_\eta - \mathcal{P}_\xi\mathcal{P}_\zeta - P_\eta\mathcal{P}_\zeta + \mathcal{P}_\xi\mathcal{P}_\eta\mathcal{P}_\zeta)(\mathbf{F}). \end{aligned} \tag{2.201}$$

We now proceed with the details regarding the construction of the spectral grid from the 8-node hexahedral elements. We begin by assuming that only linear blending functions are used, i.e., $M = L = N = 2$. Therefore, all is needed is the parametric description of the $(2 \cdot 2 + 2 \cdot 2 + 2 \cdot 2)$ 12 edges that form the hexahedral elements, $\mathbf{F}^e(\xi, \eta, \zeta)$. The linear

blending function, $\gamma(\sigma)$, is

$$\begin{cases} \gamma_0(\sigma) = (1-\sigma)/2 \\ \gamma_1(\sigma) = (1+\sigma)/2 \end{cases}, \qquad \sigma \in [-1,+1], \tag{2.202}$$

with $\sigma_0 = -\sigma_1 = -1$. So, the face projectors become

$$\mathcal{P}_\xi(\mathbf{F}^e) = \sum_{i=0}^{1} \gamma_i(\xi)\, \mathbf{F}^e(\xi_i, \eta, \zeta), \tag{2.203}$$

$$\mathcal{P}_\eta(\mathbf{F}^e) = \sum_{j=0}^{1} \gamma_j(\eta)\, \mathbf{F}^e(\xi, \eta_j, \zeta), \tag{2.204}$$

$$\mathcal{P}_\zeta(\mathbf{F}^e) = \sum_{k=0}^{1} \gamma_k(\zeta)\, \mathbf{F}^e(\xi, \eta, \zeta_k), \tag{2.205}$$

the edge projectors

$$\mathcal{P}_\xi\mathcal{P}_\eta(\mathbf{F}^e) = \sum_{i=0}^{1}\sum_{j=0}^{1} \gamma_i(\xi)\,\gamma_j(\eta)\, \mathbf{F}^e(\xi_i, \eta_j, \zeta), \tag{2.206}$$

$$\mathcal{P}_\xi\mathcal{P}_\zeta(\mathbf{F}^e) = \sum_{i=0}^{1}\sum_{k=0}^{1} \gamma_i(\xi)\,\gamma_k(\zeta)\, \mathbf{F}^e(\xi_i, \eta, \zeta_k), \tag{2.207}$$

$$\mathcal{P}_\eta\mathcal{P}_\zeta(\mathbf{F}^e) = \sum_{j=0}^{1}\sum_{k=0}^{1} \gamma_j(\eta)\,\gamma_k(\zeta)\, \mathbf{F}^e(\xi, \eta_j, \zeta_k), \tag{2.208}$$

and the vertex projector

$$\mathcal{P}_\xi\mathcal{P}_\eta\mathcal{P}_\zeta(\mathbf{F}^e) = \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} \gamma_i(\xi)\,\gamma_j(\eta)\,\gamma_k(\zeta)\, \mathbf{F}^e(\xi_i, \eta_j, \zeta_k). \tag{2.209}$$

Let Ξ_σ be the set of Gauss-Chebyshev-Lobatto points

$$\Xi_\sigma = \{\sigma_\ell \,|\, \sigma_\ell = -\cos(\ell\pi/N),\ \ell = 0, \ldots, N\}, \tag{2.210}$$

defined in $[-1,+1]$, where N is the order of polynomial approximation. The spectral grid is then a mapping of the nodal set $\Xi = \Xi_\xi \times \Xi_\eta \times \Xi_\zeta$ in the reference element, $\mathcal{Q}$, onto the hexahedral elements in the physical space. That is,

$$(x, y, z)^e = (\mathcal{P}_\xi \oplus \mathcal{P}_\eta \oplus \mathcal{P}_\zeta)(\mathbf{F}^e)\Big|_\Xi, \tag{2.211}$$

In practice, construction of the spectral grid can be done more easily by first building GCL points along individual edges, and then use face projectors of the form

$$\mathbf{x}(\xi_i, \eta, \zeta) = (\mathcal{P}_\eta \oplus \mathcal{P}_\zeta)(\mathbf{F}^e(\xi_i, \eta, \zeta)), \qquad i = 0, 1, \tag{2.212}$$

$$\mathbf{x}(\xi, \eta_j, \zeta) = (\mathcal{P}_\xi \oplus \mathcal{P}_\zeta)(\mathbf{F}^e(\xi, \eta_j, \zeta)), \qquad j = 0, 1, \tag{2.213}$$

$$\mathbf{x}(\xi, \eta, \zeta_k) = (\mathcal{P}_\xi \oplus \mathcal{P}_\eta)(\mathbf{F}^e(\xi, \eta, \zeta_k)), \qquad k = 0, 1, \tag{2.214}$$

to build the nodes on the faces. Then, the interior nodes are created using a Boolean sum, as in (2.201), defined over the interior nodal set $\tilde{\Xi} = \tilde{\Xi}_\xi \times \tilde{\Xi}_\eta \times \tilde{\Xi}_\zeta$, where $\tilde{\Xi}_\sigma = \Xi_\sigma \backslash \{\sigma_0, \sigma_N\}$.

It is worth mentioning that if the parametric curves that define the hexahedral elements, $\mathbf{F}^e(\xi, \eta, \zeta)$, are all linear, then $\mathcal{P}_\xi = \mathcal{P}_\eta = \mathcal{P}_\zeta$, and given the idempotent property, transfinite interpolation, Eq. (2.211), reduces to the familiar *trilinear* interpolation

$$(x, y, z)^e = \sum_{i=0}^{1} \sum_{j=0}^{1} \sum_{k=0}^{1} \gamma_i(\xi)\, \gamma_j(\eta)\, \gamma_k(\zeta)\, \mathbf{F}^e(\xi_i, \eta_j, \zeta_k)\Big|_\Xi. \tag{2.215}$$

Chapter 3

Parallelization

3.1 Introduction

With the continuous increase in the computational power of microprocessors, which are expected to roughly double every 18 months (according to Moore's law [72]), scientists have found the necessary tools at their disposal to tackle more realistic problems that require large-scale intensive calculations. This includes simulation of larger domain size, using finer grids, problems that are computationally intensive, such as the direct numerical simulation (DNS) of turbulence, or a combination thereof.

In computational aeroacoustics, accuracy requirements of the numerical scheme lead to larger amount of computations due to the increased size of the stencil or the higher order approximation being used. This is specially true for spectral methods, which employ high-order polynomials that have a global support, i.e., support over all the grid points within the element. This, combined with the increased size of the computational domain for realistic simulations, necessitates the use of multi-processors for parallel computations and subsequently a very large memory.

In the following sections, we briefly discuss about different computer architectures and then introduce domain decomposition method as a way of solving large-scale problems in parallel. We then continue by introducing some metrics to access the parallel performance.

The material in this chapter sets the stage for a detailed discussion of the methods used for the solution of the meanflow and acoustic problems, which are presented in Chapter 4.

3.2 Parallel Computer Architectures

Given the fact that in a parallel computer calculations are done on a set of processors, one can conceptualize two different approaches in terms of performing the operations. The first approach, called single-instruction multiple-data (SIMD), requires that all processes perform the same instruction simultaneously but on different datasets in lockstep. This concept, which was used in the first generation of parallel computers, was later replaced by the more flexible multiple-instruction multiple-data (MIMD) approach, in which synchronization is not a constraint. As a result, multiple instructions could be carried out on different processors, each with its own set of data. The majority, if not all, of current parallel computers, including the networks of workstations and Boewolfs, use the MIMD concept because of its flexibility in handling data.

Categorization of parallel computers can also be done independently based on the relation between the processors and the memory, giving rise to the shared-memory and the distributed-memory machines. In the former, Figure 3.1, each processor has a direct access

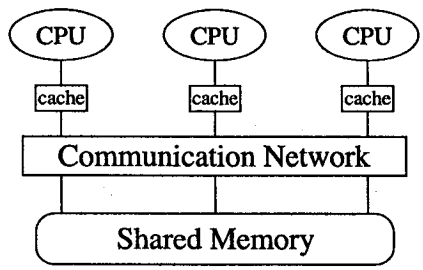

Figure 3.1: Schematic diagram of a shared-memory machine.

to the contents of the global memory (the shared data) almost simultaneously, while in the later, Figure 3.2, each processor has direct access to only a local (distributed) portion of

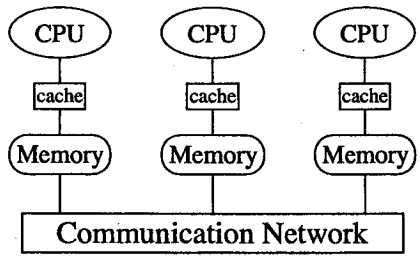

Figure 3.2: Schematic diagram of a distributed-memory machine.

the memory, with the interconnection between them provided through a global communication network. The newer generation of parallel computers usually use a combination of these two concepts. In this new approach, parallelization is achieved through a set of distributed nodes, with each node operating on a shared-memory concept among typically 2-12 processors.

In distributed-memory machines based on the MIMD paradigm, the most commonly used method of performing computations is the single-program multiple-data (SPMD) approach. Here, each processor runs the same copy of the program with a different set of data/instructions associated with it. This means that, in general, variables and arrays within the program will have different values during the execution. In order to exchange data among the processors, one makes calls to a library of message passing routines, which in their primitive form, are simply a call to a *send()* or a *receive()* procedure. Communication among processors creates a loose synchronization, mimicking the SIMD concept, while taking advantage of the MIMD flexibility.

The message passage routines, which are standardized, eliminate the need to create extensions to the original programming language, and therefore lead to programs that are easily portable. There are currently two well-known standards to handle message passing, i.e., MPI (Message Passing Interface) [37] and PVM (Parallel Virtual Machine) [31], with the PVM tailored more toward a network of "heterogeneous" workstations. MPI is the result of a collective effort within the parallel computing community to define a standard for message passing communication among the processors in distributed parallel environ-

ments. For this reason, it is a very popular choice among parallel programmers, gradually replacing PVM.

On shared memory machines, the standard of choice is Open MP. It comprises a set of compiler directives that implement parallel processing in a C, C++, Fortran77 or Fortran90 program. The directives allow the user to mark areas of the code, such as **do**, **while** or **for** loops, which are suitable for parallel processing. The directives appear as a special kind of comment, so the program can be compiled and run in serial mode. However, the user can tell the compiler to "notice" the special directives, in which case a version of the program will be created that runs in parallel. Thus the same program can easily be run in serial or parallel mode on a given computer, or run on a computer that does not have Open MP at all. In what follows, our reference to the computer architecture is that of a distributed memory machine.

In this work, we have made extensive use of the MPI and the PETSc (Portable, Extensible Toolkit for Scientific Computation) software [4] for parallel implementation of our acoustic and mean flow problems. PETSc is a library of basic linear algebra, higher level primitives, distributed data handling routines, and solvers for the numerical solution of PDE's in parallel, with the MPI as its underlying message passing protocol. Use of the PETSc library greatly simplifies the task of parallelization by allowing the programmer to concentrate on the algorithm rather than on implementation of individual tasks.

3.3 Domain Decomposition Methods

A very large class of methods for parallel solution of PDE's are based on the domain decomposition method (DDM). In these methods, as the name implies, the computational domain is first broken down into smaller sections, called subdomains. The solution procedure then consists of assigning one or more subdomains to each processor with proper matching conditions enforced between them through the interconnecting network. Due to their segregated nature, domain decomposition methods are ideal candidates for use on

distributed memory machines because the solution method and the computer architecture follow the same philosophy.

Domain decomposition is also the method of choice when solving problems of different physical nature that coexist within the same domain (e.g., conjugate heat transfer), when different solution methods are available for each subdomain, or when there is simply a need to distribute the computational load across several processors. Therefore, the principal advantage of the domain decomposition methods is enhancement of parallelism and localized treatment of complex and irregular geometries.

In any domain decomposition method, one can distinguish three levels of communication. This is shown schematically in Figure 3.3. The first level corresponds to serial

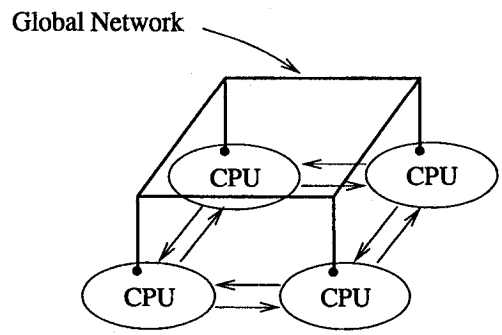

Figure 3.3: Schematic diagram showing the three levels of communication in a distributed memory machine.

computations within each CPU, where the performance is dominated by the clock rate of the CPU and the speed by which the local memory is accessed. In a distributed memory machine, since each CPU has a direct access to its own memory (usually enhanced by the use of cache) fast access to the local memory is not a major concern. The major factor in improving the performance is then getting the most out of the CPU's computational power by reducing its down time. This is accomplished through the use of robust algorithms that improve pipelining by constantly feeding the data to the processor.

The second and the third levels of communication are the local and the global, respec-

tively. The local communication, represented by opposing arrows in the figure, corresponds to the communication between neighboring processors/domains, while the global communication, represented by the wire grid, includes the communication between any two or more processors within the network.

The cost associated with the local and/or global communications depends on the time it takes to pass a message within the network. Theoretically, this can be represented by a linear formula

$$t_c(m) = (\alpha + \beta m)\, t_a, \tag{3.1}$$

in which $t_c(m)$ is the communication time for an m-word message, and t_a is the average time to perform an arithmetic operation such as addition or multiplication. The constants α and β are the message start-up time, also called latency, and the message transform rate, respectively. While t_a depends on the speed of a CPU in performing an arithmetic operation, α and β depend on the bandwidth of the network and the speed at which the memory can be accessed. In general, $\alpha \gg \beta$, indicating that it is more efficient to pass a smaller number of long messages (of the order of $m = 200$), instead of a large number of very short messages.

The communication time given by equation (3.1) is only valid when the number of processors is small, or when communication in the network is very efficient. Otherwise, barrier costs in the network, i.e., the cost of ensuring safe access to the memory, will generally degrade communication time by a factor of $\sqrt{p}$ or $\log p$, where p is the total number of CPU's.

Given the above discussion, one can deduce that the highest cost associated with parallelization in domain decomposition methods is due to the global inter-processor communication. Global communication is generally a high bandwidth process, i.e., one in which the number of messages travelled within the network are of the order of the problem size (total number of unknowns). This is in contrast with the low bandwidth communication, where the amount of messages exchanged is of the order of the number of processors and therefore much less costly, as in the local communication between neighboring domains.

The performance of an algorithm used in domain decomposition, however, does not solely depend on minimizing global communication. For example, consider the use of block Jacobi as a preconditioner for the solution of subdomain problems. Given the localized nature of this preconditioner, creating and implementing it can be done without any need for global communication. However, this will adversely affect the overall iteration count because convergence is directly related to the amount of information exchanged at the global level between various subdomains. So, an optimum algorithm is one that minimizes the level of serial communications on one hand, while on the other hand creates a suitable balance between the amount of local and global communications.

When developing an algorithm, one should also pay attention to the concept of scalability, i.e., maintaining the performance as the number of processors is increased. As a general rule, scalability is degraded with the increase in the number of processors/domains without an increase in the problem size. This is due to the fact that further partitioning of the domain increases their segregation and reduces the global exchange of data among them, an essential factor for a rapid convergence. To remedy this, one has to ensure that a proper level of global communication is maintained among processors as their number is increased, or

3.3.1 Partitioning of the Domain

In domain decomposition methods, proper partitioning of the domain among processors is of fundamental importance. By "proper", we mean a partitioning such that the amount of computations to be performed by each processor will be roughly the same, while the inter-processor communication is kept to a minimum.

Partitioning begins by first constructing a graph representing the connectivity of the nodes for the finite element grid upon which the spectral grid is built. As shown in Figure 3.4, each graph is made of vertices and edges, where each vertex represents an unknown and an edge is built between any two vertices v_i and v_j, if the corresponding entry in the global matrix a_{ij} is nonzero. In the context of the finite element method, this graph is

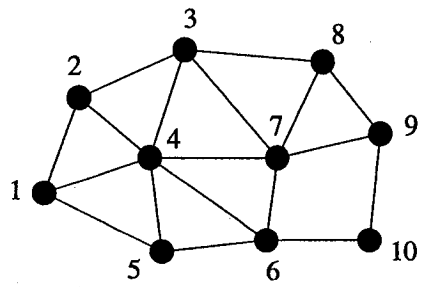

Figure 3.4: An unstructured graph of a sparse matrix.

basically a two-dimensional graph of the connectivity table.

Assuming that the computational domain is to be partitioned according to the number of processors, p, i.e., into p subdomains, the problem then reduces to finding a partitioning in which the number of the vertices (unknowns) in each subdomain are (nearly) equal, while the number of the edges that are cut through are minimized. This will lead to a load-balanced partitioning since each processor will have almost the same amount of computations to perform (equi-distribution of vertices) and the global inter-processor communication (represented by the number of edge-cuts) are kept to a minimum. So, the problem of partitioning of the domain reduces to a multi-constraint optimization problem.

In our work, in order to partition the domain, we have used a publicly available software, called ParMETIS, developed by Karypis and Kumar [47]. Without going into the details, here is a brief description of how the software works. Partitioning is based on parallel implementation of an algorithm that first constructs the graph of the finite element grid using the connectivity information, as explained above. It then uses a multi-level refining approach, where at each level neighboring vertices are collapsed to obtain smaller, simpler graphs. At the last level, the smaller, refined graph is very quickly partitioned using the weights corresponding to the number of vertices collapsed on each other as a metric representing the computational load. Now, reversing the cycle, the method successively uncoarsens the partitioned refined graph to create a partitioning for the original grid. This procedure is shown schematically in Figure 3.5.

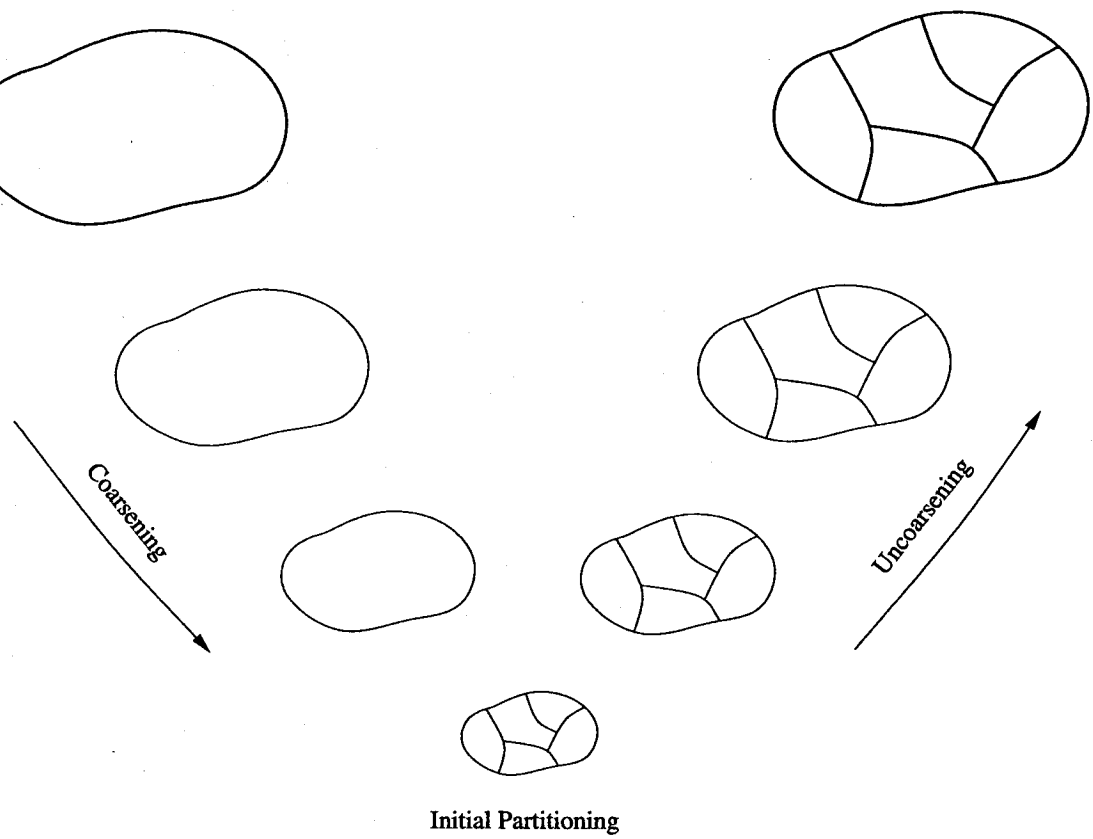

Figure 3.5: Multi-level partitioning of a graph.

The output of ParMETIS is a vector V of size E, E being the total number of elements, with entries of the form $V(e) = p$, where p is the processor number to which element e belongs. That is, partitioning is provided in terms of the element distribution in each subdomain.

3.3.2 Overlapping vs. Nonoverlapping Methods

Domain decomposition methods, which cover a wide range of algorithms, are primarily distinguished by whether the individual subdomains have any overlaps or are disjoint. The first domain decomposition algorithm, due to H.A. Schwarz (1870), was based on an overlapping concept. Although originally intended to prove existence of harmonic functions on

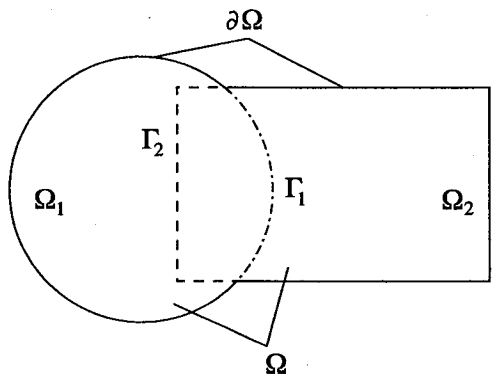

Figure 3.6: Original figure of Schwarz for overlapping subdomains.

irregular regions which are union of overlapping subregions, Schwarz method later became the corner stone for development of a wide range of domain decomposition methods after computers, and subsequently parallel computing, became available.

Due to its fundamental importance, we explain this algorithm for the solution of the problem

$$\mathcal{L}u = f, \tag{3.2}$$

in $\Omega = \Omega_1 \cup \Omega_2$ (Figure 3.6), where $\mathcal{L}$ is a differential operator and u is the vector of unknowns. Let $\partial\Omega$ represent boundary of the domain Ω on which u is given as a Dirichlet boundary condition, with Γ_1 being the portion of $\partial\Omega_1$ that lies within the overlapping region $\Gamma_1 = \partial\Omega_1 \cap \Omega_2$, and likewise for Γ_2. An iterative process is then constructed by first solving for u_1^k in Ω_1,

$$\begin{aligned} \mathcal{L}u_1^k &= f \qquad \text{in} \quad \Omega_1, \\ u_1^k &= g_1 \qquad \text{on} \quad \partial\Omega_1 \setminus \Gamma_1, \\ u_1^k &= u_1^{k-1} \qquad \text{on} \quad \Gamma_1, \end{aligned} \tag{3.3}$$

and then using the newly obtained value of u on Γ_2 to solve for u_2^k in Ω_2,

$$\begin{aligned} \mathcal{L}u_2^k &= f \qquad \text{in} \quad \Omega_2, \\ u_2^k &= g_2 \qquad \text{on} \quad \partial\Omega_2 \setminus \Gamma_2, \end{aligned}$$

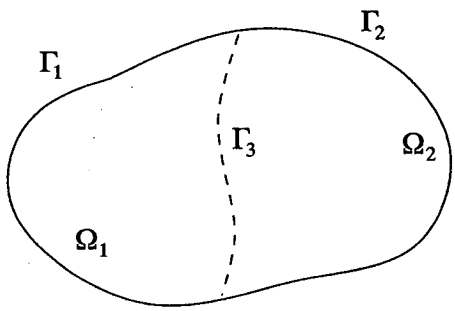

Figure 3.7: Nonoverlapping subdomains.

$$u_2^k = u_1^k|_{\Gamma_2} \quad \text{on} \quad \Gamma_2, \tag{3.4}$$

until convergence is reached. Here k is the iteration number. This method is called the *alternating Schwarz* method due to the alternating nature of the solution between subdomains.

It is readily observed, and can be proven mathematically [13], that the convergence rate of the above iterative method is accelerated as the size of the overlap region is increased. In fact, in the limit when the size of the overlap region is a maximum, i.e., when $\Omega = \Omega_1 = \Omega_2$, the solution will be obtained in only one iteration[1]. The second observation is that in overlapping methods, solution within the overlap region is an inherently iterative process, while within each subdomain both an iterative or a direct method could be used. As will be discussed later in section 4.1, we will use a variant of the Schwarz algorithm, called *additive Schwarz method* (ASM) for the solution of the mean flow equations.

The absence of overlap region between subdomains leads to a class of nonoverlapping domain decomposition methods, generally referred to as *substructuring* or *Schur complement* methods. As shown in Figure 3.7 for a two-subdomain partition, the interface boundary Γ_3 is the medium through which the information between subdomains is exchanged. In another words, the equations corresponding to the points on the interface boundary Γ_3 will contain all the couplings between the degrees of freedom of the global problem.

[1]One should note that increasing the size of the overlap region leads to more computations, which in turn may offset the advantage of lower number of iterations.

Substructuring or Schur complement methods are based on rearranging the equations such that equations corresponding to the unknowns on the shared boundaries are solved first, followed by the solution of a Dirichlet problem in each subdomain independently. We will use such a method for the solution of the acoustic problem, as explained in section 4.2.

3.4 Parallel Performance

The cost associated with parallelization is primarily dependent on two factors:

1. load balancing,

2. inter-processor communication.

As mentioned earlier in section 3.3, there are three levels of communication in a parallel computation, i.e., serial, local, and global. Load balancing represents the serial part and plays a significant role in parallel efficiency by ensuring that each processor performs roughly the same amount of computations, while processor idle times are kept to a minimum. The second factor, i.e., communication, refers to the cost of exchanging or passing messages among processors at the local (between neighboring subdomains) and at the global (in the entire domain) levels.

A very important parameter which directly influences load balancing and communication is the algorithm. For example, during serial calculations within each subdomain an efficient algorithm will ensure that variables that are already brought into the cache are used as many times as possible in order to reduce memory access time. Or, at the global level, it is sometimes more efficient to perform a certain operation on all the processors rather than assigning the job to only one processor and later distributing the result among the rest. All these and many other issues are determined by the algorithm, testifying to its central role in improving the performance.

To measure the efficiency of a parallel computation, generally two different but closely related metrics are used. The first is the parallel efficiency, defined as

$$\eta_p = \frac{t_1}{p\, t_p}. \tag{3.5}$$

Here, p is the total number of processors, t_1 the time to execute the algorithm on a single devoted processor, and t_p the time it takes to execute the same algorithm on p processors. In an ideal situation, $\eta_p = 1$. However, in practice, η_p usually lies within the range $0.8 - 0.9$ for most problems.

The second widely used parameter is called the speed-up, and is given by

$$S_p = \frac{t_1}{t_p}. \tag{3.6}$$

It is expected that at the very least, $\partial S_p / \partial p > 0$, i.e., adding more processors will lead to an overall faster computation. In the ideal case, i.e., when $\eta_p = 1$, we get $S_p = p$. That is, running the same algorithm on p processors will lead to a p-fold increase in the speed of computations. Speed-up is usually measured by plotting S_p vs. p. Ideally, this curve should have a slope of one. However, it is not unexpected that in certain situations one could obtain a slope greater than one, indicating a '*superlinear*' speed-up. Superlinear speed-up usually happens when the nature of the problem is such that the algorithm responds better to a parallel implementation than a serial one (cache-effect). It is also worth mentioning that since both t_1 and t_p refer to the same algorithm, parallel efficiency or speed-up are only weak indicators of the quality of the algorithm.

Chapter 4

Solution Method

In this chapter, the methods used for the solution of the mean flow and acoustic problems are discussed. Solution of the mean flow problem is obtained using an overlapping domain decomposition approach, where the Conjugate Gradient method is used with the additive Schwarz as the preconditioner.

Solution of the acoustic problem is based on the Schur complement method, which is a nonoverlapping technique. After presenting the Schur method, preconditioning of the reduced Schur system is discussed, where we propose a new preconditioner followed by some numerical experiments to justify its suitability. Presentation of the algorithm for implementing the Schur method and the preconditioner conclude this Chapter.

4.1 Solution Method for the Mean Flow Problem

The discretization of equations (2.133) and (2.140), which are weak forms of a Laplacian (continuity equation) in the form of a Jacobi or a Newton iteration, leads to a sparse linear system of equations

$$Au = f, \tag{4.1}$$

where the coefficient matrix, $A \in \mathbb{R}^{n \times n}$ is both symmetric and positive definite (SPD)[1]. Such systems are best solved using the Conjugate Gradient (CG) method [3].

Theoretically, for any SPD matrix A, CG will converge to the exact solution in 'n' iterations. In practice, however, convergence is achieved in much less number of iterations given that a 6-10 order of magnitude drop in the relative error is generally sufficient. The method of conjugate gradient belongs to a set of iterative methods for the solution of arbitrary matrices, called accelerators or Krylov subspace methods [86].

In any iterative method, reducing the number of iterations leading to a converged solution not only depends on choosing a proper solution method, but also to a large extent on the conditioning of the coefficient matrix. The conditioning of a matrix, directly related to its eigenvalue distribution, is an indication of the error created in the results when operations with the matrix are involved. In any computer, arithmetic operations, which are performed at a discrete level, represent a perturbation since they are not exact. So, if a matrix is well-conditioned, then the error in computations such as multiplication by a vector will be minimal, whereas for an ill-conditioned matrix these errors will be magnified and contaminate the solution. Given the fact that matrix-vector products are at the heart of any iterative algorithm, it becomes evident that conditioning will have a direct effect on error magnification and propagation, and therefore on the speed of convergence.

To represent the conditioning of a matrix A, a metric called the condition number is used. It is defined as

$$\kappa(A) = \|A\|\|A^{-1}\|, \tag{4.2}$$

where $\|\cdot\|$ is some matrix norm. This definition, however, is almost never used since it requires evaluation of the inverse matrix, which is a costly operation. Instead, the ratio of the largest to smallest singular value of A is used,

$$\kappa(A) = \frac{\sigma_{max}}{\sigma_{min}}, \tag{4.3}$$

[1]Matrix A is called positive definite, if for any nonzero vector x we have $x^T Ax > 0$, which translates into all its eigenvalues being positive real.

where the singular values are the square roots of the eigenvalues of $A^H A$, where A^H is the conjugate transpose. In the case of a symmetric matrix, $A^H = A^T$, and singular values are identical to the eigenvalues of A.

For any matrix A, the condition number $\kappa(A) \geq 1$, where the equality holds when A is the identity matrix. As a rule of thumb, if $\kappa(A) = \mathcal{O}(10^t)$, then t digits of accuracy may be lost during computations involving A. In another words, the condition number determines the maximum possible loss in accuracy due to the round-off errors for calculations involving A.

Given the significant adverse effect the ill-conditioning of a matrix can have on the convergence rate of any iterative method, it is often recommended or even necessary to use preconditioning as a way of alleviating this effect. Preconditioning refers to a procedure in which a *different yet equivalent* system of equations

$$\tilde{A}u = \tilde{f}, \tag{4.4}$$

is solved, such that $\kappa(\tilde{A}) \ll \kappa(A)$. This can be achieved, for example, by premultiplying Eq.(4.1) by the matrix M^{-1}, where M, called the *preconditioner*, is a matrix that is easily invertible, and such that M^{-1} approaches A^{-1} in some sense. Therefore, in (4.4) we have $\tilde{A} = M^{-1}A$ and $\tilde{f} = M^{-1}f$. The simplest example of M is a diagonal preconditioner, i.e., $M = \text{Diag}(A)$. In the extreme case when $M = A$, it is clear that $\kappa(M^{-1}A) = 1$, and any iteration applied to (4.4) will converge in only one iteration. Preconditioning can also be applied by multiplying (4.1) by M^{-1} from the right, or from both the left and the right (split preconditioning).

To solve the mean flow problem, a preconditioned CG method is used. Even though preconditioning will increase the computational cost of every iteration, the overall savings due to smaller number of iterations will offset the increased cost. Given that the solution is sought in the context of a domain decomposition method, it seems natural to choose a method for building the preconditioner that takes advantage of this fact by using, e.g., the information available in each subdomain to build a preconditioner for that subdomain. This approach has the advantage that much of the work in building the preconditioner will

be performed locally, therefore minimizing the expensive global communication between processors.

To explain the preconditioning process used for the solution of the mean flow problem, solution of the system

$$Au = f, \tag{4.5}$$

in the simple two-domain region (Fig. 3.6) is considered, assuming that grids in the overlap region match each other, so that interpolation from one grid to another will not be needed. Using the alternating Schwarz algorithm presented in (3.3)-(3.4), assuming that the nodes on both sides of the overlap region have no direct coupling, leads to a two-step iterative method [89]

$$u_{\Omega_1}^{k+1} = u_{\Omega_1}^{k} + A_{\Omega_1}^{-1}(f_1 - A_{\Omega_1} u_{\Omega_1}^{k} - A_{\Omega\setminus\Omega_1} u_{\Omega\setminus\Omega_1}^{k}), \tag{4.6}$$

$$u_{\Omega_2}^{k+1} = u_{\Omega_2}^{k} + A_{\Omega_2}^{-1}(f_2 - A_{\Omega\setminus\Omega_2} u_{\Omega\setminus\Omega_2}^{k+1} - A_{\Omega_2} u_{\Omega_2}^{k}), \tag{4.7}$$

in which Ω, Ω_1, and Ω_2 represent the interior nodes of their corresponding domains only, with the contribution from the boundary conditions included in f_1 and f_2. Iterations (4.6) and (4.7) can be written in a more compact form, if at each step the values of u that are not changing in that iteration are also included. Doing so changes the iteration to the following form

$$u^{k+1/2} = u^{k} + \begin{bmatrix} A_{\Omega_1}^{-1} & 0 \\ 0 & 0 \end{bmatrix} (f - Au^{k}), \tag{4.8}$$

$$u^{k+1} = u^{k+1/2} + \begin{bmatrix} 0 & 0 \\ 0 & A_{\Omega_2}^{-1} \end{bmatrix} (f - Au^{k+1/2}), \tag{4.9}$$

which can be interpreted as a block Gauss-Siedel method with an overlap.

We now introduce the restriction matrix R – a rectangular matrix containing only zeros and ones (Boolean operator) – that maps between the local and the global variables. That is, we have

$$u_{\Omega_1} = R_1 u, \qquad u_{\Omega_2} = R_2 u, \qquad \text{and} \qquad A_{\Omega_i} = R_i A R_i^T. \tag{4.10}$$

The effect of R^T, called the prolongation matrix, is a reverse mapping from the local to the global level, or the familiar assembly process in the finite element method. In parallel computing, the actions of R and R^T on a vector are that of scatter and gather operations, respectively. Using this notation, we can write (4.8) and (4.9) in the following compact form

$$u^{k+1/2} = u^k + B_1(f - Au^k), \tag{4.11}$$

$$u^{k+1} = u^{k+1/2} + B_2(f - Au^{k+1/2}), \tag{4.12}$$

where $B_i = R_i^T(R_iAR_i^T)^{-1}R_i$. Therefore, the application of B_i to a vector v, B_iv, involves a restriction to the local values ($v_i = R_iv$), solution in the local subdomain ($w_i = A_{\Omega_i}^{-1}v_i$), and a transformation back to the global level ($w = R_i^Tw_i$). This two-step method can now be easily combined into a single-step method by eliminating $u^{n+1/2}$ to yield

$$u^{k+1} = u^k + (B_1 + B_2 + B_2AB_1)(f - Au^k). \tag{4.13}$$

Equation (4.13) is called the *multiplicative Schwarz method*[2] applied to a two-domain region. A closer look reveals that this iteration is indeed a Richardson iteration with the preconditioning matrix given by

$$M^{-1} = B_1 + B_2 + B_2AB_1. \tag{4.14}$$

Due to the presence of the B_2AB_1 term, this method does not provide a symmetric preconditioning and also is not fully parallel (although one can remedy this by applying the preconditioner in the reverse order in alternate iterations). Both of these issues can be resolved, if one uses the simplified (and less accurate) preconditioner

$$M^{-1} = B_1 + B_2, \tag{4.15}$$

which is called the *additive Schwarz method* (ASM) [16]. In the case of two subdomains,

[2]The term multiplicative refers to the fact that B can be written in terms of the products of the $(I - B_iA)$ operators, $B = [I - (I - B_1A)(I - B_2A)]A^{-1}$.

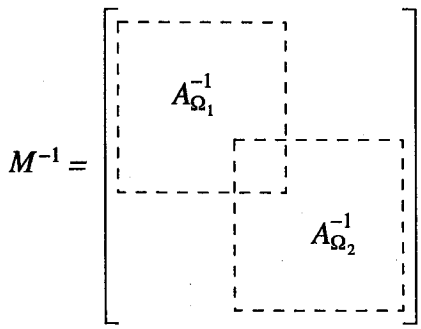

Figure 4.1: Additive Schwarz preconditioner for two subdomains.

the ASM can be written as

$$u^{k+1} = u^k + \underbrace{\left(\begin{bmatrix} A_{\Omega_1}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & A_{\Omega_2}^{-1} \end{bmatrix} \right)}_{M^{-1}} (f - Au^k), \tag{4.16}$$

which is simply a Richardson iteration with overlapping block Jacobi applied as a preconditioner (Fig. 4.1). Generalization of the additive Schwarz method to many subdomains can be easily done by simply setting

$$M^{-1} = \sum_i B_i. \tag{4.17}$$

Due to the symmetry of the system of equations (2.133) and (2.140), the additive Schwarz method provides a very suitable preconditioner for the solution of the mean flow problem. This is a preconditioner that is easy and relatively inexpensive to build, preserves symmetry, is fully parallelizable, and therefore very well suited for the solution of problems using the domain decomposition method. Development of the above preconditioner using the Richardson iteration was only used for demonstration purposes. In solving the mean flow problem, however, instead of using (4.16), which is a fixed-point iteration and known to have a very slow rate of convergence, the more optimum Conjugate Gradient method is used.

Implementation of the additive Schwarz preconditioner and subsequently solving the system of equations in parallel using the CG method is done using the PETSc library

[4], with the information required for setting up the local domains A_{Ω_i} provided by the ParMETIS [47].

4.2 Solution Method for the Acoustic Problem

The system of equations obtained through the assembly of matrices in (2.54) and (2.127) is a complex-valued system that unlike its mean flow counterpart is neither symmetric nor definite. Moreover, the system is ill-conditioned due to the choice of spectral method for discretization. An efficient approach for solving a large system of equations, of the order of $\mathcal{O}(10^5)$ and more, is an iterative method. Direct solution methods, even in parallel, will be at least several times more expensive, both in terms of memory usage and computation time. However, iterative solution of the current acoustic problem still poses a challenge given the ill-conditioning, nonsymmety and the indefiniteness of the global matrix.

Among different approaches used by other researchers for the solution of systems with similar properties [27, 63], we have chosen the Schur complement method. As briefly explained earlier in § 3.3.2, it is a nonoverlapping method based on domain decomposition, where solution of the original problem reduces to the solution of unknowns on interior boundaries of subdomains, which is usually 2-3 orders of magnitude smaller than the original matrix, followed by a solution in each subdomain. This method is appealing because it is very well suited for parallel implementation on distributed memory machines. Moreover, the solution of the much smaller Schur system would be easier, for which a new type of preconditioning is proposed.

4.2.1 Schur Complement Method

First, we explain the method for two subdomains and later discuss its extension to arbitrary number of subdomains. Consider Fig. 4.2, where domain Ω is subdivided into two nonoverlapping subdomains Ω_1 and Ω_2, with the proper boundary conditions applied on the exterior boundary, Γ. Denoting by I and B interior and interface boundary points,

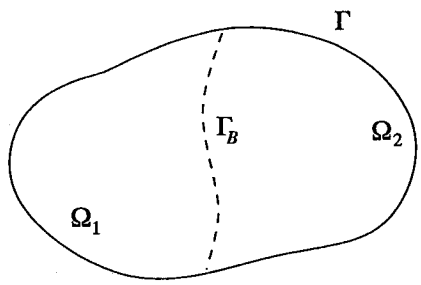

Figure 4.2: Partitioning for two subdomains.

respectively, the element matrix for subdomain 'i' can be written as

$$A^{(i)} = \left[\begin{array}{c|c} A_{II}^{(i)} & A_{IB}^{(i)} \\ \hline A_{BI}^{(i)} & A_{BB}^{(i)} \end{array}\right], \tag{4.18}$$

where $A_{II}^{(i)}$ is the submatrix containing contribution of interior nodes on each other, $A_{IB}^{(i)}$ the submatrix containing contribution of interior nodes on boundary nodes, and so on. Rearranging the variables and the right-hand side vector in a similar manner, i.e. interior nodes followed by boundary nodes, we have in subdomain 'i'

$$A^{(i)}u^{(i)} = f^{(i)}, \quad \text{where} \quad u^{(i)} = \{u_I^{(i)}, u_B^{(i)}\}^T, \quad \text{and} \quad f^{(i)} = \{f_I^{(i)}, f_B^{(i)}\}^T. \tag{4.19}$$

Assembling equations for subdomains (1) and (2) yields

$$\begin{bmatrix} A_{II}^{(1)} & & A_{IB}^{(1)} \\ & A_{II}^{(2)} & A_{IB}^{(2)} \\ A_{BI}^{(1)} & A_{BI}^{(2)} & A_{BB}^{(1)} + A_{BB}^{(2)} \end{bmatrix} \begin{Bmatrix} u_I^{(1)} \\ u_I^{(2)} \\ u_B \end{Bmatrix} = \begin{Bmatrix} f_I^{(1)} \\ f_I^{(2)} \\ f_B^{(1)} + f_B^{(2)} \end{Bmatrix}, \tag{4.20}$$

which can be solved for boundary values u_B using block Gaussian elimination

$$\begin{bmatrix} A_{II}^{(1)} & & A_{IB}^{(1)} \\ & A_{II}^{(2)} & A_{IB}^{(2)} \\ 0 & 0 & S \end{bmatrix} \begin{Bmatrix} u_I^{(1)} \\ u_I^{(2)} \\ u_B \end{Bmatrix} = \begin{Bmatrix} f_I^{(1)} \\ f_I^{(2)} \\ g \end{Bmatrix}. \tag{4.21}$$

The matrix S thus obtained is called the *Schur* matrix and the corresponding system

$$Su_B = g, \tag{4.22}$$

is called the reduced Schur system. The Schur matrix is given by the sum of the Schur components of individual subdomains, S_1 and S_2,

$$S = S_1 + S_2, \quad \text{where} \qquad S_i = A_{BB}^{(i)} - A_{BI}^{(i)} A_{II}^{(i)^{-1}} A_{IB}^{(i)}, \tag{4.23}$$

and similarly for the right-hand side vector

$$g = g^{(1)} + g^{(2)}, \quad \text{where} \qquad g^{(i)} = f_B^{(i)} - A_{BI}^{(i)} A_{II}^{(i)^{-1}} f_I^{(i)}. \tag{4.24}$$

Once the Schur system is solved and values at interface boundaries, u_B, are known, one can easily obtain the solution within each subdomain by back substitution using (4.20)

$$A_{II}^{(i)} u_I^{(i)} + A_{IB}^{(i)} u_B = f_I^{(i)}, \quad \text{or} \qquad u_I^{(i)} = A_{II}^{(i)^{-1}} (f_I^{(i)} - A_{IB}^{(i)} u_B), \quad i = 1, 2. \tag{4.25}$$

It is clear from (4.25) that solution of interior variables is done independently within each subdomain and with no interaction between them.

Let us now assume that computational domain Ω is subdivided into p disjoint subdomains Ω_i, i.e. $\Omega = \bigcup_{i=1}^{p} \Omega_i$, and $\bigcap_{i=1}^{p} \Omega_i = \varnothing$. Therefore, the global matrix can be written in the following block form

$$\begin{bmatrix} A_{II}^{(1)} & & & & A_{IB}^{(1)} \\ & A_{II}^{(2)} & & & A_{IB}^{(2)} \\ & & \ddots & & \vdots \\ & & & A_{II}^{(i)} & A_{IB}^{(i)} \\ A_{BI}^{(1)} & A_{BI}^{(1)} & \dots & A_{BI}^{(i)} & \bar{\Sigma} A_{BB}^{(i)} \end{bmatrix} \begin{Bmatrix} u_I^{(1)} \\ u_I^{(2)} \\ \vdots \\ u_I^{(i)} \\ u_B \end{Bmatrix} = \begin{Bmatrix} f_I^{(1)} \\ f_I^{(2)} \\ \vdots \\ f_I^{(i)} \\ \bar{\Sigma} f_B^{(i)} \end{Bmatrix}, \tag{4.26}$$

in which $\bar{\Sigma}$ indicates assembly. Using block Gaussian elimination, as before, the above

system can be reduced to

$$\begin{bmatrix} A_{II}^{(1)} & & & A_{IB}^{(1)} \\ & A_{II}^{(2)} & & A_{IB}^{(2)} \\ & & \ddots & \vdots \\ & & A_{II}^{(i)} & A_{IB}^{(i)} \\ & & & S \end{bmatrix} \begin{Bmatrix} u_I^{(1)} \\ u_I^{(2)} \\ \vdots \\ u_I^{(i)} \\ u_B \end{Bmatrix} = \begin{Bmatrix} f_I^{(1)} \\ f_I^{(2)} \\ \vdots \\ f_I^{(i)} \\ g \end{Bmatrix}, \tag{4.27}$$

where the last equation forms the Schur system.

Define $\tilde{R}_i$ as the restriction matrix which maps global boundary points to boundary points on subdomain 'i',

$$u_B^{(i)} = \tilde{R}_i u_B. \tag{4.28}$$

$\tilde{R}_i$ is a rectangular matrix of size $n_B^{(i)} \times n_B$, containing only of zero's and one's, with $n_B^{(i)}$ being the number of interface boundary points in subdomain 'i', and n_B the total number of interface boundary unknowns in Ω. Using this operator, the Schur system in (4.27) can be written as

$$Su_B = g, \quad \text{where} \quad S = \sum_{i=1}^{p} \tilde{R}_i^T S_i \tilde{R}_i, \quad \text{and} \quad g = \sum_{i=1}^{p} \tilde{R}_i^T g^{(i)}, \tag{4.29}$$

with $S^{(i)}$ and $g^{(i)}$ defined in (4.23) and (4.24). Once the Schur system is solved, unknown variables within each subdomain can be calculated

$$u_I^{(i)} = A_{II}^{(i)^{-1}} (f_I^{(i)} - A_{IB}^{(i)} u_B^{(i)}), \quad i = 1, 2, \ldots, p. \tag{4.30}$$

At this point, the solution to the original system $u = \{u_I^{(1)}, u_I^{(2)}, \ldots, u_I^{(i)}, u_B\}^T$ will be obtained. Equations (4.29) and (4.30) along with definitions in (4.23) and (4.24) provide the mathematical formulation for solution of the systems of equations using the Schur complement method. Details of implementation using these equations are presented at the end of this chapter.

It is worth mentioning that one could apply the Schur complement method repeatedly, in what is called a multi-level domain decomposition method. That is, S, could be used to

build a lower-level Schur system by further partitioning of subdomains. After applying this method several times, the Schur matrix of the last level will be small enough to be solved using a direct method or an iterative method with a robust preconditioner. Marching backwards, one can then obtain the solution to the original system. A version of this method, called recursive multilevel solver, is discussed in detail by Li and Saad [58], and is publicly available as the "pARMS" package.

4.2.2 Preconditioning

Solution of the reduced Schur system is expected to be easier than the original matrix due to the fact that it is denser and generally better conditioned, especially for elliptic differential operators. Solving this smaller system, however, still poses a challenge because the Schur matrix implicitly contains the couplings between various subdomains and therefore inherits the negative properties of the original system. Preconditioning, therefore, is considered an essential step, which will reduce the solution time (iteration count) and consequently offset the extra cost associated with reformulating the original problem into the Schur complement form.

The task of building a suitable preconditioner for the Schur matrix is somewhat complicated by the fact that S is almost never formed explicitly due to the high cost. The many different preconditioners proposed by researchers, therefore, rely on the Schur complement matrices of individual subdomains or their components [79, 103].

Here, we discuss the Neumann-Neumann method (NN) [57], which forms the basis for our proposed preconditioner. This method relies on the simple observation that a suitable preconditioner M for the Schur matrix could be built by assembling the inverse Schur complements of individual subdomains. That is,

$$M^{-1} = \sum_{i=1}^{p} \tilde{R}_i^T S_i^{-1} \tilde{R}_i, \tag{4.31}$$

where p is the total number of subdomains.

For a better convergence, Schur complements of individual subdomains are weighted

by a diagonal matrix $\tilde{D}_i$ of size $n_B^{(i)} \times n_B^{(i)}$, whose components, $d_{kk}^{(i)}$, are the inverse of the number of subdomains shared by node k on interior boundary Ω_i. As a result, weight matrices satisfy the relation

$$\sum_{i=1}^{p} \tilde{R}_i^T \tilde{D}_i \tilde{R}_i = I, \tag{4.32}$$

where I is the identity matrix. So, the final form of the Neumann-Neumann preconditioner is

$$M_{NN}{}^{-1} = \sum_{i=1}^{p} \tilde{R}_i^T \tilde{D}_i S_i^{-1} \tilde{D}_i \tilde{R}_i. \tag{4.33}$$

In applying (4.33), the action of S_i^{-1} on a vector is calculated by solving a system involving $A^{(i)}$ and without explicitly forming S_i^{-1} [89].

The Neumann-Neumann preconditioner derives its name from the fact that at each iteration, 'p' Neumann problems has to be solved, one in each subdomain. That is, the action of S_i^{-1} amounts to solving a discrete problem in Ω_i with Neumann boundary conditions applied on its interface. Consequently, if Ω_i is an interior subdomain, the Neumann-Neumann preconditioner will fail because it entails solving of a problem which is not well-posed because all boundary conditions are of Neumann type. The remedy is to apply a pseudo-inverse, whose accuracy will strongly affect the convergence of the Schur System. On the positive side, the Neumann-Neumann method has the favorable characteristic that can be easily extended to multi-dimensional domains and unstructured meshes without any special modifications.

In domain decomposition methods, the condition number of the preconditioned system is in general a function of the characteristic size of subdomains (H), characteristic element size (h), and coefficients of the PDE (more specifically, their variation along subdomain boundaries). An optimal preconditioner is, therefore, one that is relatively insensitive to variations of the above parameters. For the NN method, the condition number for an elliptic operator is given by [57]

$$\kappa(M_{NN}^{-1} S) \leq C\Big(1 + \log\frac{H}{h}\Big)^2 / H^2, \tag{4.34}$$

in which C is a positive constant. This bound clearly shows that convergence rate of the

NN method strongly depends on the relative size of subdomains and so will rapidly deteriorate (not scalable) as the number of subdomains are increased, i.e. as $H \rightarrow 0$. This deterioration is to be expected because in the NN method the only mechanism for propagation of information between subdomains is through their mutual boundaries. Increasing the number of subdomains will lead to interface boundaries that are only remotely attached, hence convergence slowdown.

To resolve this problem and improve scalability, Mandel [68] proposed the balancing Neumann-Neumann (BNN) method by adding a coarse grid correction to (4.33). This correction resulted in faster communication between subdomains and led to a better convergence rate, as reflected in the new bound for the condition number

$$\kappa(M_{BNN}^{-1}S) \leq C\Big(1+\log\frac{H}{h}\Big)^2, \tag{4.35}$$

where dependency on the number of subdomains has been greatly reduced due to elimination of the $1/H^2$ term.

Using a 3-step Richardson iteration, the BNN method is as follows

$$\begin{aligned}
u_B^{k+1/3} &\longleftarrow u_B^k + (\tilde{R}_0^T S_0^{-1}\tilde{R}_0)(g - Su_B^k),\\
u_B^{k+2/3} &\longleftarrow u_B^{k+1/3} + \Big(\sum_{i=1}^{p} \tilde{R}_i^T \tilde{D}_i S_i^{-1} \tilde{D}_i \tilde{R}_i\Big)(g - Su_B^{k+1/3}),\\
u_B^{k+1} &\longleftarrow u_B^{k+2/3} + (\tilde{R}_0^T S_0^{-1}\tilde{R}_0)(g - Su_B^{k+2/3}),
\end{aligned} \tag{4.36}$$

where preconditioning is balanced by applying the coarse grid correction in both the first and last steps. The restriction operator $\tilde{R}_0$ is a rectangular matrix of size $p \times n_B$, which when multiplied by the global boundary vector u_B returns the weighted sum of boundary variables for each subdomain. Components of row 'i' of $\tilde{R}_0$, i.e. weights, are therefore the inverse of the number of subdomains each point on interface boundary of Ω_i shares. So, $\tilde{R}_0$ can be easily built by properly assembling the diagonal of the weight matrices, $\tilde{D}_i$. Having built $\tilde{R}_0$, the $p \times p$ coarse grid matrix, S_0, is obtained from

$$S_0 = \tilde{R}_0 S \tilde{R}_0^T. \tag{4.37}$$

The action of the coarse grid correction can therefore be interpreted as removal of the constant part of the low-frequency error from each subdomain at each iteration.

Denoting by B_0 and B_{NN} the coarse grid and Neumann-Neumann preconditioner operators, respectively,

$$B_0 = \tilde{R}_0^T S_0^{-1} \tilde{R}_0, \qquad \text{and} \qquad B_{NN} = \sum_{i=1}^{p} \tilde{R}_i^T \tilde{D}_i S_i^{-1} \tilde{D}_i \tilde{R}_i, \tag{4.38}$$

the balancing Neumann-Neumann method can be simplified to a one-step formulation

$$M_{BNN}{}^{-1} = (I - B_0 S) B_{NN} (I - S B_0) + B_0. \tag{4.39}$$

The BNN is considered a two-level preconditioner. Single-level methods, such as the Neumann-Neumann are efficient only for a small number of subdomains. When the number of subdomains increases, multi-level preconditioners provide a better convergence rate because they eliminate or weaken dependency of the condition number of the preconditioned system on the number of subdomains. As a result, the extra cost associated with adding extra levels is properly offset by significant reduction in the number of iterations and also by improved robustness.

4.2.3 Proposed Preconditioner

The method proposed here for preconditioning the Schur system is a modified version of the balancing Neumann-Neumann method. First, mathematical formulation of the proposed approach is discussed followed by some numerical experiments to show its suitability.

Recall that the Schur complement for subdomain 'i' is

$$S_i = A_{BB}^{(i)} - A_{BI}^{(i)} A_{II}^{(i)^{-1}} A_{IB}^{(i)}. \tag{4.40}$$

Ignoring the last term on the right side of the equation, one can approximate S_i with the local matrix of interface boundary points. That is,

$$S_i \approx A_{BB}^{(i)}. \tag{4.41}$$

This simplification will eliminate, at each application of preconditioning, two matrix-vector multiplications involving $A_{BI}^{(i)}$ and $A_{BI}^{(i)}$, and one (costly) matrix solve involving A_{II}. Using (4.41), the modified NN (MNN) preconditioner operator is defined as

$$B_{MNN} = \sum_{i=1}^{p} \tilde{R}_i^T \tilde{D}_i {A_{BB}^{(i)}}^{-1} \tilde{D}_i \tilde{R}_i. \tag{4.42}$$

One would expect that, similar to the Neumann-Neumann preconditioner, convergence rate of the above preconditioner will worsen with an increase in the number of subdomains. So, following the same line of thought, one could combine the above preconditioner with a coarse grid solve.

It should be pointed out that in the BNN method, Eq. (4.36), the last coarse grid preconditioner was applied to preserve the symmetry of the Schur system. So, for nonsymmetric problems, such as the acoustic problem, one can simply eliminate the last step (or the first step). In such a case, (4.39) simplifies to

$$M^{-1} = B_{NN}(I - SB_0) + B_0, \tag{4.43}$$

which is a multiplicative preconditioner. The equivalent preconditioner, using the proposed modified operator B_{MNN}, is

$$M^{-1} = B_{MNN}(I - SB_0) + B_0 = B_{MNN} + B_0 - B_{MNN}SB_0. \tag{4.44}$$

Eliminating the last term leads to a preconditioner of additive type

$$M^{-1} = \sum_{i=1}^{p} \tilde{R}_i^T \tilde{D}_i {A_{BB}^{(i)}}^{-1} \tilde{D}_i \tilde{R}_i + \tilde{R}_0^T S_0^{-1} \tilde{R}_0 = B_{MNN} + B_0. \tag{4.45}$$

So, the preconditioning will consist of a local operation within each subdomain represented by B_{MNN}, followed by an operation involving B_0 at the global level. Tests were conducted using both the multiplicative form, (4.44), and the additive form, (4.45). The results indicated a very negligible difference between the two approaches in terms of the rate of convergence. This is somewhat expected because B_{MNN} is only an approximation

to the original operator B_{NN}, and unlike exact operators where multiplicative form outperforms the additive type, the changes here were insignificant. Therefore, for subsequent calculations only the additive form was used.

The method in 4.45, being a modified version of the balancing Neumann-Neumann method possesses many of its desirable features. This includes straightforward extension to three dimensions and its applicability to irregular domains and unstructured meshes. A closer look at (4.45) also shows that the most expensive part of the preconditioning, i.e. application of the B_{MNN} operator, happens at the local level (assuming that computational domain is subdivided such that each partition is assigned to only one processor). This means interprocessor communications, which are the most expensive type, will be minimized during the bulk of preconditioning step. The only part of (4.45) that requires global communication is application of the coarse grid operator, B_0. Given that S_0 is only a matrix of size p, i.e. the number of subdomain partitions, applying a coarse grid preconditioner will require global communication of the order $\mathcal{O}(p^2)$. This is extremely small compared to the dominant global message passing operation (scatter/gather) for vectors of size n_B, which occurs several times at each iteration.

However, as will be shown later in § 4.2.7, it was found that the coarse grid component B_0 in (4.45) had essentially no effect on the convergence rate, and hence the number of iterations. It was expected that for a large number of subdomains, e.g. $p > 10$, addition of the coarse grid preconditioner would be necessary otherwise the convergence would deteriorate. However, the results for even a large number of subdomains, i.e. $p = 32$, showed that the preconditioner as defined in (4.42) performed as well as its counterpart with the coarse grid component, Eq. (4.45).

Therefore, all calculations for the numerical test cases were performed with the following preconditioner, which will be referred to as the '*proposed*' preconditioner in the rest of this thesis:

$$\boxed{M_P^{-1} = \sum_{i=1}^{p} \tilde{R}_i^T \tilde{D}_i \, {A_{BB}^{(i)}}^{-1} \tilde{D}_i \tilde{R}_i} \tag{4.46}$$

As mentioned earlier, this preconditioner is a local operator, i.e., is applied on individual subdomains; a very attractive feature. Furthermore, absence of the coarse grid component reduces the computation time by eliminating the global communication associated with it. These factors combined, make the proposed preconditioner a very good choice.

4.2.4 Other Preconditioners

Recall that the Schur matrix is given by

$$S = \sum_{i=1}^{p} \tilde{R}_i^T S_i \tilde{R}_i, \quad \text{where} \qquad S_i = A_{BB}^{(i)} - A_{BI}^{(i)} {A_{II}^{(i)}}^{-1} A_{IB}^{(i)}. \tag{4.47}$$

So, one can write

$$\begin{aligned} S &= \sum_{i=1}^{p} \tilde{R}_i^T \Big(A_{BB}^{(i)} - A_{BI}^{(i)} {A_{II}^{(i)}}^{-1} A_{IB}^{(i)} \Big) \tilde{R}_i, \\ &= \sum_{i=1}^{p} \tilde{R}_i^T A_{BB}^{(i)} \tilde{R}_i - \sum_{i=1}^{p} \tilde{R}_i^T \Big(A_{BI}^{(i)} {A_{II}^{(i)}}^{-1} A_{IB}^{(i)} \Big) \tilde{R}_i, \\ &= A_{BB} - \sum_{i=1}^{p} \tilde{R}_i^T A_{BI}^{(i)} {A_{II}^{(i)}}^{-1} A_{IB}^{(i)} \tilde{R}_i. \end{aligned} \tag{4.48}$$

As an approximation to S, it then seems logical to use A_{BB} as a preconditioner

$$M_B^{-1} = {A_{BB}}^{-1} = \Big(\sum_{i=1}^{p} \tilde{R}_i^T A_{BB}^{(i)} \tilde{R}_i \Big)^{-1}. \tag{4.49}$$

The global matrix of boundary points, A_{BB}, is equal to the B_{MNN} operator for $p = 2$, i.e., two subdomains, but it better approximates S for much larger number of subdomains ($p \gg 2$). However, the disadvantage in using A_{BB} as a preconditioner is that its application requires solution of a matrix at the global level, hence increased interprocessor communication. Also, since the size of A_{BB} is the same as the Schur matrix, each application of the preconditioner using this matrix will be several times more expensive than solving much smaller systems involving $A_{BB}^{(i)}$ within each subdomain. However, due to better convergence rate, use of A_{BB} as a preconditioner could be justified if n_B is not very large, say $n_B \approx \mathcal{O}(10^3)$.

As an alternative, one can use the diagonal part of A_{BB} to build a Jacobi preconditioner

$$M_J^{-1} = \Big(\text{Diag}(A_{BB})\Big)^{-1}. \tag{4.50}$$

While this simple preconditioner could provide a very good convergence for certain problems [12], it was found that for the acoustic problem this method is not robust and iterations could easily stagnate unless an unrealistically large amount of damping is applied [108].

In sections 4.2.6 and 4.2.7, performance of different iterative methods and preconditioners, including the proposed method, with and without the coarse grid solve will be discussed in detail.

It should also be mentioned that we also investigated the use of a *probing* method for preconditioning of the Schur matrix [10]. This method relies on the fact that if the Schur matrix is highly localized, i.e. has its largest components around the diagonal, one can obtain an approximation to its diagonal or tridiagonal part by multiplying it with a set of probing vectors.

In its simplest form, the probing vector is a vector of unity

$$v = \{1, 1, 1, \ldots, 1\}^T. \tag{4.51}$$

When multiplied by S, the result is a vector $d = Sv$, whose components are the row-sum of S. Vector d will be a very good approximation to $\text{Diag}(S)$, if S is highly diagonally dominant[3]. So, at the cost of one matrix-vector multiplication one can find a very good approximation to diagonal of S without explicitly forming it, and then use this approximation to build a Jacobi preconditioner.

Probing techniques have been shown to significantly improve the convergence for elliptic problems, despite being a purely algebraic preconditioner[4] [48]. However, after some experimentation, it was found that this method is not suitable for the acoustic problem as it failed to improve the convergence in any meaningful way (see § 4.2.7).

[3] d will be equal to $\text{Diag}(S)$, only if S is a diagonal matrix itself.

[4] The preconditioned system is not spectrally equivalent to the original system.

N	n_B	$n_I^{(1)}$	$n_I^{(2)}$
2	337	1484	1412
3	727	4773	4617

Table 4.1: Number of interior and boundary nodes in the subdomains.

4.2.5 Numerical Analysis

After explaining the theory, the results are now presented for several numerical experiments conducted using MATLAB to justify the suitability of the proposed preconditioner, Eq. (4.46). The test problem is the acoustic propagation of the plane wave from a zero thickness cylinder in the absence of mean flow. The reduced frequency is $\bar{\omega} = 5.91$, and the number of elements is 320. In order to carry out the analysis in a reasonable amount of time, given that the Schur matrix had to be built explicitly, the number of equations was limited to only a few thousands by performing the calculations for only the quadratic and cubic spectral approximations, i.e., $N = 2, 3$. Also, only the case of two subdomains was considered in order to simplify the assembly of subdomain matrices in MATLAB.

Table 4.1 shows the number of points within each subdomain and along the interface boundary using the notation defined in § 4.2.1. Using the Schur complement method, the block matrices in each subdomain ($A_{II}^{(i)}$, $A_{IB}^{(i)}$, $A_{BI}^{(i)}$, and $A_{BB}^{(i)}$) are used to build the Schur complements, $S_i = A_{BB}^{(i)} - A_{BI}^{(i)} {A_{II}^{(i)}}^{-1} A_{IB}^{(i)}$, $i = 1, 2$.

As an example, Figure 4.3 shows sparsity pattern for individual boundary matrices $A_{BB}^{(1)}$ and $A_{BB}^{(2)}$ and the global boundary matrix $A_{BB} = A_{BB}^{(1)} + A_{BB}^{(2)}$ for $N = 2$. The sparsity pattern for the cubic approximation would be similar. As shown in the figure, the matrices are both sparse and diagonally dominant, but not symmetric. The Schur complements and interior matrices, S_i and $A_{II}^{(i)}$, $i = 1, 2$, are also nonsymmetric, but full matrices (not shown here).

The behavior of the three different preconditioners discussed earlier are now examined. The first two are based on the global boundary matrix A_{BB} and its diagonal, i.e., M_B and

N	$\kappa(S)$	$\kappa(M_J^{-1}S)$	$\kappa(M_B^{-1}S)$	$\kappa(M_P^{-1}S)$
2	60.18	190.61	108.99	252.85
3	124.44	58.64	88.19	350.32

Table 4.2: Condition number of the Schur system before and after preconditioning.

M_J, respectively. The third preconditioner is the one proposed in (4.46) which simplifies to ${A_{BB}^{(1)}}^{-1} S_1 + {A_{BB}^{(2)}}^{-1} S_2$ for the case of two subdomains. As an initial metric assessing the performance of the preconditioners, the condition number for the Schur system before and after preconditioning was calculated. The results are presented in Table 4.2.

The results show that the condition number of the Schur system increases with an increase in the order of approximation, N, as expected. It is also observed that for a quadratic approximation ($N = 2$), preconditioning actually increases the condition number. The results also indicate that in both cases a diagonal preconditioner performs better than the proposed method. Note that condition number only provides a relative measure for determining the efficiency of a preconditioner and using it as the only indicator could be misleading. To make a better judgment, one should examine the eigenvalue distribution of the system before and after preconditioning. Figures 4.4 and 4.5 show such results for the Schur matrix and its preconditioned forms for $N = 2$, 3, respectively. In both cases, the eigenvalues of the Schur matrix all have negative imaginary parts and the majority of them also have negative real parts. However, preconditioning in all three cases has the favorable effect of moving most of the eigenvalues to the positive side of the real axis and away from the zero, with a large number of them being clustered around one.

For $N = 2$, the proposed preconditioner is better than the Jacobi method despite having a larger condition number. This is indicated by the eigenvalues that are less scattered and also mostly located on the positive side of the real axis. The effect of preconditioning for $N = 3$ is similar. Although we have not been able to show the results for higher values of

N, preconditioning is expected to have the same qualitative effect on the eigen distribution of the Schur system. It should be pointed out that despite seemingly better performance of the Jacobi preconditioner with respect to the proposed method, as indicated in the figures, it provides a relatively poor convergence for large values of N, and could easily stagnate unless small number of subdomains are used.

Given that a good preconditioner is one for which the eigenvalues of the preconditioned system are closely clustered (little scattering) and preferably positioned around unity, therefore, given the above results, it can be concluded that the proposed method is well suited for preconditioning of the Schur matrix. This fact is supported by numerical results of the many different test cases presented in Chapter 5.

4.2.6 Solver Analysis

Among different Krylov subspace methods, tests were conducted using the CGS (Conjugate Gradients Squared), GMRES (Generalized Minimal Residual), TFQMR (Transpose-free Quasi-minimal Residual) and Bi-CGSTAB algorithms. The main reason for limiting the tests to these four was that the algorithm of other iterative methods involve a matrix-vector multiplication with the transpose of the coefficient matrix, in addition to a multiplication with the matrix itself. In the current implementation, however, the coefficient matrix (the Schur matrix) is not formed explicitly and so its transpose is not available.

For a typical test case (solution of the plane wave propagation from an unflanged cylinder) convergence history of the above four methods is compared in Figure 4.6. Table 4.3 gives the solution time and the corresponding iteration count for each method. The CGS,

Method	CGS	GMRES	TFQMR	Bi-CGSTAB
Solution Time (min:sec)	15:07	9:29	13:54	15:47
No. of Iterations	190	173	142	139

Table 4.3: Solution time and number of iterations for different algorithms.

TFQMR, and Bi-CGSTAB take much longer to converge compared to the GMRES method. This is because each iteration of these three algorithms involves 2 matrix-vector multiplications as opposed to only one such operation for the GMRES. In terms of the rate of convergence, both Bi-CGSTAB and GMRES exhibit a monotonic behavior as opposed to a very violent and irregular convergence history of the CGS. In fact, Bi-CGSTAB was developed as a smooth and more stable variant of the CGS and Bi-CG algorithms [106]. The TFQMR, also has a smooth convergence with typical plateaus before each drop in the residual norm.

Another issue to be considered in choosing the solver is its sensitivity to the round-off errors. This becomes an even more important issue for ill-conditioned systems, such as the acoustic problem, where the coefficient matrix would magnify small perturbations. To examine this for the three algorithms (TFQMR, Bi-CGSTAB, GMRES), calculations were performed twice using the exact same conditions and the residual history was compared. It was found that the convergence history of both TFQMR and Bi-CGSTAB were only identical in the first few iterations and they started to differ at larger iterations. However, the GMRES method produced exactly the same residuals in both runs. Figure 4.7 shows such a behavior for the TFQMR method (a similar trend was observed for the Bi-CGSTAB). Referring to the figure, while the difference in the iteration count between the two cases was negligible, for much larger test cases and higher orders of approximations the difference could be higher. Given the above discussion, the GMRES method was considered the most suitable algorithm for solution of the acoustic problem.

Theoretically, the full GMRES is guaranteed to converge in n iterations, n being the size of the coefficient matrix, while such a bound does not exist for other methods. Due to memory requirements and computational costs associated with the full GMRES, the restarted version is commonly used. The restart parameter, m, determines the size of the Krylov subspace used for building orthogonal search directions. In most applications, a value for m in the range of $m =$ 20-35 is generally used. However, for problems where the coefficient matrix is not positive definite (such as the acoustic problem), restarted GMRES

could stagnate or lead to a very slow convergence [86]. Figure 5.58 shows the convergence rate for the scarfed inlet test case for two different values of the restart parameter, m. Note the four-fold increase in iteration count (solution time) when m is small.

Given that tests showed that with the proposed preconditioner most problems would converge in about 200-450 iterations, to guarantee a fast convergence rate a restart parameter of $m =$ 400-600 was used for most calculations. Using such a large value for m would obviously demand a larger memory usage. However, given the poor convergence rate associated with the smaller values of the restart parameter, it would be a logical trade-off to use more memory and instead significantly reduce the computation time. Moreover, extra memory usage would not be extraordinarily high given that the size of the Schur matrix is of the order of $\mathcal{O}(10^5)$ for the largest test problem. Also, by comparison, the Schur complement method would require a much larger amount of memory.

4.2.7 Preconditioner Performance

For a sample problem, tests were performed to demonstrate the performance of different preconditioners. Figure 4.8 shows the convergence history. As expected, using the global boundary matrix(A_{BB}) as the preconditioner leads to the fastest convergence, followed by its less accurate approximations, M_P^{-1} and M_J^{-1}, respectively. However, referring to Table 4.4, it is clear that the proposed preconditioner is the best choice in terms of the solution time. It is also a good compromise between the simple Jacobi method M_J^{-1} and the more expensive M_B^{-1} preconditioner. Also shown in Figure 4.8, is the convergence behavior of the probing method. It is clear that that probing technique has actually a negative effect on

Preconditioner	None	M_J^{-1}	M_P^{-1}	M_B^{-1}
Solution Time (min:sec)	10:55	9:39	9:29	14:58
No. of Iterations	261	223	173	129

Table 4.4: Solution time and iteration count for different preconditioners.

the convergence rate as its performance is even worse that the nonpreconditioned system.

Effect of the order of approximation on iteration count is shown in Figure 4.9. The figure confirms the expected effect of N on the rate of convergence, i.e., worsening of the condition number of the original system and consequently the Schur matrix as N increases. A small value of N, however, does not necessarily mean a smaller number of iterations. It is possible to have a poor convergence for small values of N despite relatively low condition number of the matrix. This could happen if the grid resolution is far less than the minimum required to resolve the acoustic waves, leaving the high frequency components of the error trapped in the grid and therefore unresolved.

The effect of adding the coarse grid preconditioner on the convergence rate is shown in Figures 4.10 and 4.11 for the TFQMR and GMRES methods, respectively. As the results indicate, adding the coarse grid component had essentially no effect on the convergence rate. On the other hand, given the coarse grid component of the preconditioner requires communication among subdomains at the global level, its addition would increase the solution time. Therefore, as mentioned earlier, we conclude that the proposed preconditioner without coarse grid component, i.e., Eq. (4.46), is the most appropriate preconditioner to be used.

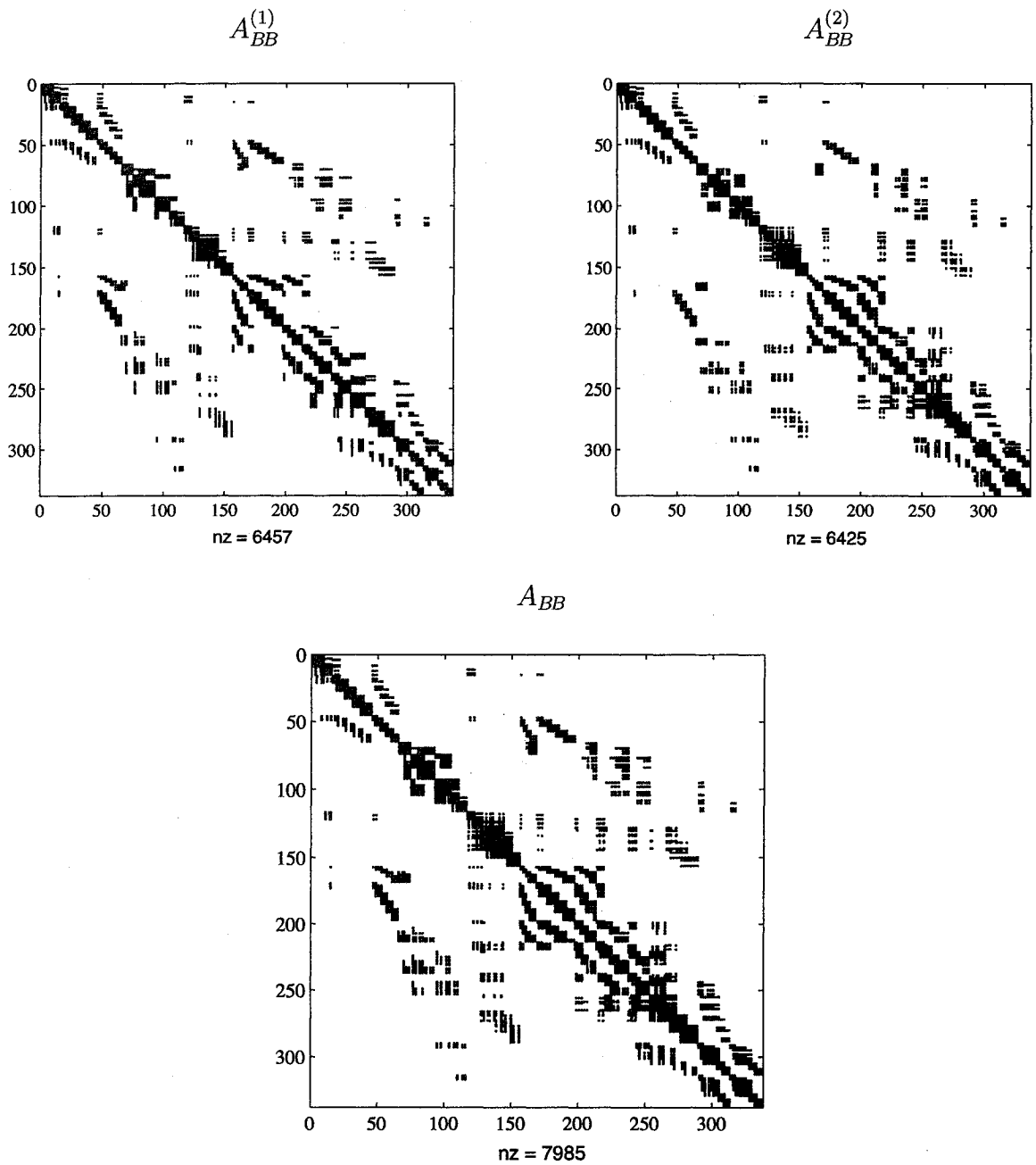

Figure 4.3: Sparsity pattern of boundary matrices; 'nz' is the total number of nonzeros.

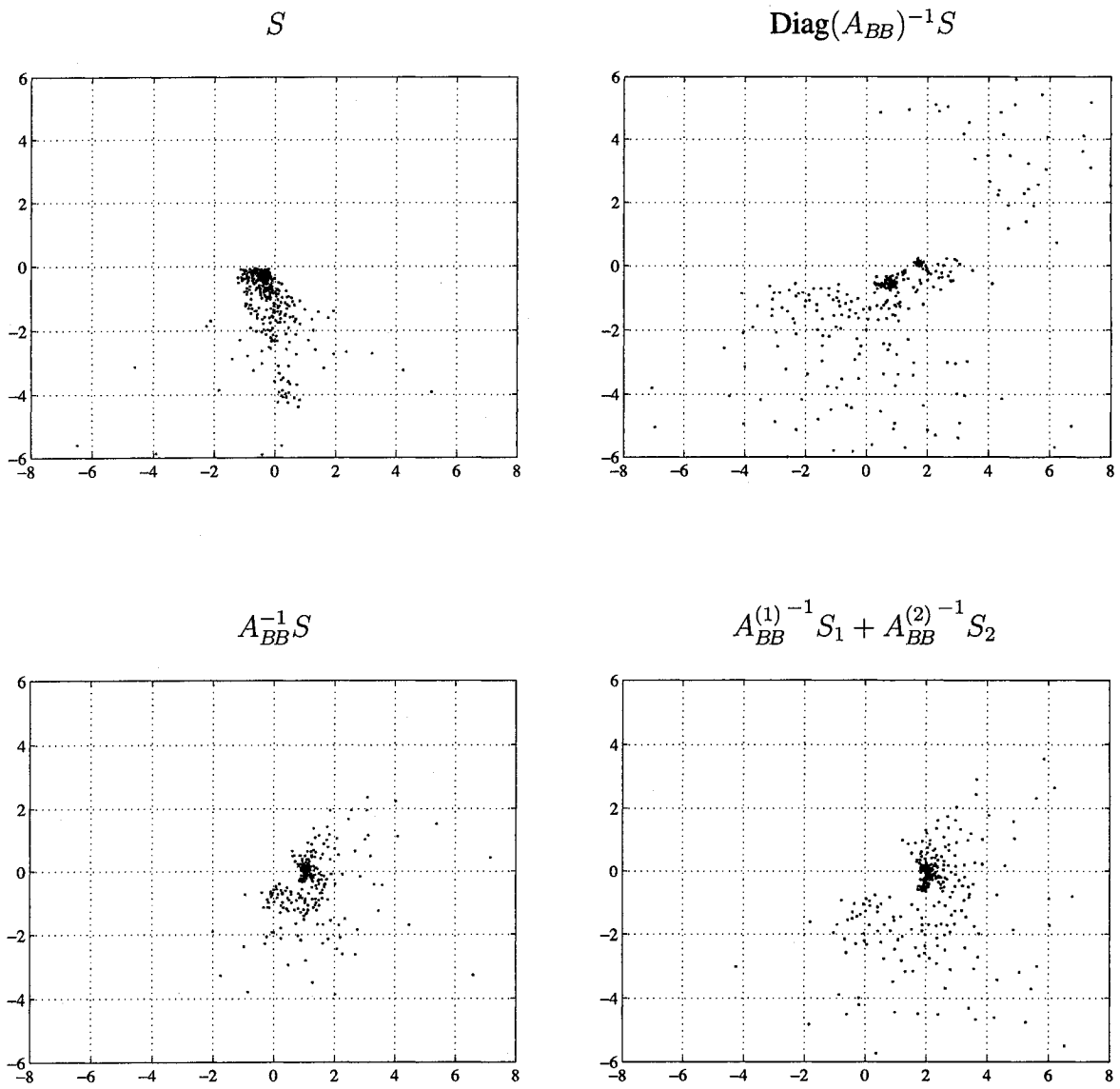

Figure 4.4: Eigenvalue distribution for the Schur matrix, before and after preconditioning; $N = 2$.

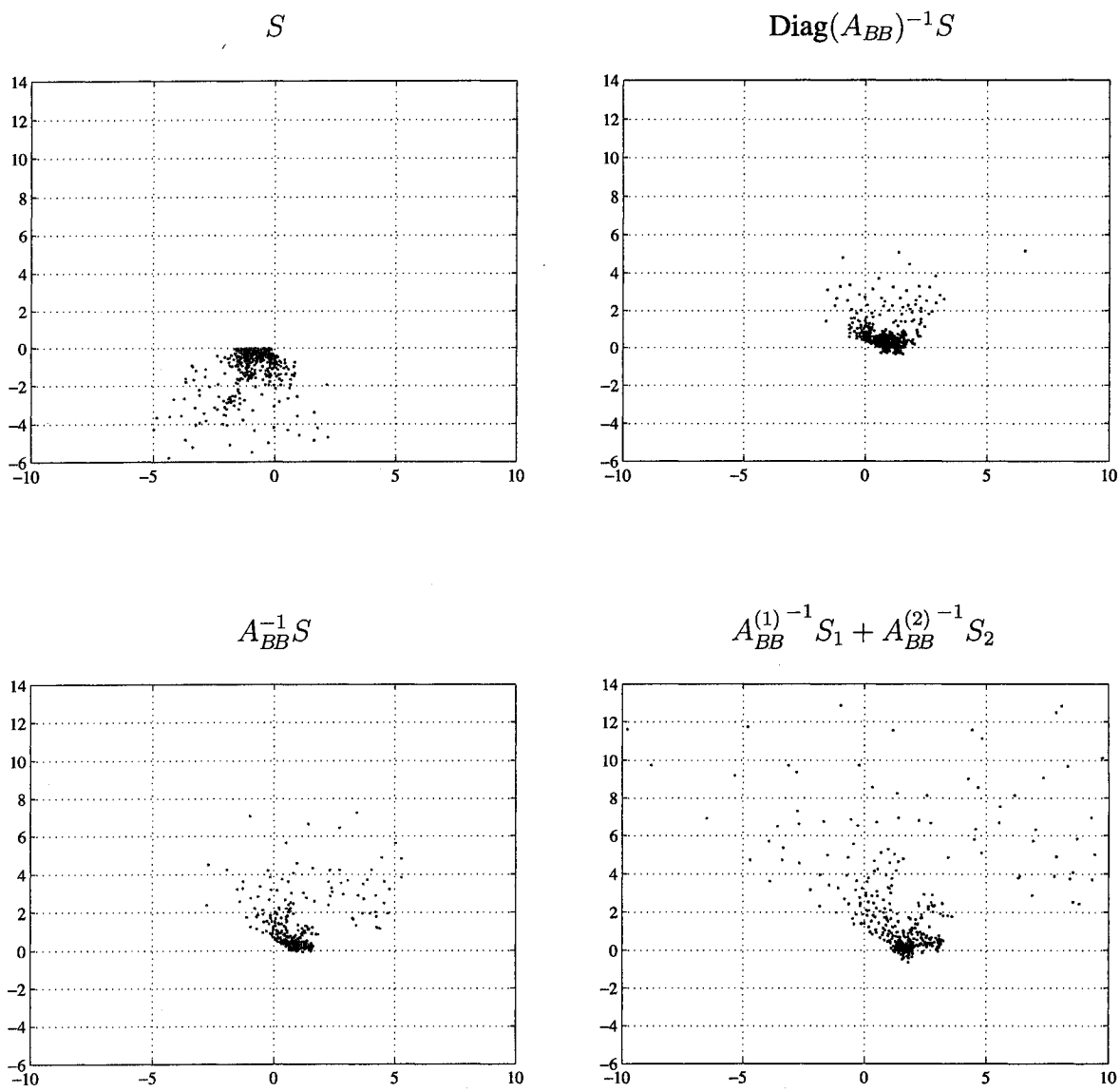

Figure 4.5: Eigenvalue distribution for the Schur matrix, before and after preconditioning; $N = 3$.

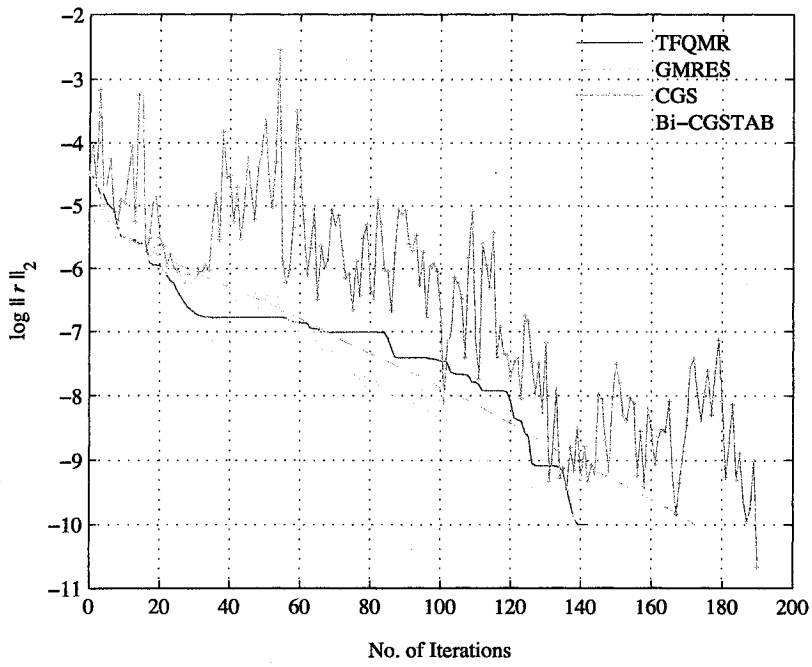

Figure 4.6: Convergence behavior of the Schur matrix using different iterative solvers.

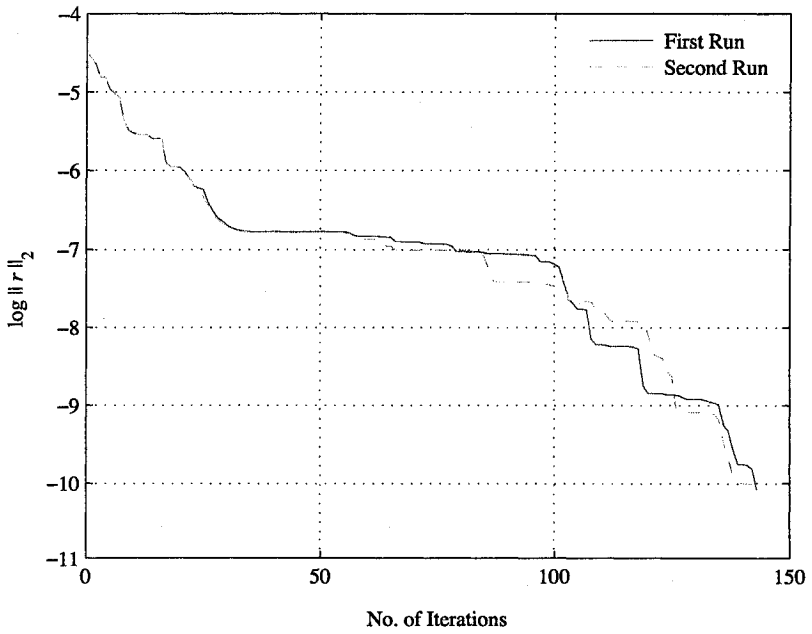

Figure 4.7: Variations in residual of the TFQMR method due to round-off errors.

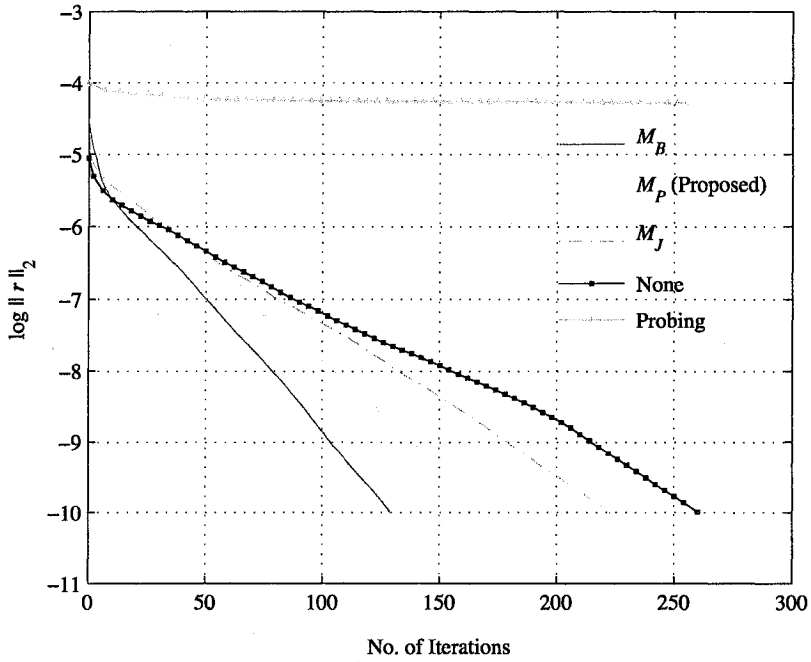

Figure 4.8: Effect of different preconditioners on convergence history of the Schur matrix using the GMRES solver.

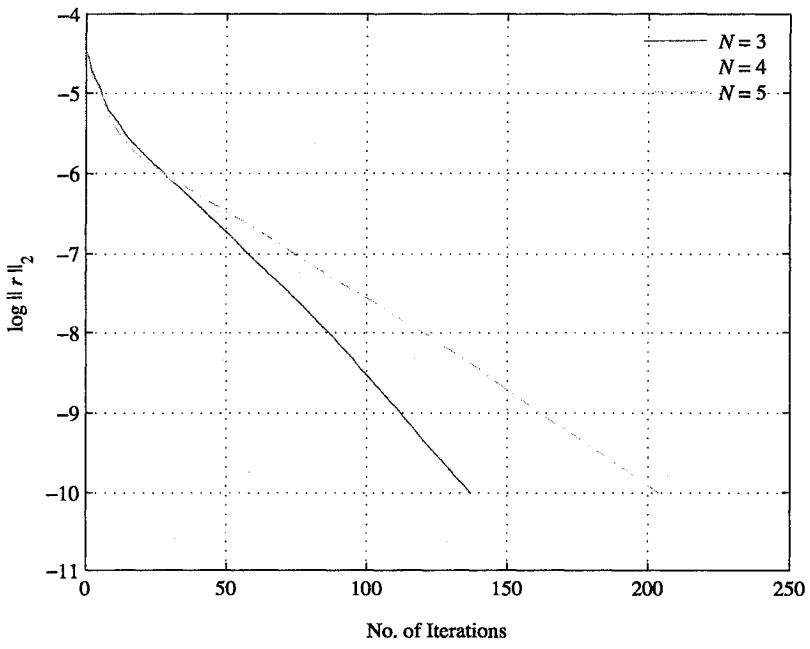

Figure 4.9: Effect of the order of approximation on convergence of the GMRES algorithm..

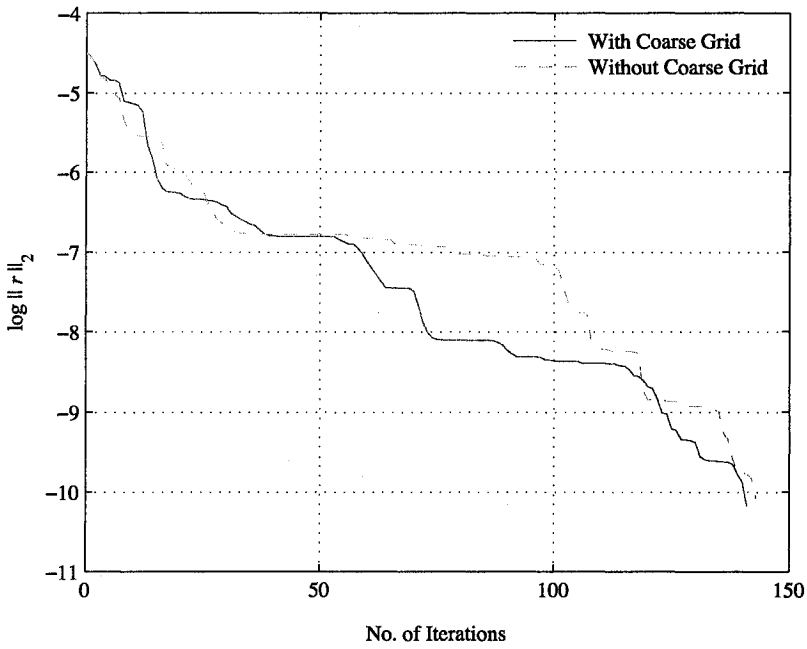

Figure 4.10: Convergence behavior of the preconditioned TFQMR solver with and without the coarse grid preconditioner.

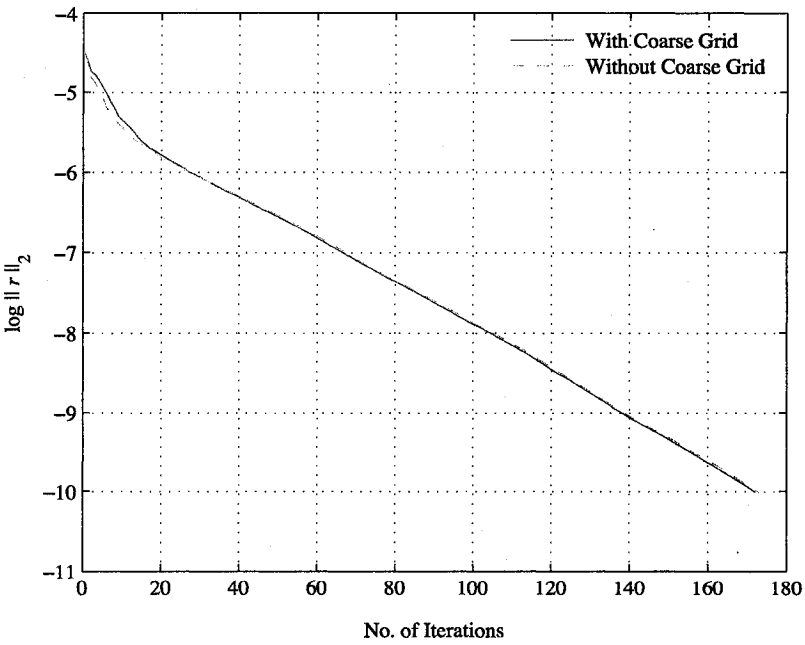

Figure 4.11: Convergence behavior of the preconditioned GMRES solver with and without the coarse grid preconditioner.

4.2.8 Implementation

In this section, details regarding the implementation of the Schur complement method and its solution using an iterative method with the proposed preconditioner is discussed. It should be mentioned that other than simple operations and those related to setup of data structures, which were done in Fortran 90, the bulk of implementation was programmed using various commands of the PETSc library [4]. This included vector and matrix operations, solution of the system of equations at the local and global levels, vector gather/scatter, and local to global mappings and vice versa.

Step 1: Decomposing the domain and building interior/boundary pointers

- Decompose the domain into 'p' subdomains (processors) using the ParMETIS library.
- For each subdomain, build the pointer of interior and interface boundary points.
- Build the global pointer of interface boundary points.

Step 2: Setting up subdomain systems

- For each subdomain, build the local matrix $A^{(i)}$ and the local right-hand side vector $f^{(i)}$, with interior nodes ordered first followed by the interface boundary nodes

$$A^{(i)} = \left[\begin{array}{c|c} A_{II}^{(i)} & A_{IB}^{(i)} \\ \hline A_{BI}^{(i)} & A_{BB}^{(i)} \end{array}\right], \quad \text{and} \quad f^{(i)} = \left\{\begin{array}{c} f_I^{(i)} \\ \hline f_B^{(i)} \end{array}\right\}. \tag{4.52}$$

- Apply boundary conditions in appropriate subdomains[5].
- Build the right-hand side vector[6], $g^{(i)} = f_B^{(i)} - A_{BI}^{(i)} A_{II}^{(i)^{-1}} f_I^{(i)}$:

[5]Note that in subdomains where boundary conditions are applied $A_{BI}^{(i)} \neq A_{IB}^{(i)^T}$

[6]In the following, u and v and their interior/boundary components are dummy vectors.

- solve for $u^{(i)}$: $A_{II}^{(i)} u^{(i)} = f_I^{(i)}$,
- calculate: $v^{(i)} = A_{BI}^{(i)} u^{(i)}$,
- calculate: $g^{(i)} = f_B^{(i)} - v^{(i)}$.

Step 3: Solving the Schur system

As mentioned earlier, creating the Schur matrix explicitly is very expensive. Therefore, solution of the Schur matrix is implemented using the *matrix-free* method. In any iterative method, the only operation which involves the coefficient matrix is in the form of a matrix-vector product, $w = Sv$. This operation can be performed without explicitly building S by using Schur complements of individual subdomains, as explained below.

- Create the scatter (gather) operator $\tilde{R}_i$ ($\tilde{R}_i^T$) using the pointers defined in step 1. So, the action of $\tilde{R}_i$ on a global vector of size n_B will result in the local boundary vector of size $n_B^{(i)}$,

$$u_B^{(i)} = \tilde{R}_i u_B.$$

  Similarly, the action of $\tilde{R}_i^T$ will be that of assembly

$$u_B = \tilde{R}_i^T u_B^{(i)}.$$

- Create the right-hand side vector for the Schur system, $g = \sum_{i=1}^{p} \tilde{R}_i^T g^{(i)}$.
- Perform the matrix-vector operation, $w = Sv$. Recall that,

$$S = \sum_{i=1}^{p} \tilde{R}_i^T S_i \tilde{R}_i = \sum_{i=1}^{p} \tilde{R}_i^T \left(A_{BB}^{(i)} - A_{BI}^{(i)} {A_{II}^{(i)}}^{-1} A_{IB}^{(i)} \right) \tilde{R}_i.$$

  Therefore,

$$w = Sv = \sum_{i=1}^{p} \tilde{R}_i^T \left(A_{BB}^{(i)} - A_{BI}^{(i)} {A_{II}^{(i)}}^{-1} A_{IB}^{(i)} \right) \tilde{R}_i v.$$

  - scatter v: $v_B^{(i)} = \tilde{R}_i v$,
  - calculate: $v_I^{(i)} = A_{IB}^{(i)} v_B^{(i)}$,

- solve for $u_I^{(i)}$: $A_{II}^{(i)} u_I^{(i)} = v_I^{(i)}$,
- calculate: $w_B^{(i)} = A_{BB}^{(i)} v_B^{(i)}$,
- calculate: $v_B^{(i)} = A_{BI}^{(i)} u_I^{(i)}$,
- update: $w_B^{(i)} = w_B^{(i)} - v_B^{(i)}$,
- assemble: $w = \sum_{i=1}^{p} \tilde{R}_i^T w_B^{(i)}$.

Step 4: Preconditioning

For the preconditioner, Eq. (4.45), each step involves a local preconditioning in each subdomain, followed by a piecewise correction over the entire domain

$$\begin{aligned} z = M^{-1} w &= \Big(\sum_{i=1}^{p} \tilde{R}_i^T \tilde{D}_i {A_{BB}^{(i)}}^{-1} \tilde{D}_i \tilde{R}_i + \tilde{R}_0^T S_0^{-1} \tilde{R}_0 \Big) w, \\ &= \sum_{i=1}^{p} \tilde{R}_i^T \Big(\tilde{D}_i {A_{BB}^{(i)}}^{-1} \tilde{D}_i \Big) \tilde{R}_i w + \Big(\tilde{R}_0^T S_0^{-1} \tilde{R}_0 \Big) w, \\ &= \tilde{w} + w_0. \end{aligned}$$

The first step in applying the preconditioner is to build the scaling matrix $\tilde{D}_i$ (represented by a vector), and the coarse grid operator, $\tilde{R}_0$.

- Using the pointers defined in step 1, build the diagonal matrix $\tilde{D}_i$.
- Build $\tilde{R}_0$ operator. Elements in row 'i' of $\tilde{R}_0$ are indeed components of $\tilde{D}_i$ that are properly assembled. Note that $\tilde{R}_0$ and $\tilde{R}_0^T$ are in fact the coarse grid scatter and gather operators.
- Build the coarse grid matrix[7], S_0,

$$S_0 = \tilde{R}_0 S \tilde{R}_0^T.$$

- Calculate $\tilde{w}$:

[7]Since S is not explicitly formed, matrix-matrix multiplication is carried out using matrix-vector products.

- scatter w: $\tilde{w}_B^{(i)} = \tilde{R}_i w$,
- calculate: $v_B^{(i)} = \tilde{D}_i \tilde{w}_B^{(i)}$,
- solve for $u_B^{(i)}$: $A_{BB}^{(i)} u_B^{(i)} = v_B^{(i)}$,
- calculate: $\tilde{w}_B^{(i)} = \tilde{D}_i u_B^{(i)}$,
- assemble: $\tilde{w} = \sum_{i=1}^{p} \tilde{R}_i^T \tilde{w}_B^{(i)}$.

- Calculate w_0:
  - scatter w: $w_0 = \tilde{R}_0 w$,
  - solve for u_0: $S_0 u_0 = w_0$,
  - assemble: $w_0 = \sum_{i=1}^{p} \tilde{R}_i^T u_0$.

It should be noted that when there are only two subdomains, $\tilde{R}_0$ will have identical rows: $\tilde{R}_0 = \frac{1}{2}\{\mathbf{u}, \mathbf{u}\}^T$, where '$\mathbf{u}$' represents a vector whose components are all one. Consequently, $\tilde{R}_0$ and S_0 will both be singular and the coarse grid solve will require special treatment.

Step 5: Solution of the interior problems

- Once the Schur system is solved, once can easily obtain the solution to interior domains by solving a Dirichlet problem in each subdomain:
  - scatter u_B: $u_B^{(i)} = \tilde{R}_i u_B$,
  - calculate: $v_I^{(i)} = A_{IB}^{(i)} u_B^{(i)}$,
  - update: $v_I^{(i)} = f_I^{(i)} - v_I^{(i)}$,
  - solve for $u_I^{(i)}$: $A_{II}^{(i)} u_I^{(i)} = v_I^{(i)}$. ■

As one can see, solution of a system of equations using the Schur complement method is an *expensive* procedure. Each matrix-vector product with the Schur matrix (Sv) requires

three matrix-vector operations with $A_{IB}^{(i)}$, $A_{BI}^{(i)}$, and $A_{BB}^{(i)}$, and a solve involving $A_{II}^{(i)}$ within "*each*" subdomain, keeping in mind that any Krylov subspace iterative method contains at least one such matrix-vector product per iteration. Also, preconditioning is almost always applied, adding to the overall iteration cost.

Since the convergence of the Schur complement method depends largely on the accuracy of the solution involving $A_{II}^{(i)}$ within each subdomain, these solves were performed with a direct solver in the present implementation. That is, $A_{II}^{(i)}$ matrices were factored only once before the start of the iteration loop. Factored matrices were then used for successive solves. In applying preconditioners, the same approach was followed, i.e. subdomain solves involving $A_{BB}^{(i)}$ were performed with a direct solver. This approach, even though memory intensive, insures satisfactory convergence of the Schur system. It also eliminated the need for building robust preconditioners required for iterative solution of systems involving $A_{II}^{(i)}$ and $A_{BB}^{(i)}$. Also, due to the very small size of the S_0 matrix, coarse grid solves were performed using a direct solver.

Chapter 5

Numerical Results

In this chapter, numerical results will be presented for four different problems, to both validate and demonstrate the suitability of the spectral element method for the simulation of acoustic problems. All geometries considered are three-dimensional and all calculations have been performed on multiprocessors ranging from 8 to 48 CPU's, depending on the size of the problem. The test problems cover a range of different geometries and frequencies. The results for all cases are compared with other numerical results or with analytical data, when available.

5.1 Uniform Cylinder

5.1.1 Validation

We start by presenting the results for acoustic propagation in a uniform cylinder of zero thickness (unflanged). This test case is important not only because its analytical solution is known, but also because it is the simplest geometry which resembles the engine inlet. For the sake of completeness. three different combinations of azimuthal (m) and radial (s) modes (m, s) are considered: the plane wave (0,0), the first radial mode (0,1), and the first azimuthal mode (1,0). The frequency at the source is 320.075 Hz, which is equivalent

to a reduced (nondimensional) frequency of $\bar{\omega} = \omega R/c_\infty = 5.91$, or a nondimensional wavelength of $\lambda = 1.063$[1]. The mean flow is assumed to be zero. All calculations are run in parallel using 8 processors on an SGI Origin 3400 machine with 400MHz MIPS R12000 CPU's.

Figure 5.1 shows the outline of the coarse or finite element grid used for the plane wave and first radial mode calculations. Using the finite element grid, the code automatically builds a spectral grid within each element given the number of GCL points, N. A close-up of such a mesh in the inlet area is shown in Figure 5.2. The grid dimensions are $10\times7\times7$, with a total of 3258 elements. The duct radius, R, is chosen to be one and the duct length is $2.5R$. By taking advantage of symmetry, calculations are performed in only a quarter of the domain, as shown (see § 2.4 for details), which obviously leads to a significant savings in computation time .

Figures 5.3 and 5.4 show the acoustic pressure and sound pressure level (SPL) contours, respectively, along the xy-plane for propagation of the plane wave. For this calculation, the number of Gauss-Chebyshev-Lobatto points is $N = 4$, leading to a spectral grid with 220 157 nodes. The size of the Schur system using 8 processors is 16 111. Given the uniformity of the grid, there are an average of 7.4 points per wavelength (*PPW*) in all three directions.

Damping is applied on all external boundaries, except on symmetry planes. The effect of damping is visible in the form of compressed contour lines near the boundaries. The damping parameters are set to $(D, \nu_0, n) = (2, 8, 2)$, where D is the thickness of the damping layer in units of the wavelength, and ν_0 and n are the damping amplitude and exponent, respectively, as defined in Eq. (2.124). For the purpose of validation, directivity at the far-field is compared to the exact solution of Homicz and Lordi [41] and the boundary integral calculations of Lidoine *et al.* [59] in Figure 5.5. The results are adjusted so that SPL peaks at 100 dB. The results show a very good agreement with the exact solution both at small

[1] In the results presented in this chapter, freestream conditions, $T_\infty = 288.15$ K and $p_\infty = 101.325$ kPa, and duct radius or thickness are used for nondimensionalization.

and large angles from the duct axis and also in predicting the location of the minimum SPL, which occurs at around 40°.

The effect of the far-field boundary on SPL is also examined. For this purpose, the same test case was run on a domain of size $13\times10\times10$, leading to 536 813 unknowns, more than twice the number of equations for the smaller domain. The SPL directivity at the same radial distance as in the smaller domain is shown in Figure 5.6. As the results indicate, there is practically no difference between the calculations for the closer and farther domains. Therefore, it is sufficent to use the smaller domain for calculations.

For the first radial mode, (m, s) = (0,1), acoustic pressure contours and directivity plots are shown in Figures 5.7 and 5.8, respectively. The results are obtained on the same grid as the plane wave problem with the same order of approximation ($N = 4$). When only radial modes are present, the fluid merely performs transverse oscillations (while convected along the duct axis). That is, radial modes have transverse but no azimuthal dependence, and therefore their propagation in axisymmetric. Depending on the order of the mode 's' being propagated, there will be the same number of nodal '*circles*' on any cross section of the duct. For the first radial mode, only one such nodal circle exists, which is located at about $0.63R$ from the duct axis[2], as shown by the gap between pressure contour lines in Figure 5.7.

Directivity of the SPL (Figure 5.8) compares well with the exact solution except near the axis. This behavior also exists in the other numerical results. This is due to the fact that directivity is measured at a "finite" distance from the duct exit, as compared to the "infinite" distance for the exact solution. As the waves move farther from the duct exit, acoustic intensity along the axis, and therefore SPL, will diminish since acoustic energy has to be distributed over a larger surface area. In Figure 5.8, directivity is measured at a distance of $d = 4.14R$ =3.9λ from the duct exit, while in Lidoine *et al.* [59] the directivity surface is $25R$ away. To further demonstrate this, the same test was performed in a larger domain and directivity was measured at a distance of $d = 6.7R$ =6.3λ from the duct exit,

[2]This distance is the ratio of the zero's of $J_0(k_{01}r)$ and $J_0'(k_{01}r)$: 2.4048/3.8317 = 0.6276.

as shown in Figure 5.9. The SPL profile for the farther directivity measurement clearly indicates an improved solution, confirming the above argument.

For propagation of a uniformly moving fluid in an infinite duct, Farassat and Myers [25] use a graphical interpretation to derive the following formula for location of the principal lobe measured in terms of the angle from the duct axis, θ_p,

$$\tan\theta_p = \frac{\beta}{\left|(M_\infty - M)\,\xi_{ms} - \sqrt{\xi_{ms}^2 - 1}\right|}. \tag{5.1}$$

In the above equation, $\beta = \sqrt{1-M^2}$, ξ_{ms} is the cut-off ratio, and M and M_∞ are the Mach numbers in the duct and at the far-field, respectively. In the absence of mean flow, $M_\infty = M = 0$, and the equation simplifies to

$$\tan\theta_p = \frac{1}{\sqrt{\xi_{ms}^2 - 1}}, \tag{5.2}$$

which is the same result obtained by Rice *at al.* [82]. For the plane wave, the principal lobe occurs on the axis ($\theta_p = 0$) because the cut-off ratio $\xi_{00} = \infty$. For the first radial mode at $\bar{\omega} = 5.91$, the radial wave number $k_{01} = 3.8317$, which gives a cut-off ratio of $\xi_{01} = 1.542$. This predicted location of the principal lobe is $\theta_p = 40.42°$, which is very close to the calculated value of 41.17° in Fig. 5.8.

The last validation test concerns radiation of the first azimuthal (spinning) mode, $(m, s) = (1,0)$. This test is also used to demonstrate the validity of the theoretical development for symmetry boundaries conditions, as presented in § 2.4. Figure 5.10 shows the computational domain with dimensions of $7\times14\times10$. The duct geometry is the same as before, and so are the frequency and damping parameters. The yellow surface indicates the plane of symmetry. There are a total of 2004 elements in the domain. Given that domain size is twice as large as the domain in the plane wave problem, but the number of elements are not doubled, a higher order approximation ($N = 5$) is used to provide the necessary resolution. A 5^{th}-order approximation gives an average of $PPW = 6.38$. The total number of unknowns and the size of the Schur system for this case are 263 816 and 18 685, respectively.

Mode (m, s)	(0,0)	(0,1)	(1,0)
Building Matrix	10:46	10:43	33:34
Solving Equations	9:29	7:57	14:43
Total (min:sec)	20:15	18:40	48:17

Table 5.1: Computation time for validation test cases.

Figure 5.11 shows the acoustic pressure along the xy-plane. Tangential variations are evident in the figure, pointing to a counter clockwise swirling motion, as shown in several cross sectional views. Similar to the radial modes, azimuthal modes form nodal '*lines*' at any cross section of the duct that rotate 2π radians per wavelength. The number of nodal lines is equal to the order of the mode m being propagated. In this case, only one such line exists, which is clearly visible at the acoustic source in the form of a straight line parallel to the z-axis.

Unlike the plane wave, the acoustic energy of both the radial and azimuthal modes peaks off the axis. This is shown in Figure 5.12 for the first azimuthal mode, where the SPL isosurface is plotted along with the SPL contour lines in the lower plane. The maximum SPL occurs at around 20° from the axis, with zero acoustic energy radiated on the axis itself. Figure 5.13 shows directivity of the radiated field and its comparison with theoretical data. Overall, the results show a very good agreement both qualitatively and quantitatively.

All the above test cases were solved using the preconditioned Schur complement method with the GMRES iterative solver, for which the restart parameter was set to 400. Figure 5.14 shows the convergence history for the three problems and the corresponding computation time is given in Table 5.1.

5.1.2 Effect of the Order of Approximation

One of the main advantages of high-order methods, especially spectral methods, is that one can use a smaller number of *PPW* by using a higher order of approximation within each

element (we already observed this in the previous section when presenting the results for the first azimuthal mode). This relation, however, is not linear, in the sense that by doubling the order of approximation *less* than twice the number of points are required to obtain the same level of accuracy. As a result, increasing the order of approximation will lead to a smaller number of unknowns. To show the effect of the order of approximation (N) on accuracy and *PPW* requirement, two different tests are presented.

First, the effect of N for a fixed number of elements has been studied. Using the grid with 3258 elements, calculations were performed for three different values of N. Figures 5.15 and 5.16 show the contours of acoustic pressure amplitude along the xy-plane as N is varied between 3 and 5, corresponding to 5.5, 7.4, and 9.2 points per wavelength, respectively. The figures also show the spectral grid used for each case. For $N = 3$, the acoustic field is clearly underresolved. But, as N increases, the contour lines become smoother indicating improvement in the accuracy, as expected. The figure also shows that the results for the fourth and fifth order approximations are almost identical. This indicates that further increase in the order of approximation do not yield any significant changes in the accuracy. This is also shown in Figure 5.17, where the far-field directivity is plotted for different values of N. Given that increasing the order of approximation from fourth to fifth order yields the same results, one can conclude for the given frequency and grid, that a 4^{th}-order spectral method provides an adequate resolution.

The second test demonstrates another aspect of using a high-order method. Here, the number of elements in the domain has been reduced by about 10 times from 3258 to 320, but the order of approximation has been increased from $N = 4$ to $N = 6$, i.e. only 1.5 times. Coarsening the grid leads to an approximately 66% reduction in the total number of unknowns from 220 157 for $N = 4$, to 74 881 for $N = 6$. Note, however, that decreasing the number of elements by almost a factor of ten does not lead to a tenfold decrease in the number of equations. This is because when a higher order of approximation is used many extra points are generated within each element. For the N =6 case, 10 processors were used and total assembly time depsite a smaller number of equations was 17 min. and 20

sec., which was higher compared to an assembly time of 11 min. for the N =4 run. This was, however, somewhat compensated by the solution time, which was only 4 minutes for N =6 as compared to about 9 minutes for the lower-order test.

Figure 5.18 shows part of the coarse grid with 320 elements. Also shown are the SPL contour lines along the xy-plane for the plane wave propagation, which are comparable to those in Fig. 5.4. In the coarse grid, even with increased N, the *PPW* has been reduced significantly to 5.1 from 7.4 for the 3258-element grid without sacrificing accuracy (Figure 5.19).

Another attractive feature of high-order methods is that once the finite element grid is generated, it can be used for computations over a range of frequencies by simply changing the order of approximation within each element, without having to regenerate another grid. An example of this approach is demonstrated in section 5.1.3.

Increasing the order of approximation, however, has negative consequences in terms of increased cost associated with building the matrix, which varies exponentially. This is demonstrated in Figure 5.20, where the CPU time is plotted versus N for the test problem with 320 elements.

While for 2D problems one can use a very high order of approximation, e.g., N = 10-11 [93], such a practice for 3D problems would be very costly, if not computationally prohibitive. Our experience has been that for 3D problems, a value in the range of N = 4-6 would be optimum in terms of minimizing the computation time while providing sufficient accuracy.

5.1.3 Effect of the Lip Thickness

Given that realistic engine inlet geometries have a finite thickness, it would be appropriate to study the effect of lip thickness on the radiated acoustic field. For this purpose, two inlet shapes with circular lips are considered, with e/R = 1/8 and 1/4, where e is the lip thickness (Figures 5.21 and 5.22). Using the plane wave test conditions described earlier, i.e. $\bar{\omega}$ = 5.91 and a grid with 3258 elements and N = 4, the SPL directivity was calculated

for the two geometries and compared with the results for the zero-thickness cylinder. The results are shown in Figure 5.23. According to the figure, at small angles near the duct axis ($\theta \leq 30°$) there is a little difference between the flanged and unflanged duct results. The situation, however, is different at higher radiation angles, where the minimum SPL occurs at smaller angles due to the finite thickness of the wall. Also, there is a significant reduction in the sound pressure level for $\theta > 60°$ due to reflection of sound waves near the lip. These changes are more significant for the thicker lip, as expected.

The effect of lip thickness has been studied by a number of researchers [59, 88], who have documented similar variations in SPL directivity. The variations are attributed to the frequency as well as the level of geometrical singularity (curvature radius) of the lip, which plays a key role. For the same inlet shape, these two parameters are interrelated in the sense that with an increase in frequency, the ratio e/λ also increases making the lip appear thicker to the waves. This effect is shown in Figure 5.24 for a much higher frequency ($\bar{\omega} = 10.3$). The grid is the same as before, but due to higher source frequency, $N = 6$ is used. Compared to the results for the lower frequency, $\bar{\omega} = 5.91$, SPL deteriorates much worse at higher angles. Also, for a given frequency, higher modes (both azimuthal and radial) are more sensitive to the shape of the lip and its thickness. This is because such modes carry a larger part of their energy away from the duct axis and closer to the wall, noting the shape of the Bessel functions for such modes.

5.1.4 Effect of the Uniform Flow

So far, acoustic radiation from the ducts in the absence of any external flow was considered. For problems of interest to the subject of this research, that is the simulation of engine noise during take-off and landing, the inlet duct is exposed to a subsonic inflow with a Mach number in the range $M_\infty = 0.2$-0.3. To study the effect of external flow, a set of tests with different Mach numbers are conducted, assuming uniform flow. Plane wave propagation at a frequency of $\bar{\omega} = 5.91$ and a grid with 3258 elements is considered.

The results are shown in Figure 5.25. At low Mach numbers, the effect of uniform

flow on acoustic radiation in the far-field is very similar to the effect of lip thickness. Note that increasing the Mach number leads to an increase in the axial wave number k_x, and consequently a decrease in the radial wave number, k_{ms}, according to Eq. (2.94). So, the cut-off ratio not only will be higher, but in some cases could become greater than one. As a result, certain modes that were originally cut-off, will become cut-on. This is shown for the plane wave at $M = 0.6$, where it appears to have another minimum at around $\theta = 80°$, not very well captured in the figure.

5.1.5 Effect of the Mean Flow

One of the objectives of the current research was to include the effect of mean flow on the radiated sound field. For this purpose, a mean flow solver using a compressible potential model was included to accurately determine variations of the flow around the inlet. Figure 5.26 shows the results of such calculations, where the velocity field and Mach contours around an inlet with a finite thickness of $e/R = 1/8$ are plotted. The results correspond to a freestream Mach number of $M_\infty = 0.2$ and a mass flow rate on the fan face ($x = 0$) of 50 kg/s. These conditions translate into a Mach number of $M_f = 0.6$ on the fan face.

Given the high gradient of velocity field near the lip, as compared to the uniform flow, it is expected that far-field directivity will be affected more negatively at higher radiation angles. Calculated SPL in Figure 5.27 confirms this variation. Since the acoustic and mean flow solvers are not coupled, the effect of mean flow solution on convergence of the acoustic solver is expected to be negligible. This is shown in Figure 5.28, which also shows the convergence history of the mean flow solver.

For the mean flow problem, a grid with 3258 elements was used, and the order of approximation was $N = 4$. This is the same order used for the acoustic solver, because the mean flow variables (ρ_0, u_0, v_0, w_0) at the nodes of the spectral grid are needed for the acoustic calculations. This leads to a system of equations with 220 157 unknowns. As explained in Chapter 4, a Conjugate Gradient solver with additive Schwarz preconditioner was used to solve the nonlinear equations using a Newton iterative loop. At every iteration

of the Newton loop, the computation time to build and assemble the mean flow coefficient matrix was about 8 minutes and the corresponding solution time close to 78 seconds. Each solve with the preconditioned CG method required about 76 iterations to converge.

5.2 2D Semi-infinite Duct

This test case will demonstrate the ability of the code to perform simulations for 2D problems and rectangular inlets. The geometry consists of two horizontal walls, one unit apart, with a ratio of duct length to height, $L/H = 2.5$, as shown in Figure 5.29. The size of the domain is $7\times11\times1$, comprised of 459 elements. A sixth-order approximation is used within each element ($N = 6$), leading to a total of 118 027 equations.

Figures 5.30 and 5.31 show the acoustic propagation within the duct and into the far-field for the second and third transverse modes, respectively, for a reduced frequency of $\bar{\omega} = 15$ in the absence of external flow. Note the change in directionality of the acoustic energy away from the axis for the higher cross mode. As discussed in § 2.2.4, acoustic modes in a rectangular duct are cosine functions, with their maximum amplitude at the walls to satisfy the zero normal velocity condition. Looking closely at contour lines within the duct, especially near the source, one could recognize the cosine curves enveloping pressure contours.

For the semi-infinite duct, the cut-off ratio, Eq. (2.118), simplifies to $\xi_m = \bar{\omega}/(m\pi)$. With $\bar{\omega} = 15$, this value for the second and third transverse modes is: $\xi_2 = 2.387$ and $\xi_3 = 1.591$, respectively. Both values are greater than one, which indicates a cut-on or propagating mode.

Simulations in 2D using a 3D code was made possible by applying periodic boundary conditions in the z-direction. As a result, as shown in Figure 5.32, only one element in that direction is required, irrespective of the domain length. Also shown in the figure, are the acoustic pressure contour lines, which show no variations in the z-direction– an indication of proper implementation of periodic boundary conditions.

Far-field directivity is compared to the exact solution of Mani [69] and numerical results of Dong *et al.* [15] in Figures 5.33 and 5.34 for the second and third cross modes, respectively. SPL is adjusted so that its maximum occurs at 100 dB. The results are in very good agreement with the analytical data. Again, notice the discrepancy in SPL near the axis (similar to the (0,1) mode propagation in § 5.1.1), which is due to the finite radius of the surface on which directivity is measured. For this problem, directivity surface was 7λ away from the duct exit plane.

Given the relatively small size of the problem (118 027 unknowns), a parallel direct solver called SPOOLES [1] was used to solve the system of equations. A total of 12 CPU's were used on an SGI Origin 3800 machine. Solution of the system was obtained in about 90 seconds. The time needed to build and assemble the coefficient matrix was close to 22 minutes, which was relatively high and due to the fact that a high order of approximation ($N = 6$) was used. Due to the higher frequency of the acoustic source, damping layer thickness and amplitude were slightly increased to $(D, \nu_0, n) = (2.5, 15, 2)$, compared to the cylinder test problem.

The results in Dong *et al.* were obtained using a Dispersion-Preserving-Relation (DPR) finite difference scheme [99] in the near-field and a Kirchhoff method in the far-field. They used a minimum of 10 mesh points for the shortest wavelength. However, in the present calculation with $N = 6$, the number of points per wavelength was approximately $PPW = 6$, showing the advantage in using a spectral method.

5.3 3D Curved Duct

This is a test case for which experimental data and numerical results for two different methods, i.e. a boundary integral method [67] and nonlinear Euler calculations [91] are available. Therefore, it provides a good opportunity to compare three different numerical methods. The problem is also interesting in that it involves propagation of noise through a curved duct, as opposed to straight ducts studied earlier, with implications that will be

discussed later.

Figure 5.35 shows the duct geometry, with the outline of the finite element grid built around the duct shown in Figure 5.36. For the integral method, the acoustic pressure on the surface of the duct exit plane was used for calculating far-field radiation. However, present calculations and those in Ref. [91] both require an external grid surrounding the duct exit, as shown. The external grid starts right after the bend and extends $10R$ in all three directions relative to the duct exit plane. The grid contains a total of 1776 elements. Figure 5.37 shows a close-up of the duct and the spectral grid. Note that the spectral grid is obtained by projecting wall boundary nodes on the CAD surface. This is done automatically by the spectral mesh generation routine.

The first set of calculations were performed for the plane wave ($m = 0$) and the first azimuthal mode ($m = 1$) using a frequency of 1000 Hz ($\bar{\omega} = 2.769$). Within each element, a 5^{th}-order spectral approximation was used, leading to a spectral grid with 234 341 unknowns. Damping parameters were set to $(D, \nu_0, n) = (1.5, 6, 2)$. The domain was partitioned into 12 regions ($p = 12$), with a total of 20 784 unknowns on the interface boundaries (size of the Schur system). Building the matrix for the entire spectral grid took about 20 minutes of CPU time while solving the system of equations using the Schur complement method required about 7 minutes for both tests.

Figure 5.38 shows the pressure contours for the plane wave propagation on the duct wall and along the exit plane at the far-field . Note the symmetry of the radiated sound with respect to the xy-plane. For $\bar{\omega} = 2.769$, the wavelength is $\lambda = 2.268$, which means the wave at the source will travel about 5λ along the straight part of the duct before reaching the bend, as shown by the yellow stripes in the figure. The acoustic field outside the duct, however, is not symmetric in all directions. The symmetry exists only in the xy-plane, as the acoustic pressure contour lines and the isosurface of SPL directivity at the duct exit indicate (Figure 5.39).

As demonstrated by Malbéqui *et al.* [67], the presence of the bend and reflection of the waves from the duct exit's create additional azimuthal modes which are responsible

for asymmetry of the radiated field. For the plane wave, their modal analysis of the wall pressure field at the duct exit shows strong presence of two other modes, $m = \pm 1$, with an amplitude comparable to the mode specified at the source ($m = 0$). This is also apparent in the shape of acoustic pressure contours at the duct exit in Fig. 5.39.

Figures 5.40 and 5.41 show the comparison of numerical results for the normalized pressure amplitude along the yz-plane for the $m = 0$ and $m = 1$ modes, respectively. In both cases, especially for the plane wave, the results are in a very good agreement with both the experiment and other numerical data. The results by Stanescu *et al.* [91] were obtained using a multidomain spectral method with a radiation boundary condition applied at the far-field. For the first azimuthal mode ($m = 1$), all numerical methods show discrepancy with experimental results for $\theta > 20°$, which could be attributed to the extra modes are present at the duct exit, but not specified as boundary conditions at the source.

Figure 5.42 shows the variation of acoustic pressure on the duct plane of symmetry (xy-plane) for the first azimuthal mode. For a straight circular duct, the pressure amplitude along the duct axis should be zero (see, e.g., Fig. 5.13). However, this is not the case for the curved duct. This confirms that there are other modes with different phase angles present at the duct exit.

Increasing the frequency to 1500 Hz ($\bar{\omega} = 4.154$), calculations were performed for the propagation of the second azimuthal mode $(m, s) = (2,0)$. Due to higher frequency, a slightly higher order of approximation ($N = 6$) was used. Pressure contours with only positive values are shown in Figure 5.43 to demonstrate the tangential variation of these modes as they travel along the duct. At the duct exit, directionality of sound pressure level is shown with an isosurface. For a straight duct, $m = 2$ would create only 2 lobes with a large common surface area (similar to the $m = 1$ mode in Fig. 5.12). However, the irregular shape of the SPL isosurface indicates that presence of the bend has caused significant change in the structure of the radiated field. In fact, for this case modal analysis indicated the presence of 5 modes ($m = 0, \pm 1, \pm 2$) with relatively equal amplitude at the duct exit [67]. The asymetry is further evident in Figure 5.44, where contour lines of the

Mode (m, s)	(0,0)	(0,1)	(0,2)
Building Matrix	19:54	19:56	80:13
Solving Equations	7:07	6:36	23:47
Total (min:sec)	26:51	26:32	104:00
No. of Iterations	215	198	259

Table 5.2: Computation time for the curved duct test cases.

SPL at the duct exit plane ($y = 2.26R$) are plotted.

Figure 5.45 shows the convergence history for the three tests presented above. For all three, calculations were performed on 12 CPU's using the preconditioned GMRES algorithm. For the $m = 2$ test, the total number of unknowns were 401 329, while the Schur matrix was about one order of magnitude smaller with only 29 903 unknowns. Also, due to higher frequency, slightly larger damping layer and amplitude was used, $(D, \nu_0, n) = (2, 8, 2)$, compared to the first two calculations. Computation time for the core calculations (building the coefficient matrix and solving the equations) is given in Table 5.2.

5.4 Generic Scarfed Inlet

As the final test case and also the largest numerical calculation, the results of acoustic radiation from a generic scarfed inlet are presented. This problem represents a realistic example of an engine geometry and its propagating mode. Figure 5.46 shows the geometry of the scarfed inlet in which the lips have elliptical profiles. For numerical calculations, the exterior boundary is moved by one meter behind the fan face to create enough thickness for damping to be applied. The size of the computational domain is therefore 4.7×7.4×7.4 m, with a total of 15 328 elements.

A close-up of the spectral mesh around the inlet is shown in Figure 5.47. The ParMETIS library [47] was used to partition the computational domain into smaller nonoverlapping

subdomains for implementation of the Schur complement method. Figure 5.48 shows the outline for a typical partitioning, where each color represents a separate subdomain. ParMETIS performs the partitioning in such a way that each subdomain contains relatively the same number of nodes, interior and boundary combined. This is to ensure a balanced load distribution between the processors and subsequently a better parallel efficiency. Figures 5.49 and 5.50 show the element and corresponding node distribution for a 48-subdomain partitioning.

We examine the acoustic propagation of the (13,0) mode in the absence of mean flow at a relatively high frequency of $\bar{\omega} = 17$. For this frequency and azimuthal mode, the cut-off ratio is $\xi_{13,0} = 1.138 > 1$, which indicates a propagating mode. A fifth-order approximation is used within each element ($N = 5$) equivalent to an approximate number of 7.77 points per wavelength. Given the relatively high frequency, a larger damping layer is used with damping parameters set to $(D, \nu_0, n) = (3, 10, 2)$.

Figure 5.51 shows the acoustic pressure contours at the inlet (fan face), inside the nacelle and on its walls, and at a distance from the duct exit. Given the shape of the Bessel functions, for higher azimuthal modes the acoustic energy is convected closer to the walls. This is clearly visible by the shape of contour lines on the inlet face and also by the pressure amplitude contour lines shown in Figure 5.52.

The difference in the thickness of the lower and upper lips as well as the extended lower lip causes asymmetry in the radiated field. This is well demonstrated in Figure 5.53, where the sound pressure level isosurface is plotted. Also, note the directionality of the radiated sound toward the upper part of the xy-plane. The asymmetry is also visible in Figure 5.54, which shows the radiation in the xz-plane using the acoustic pressure contours at the top, middle, and bottom of the computational domain. Note the lower magnitude of the acoustic pressure on the bottom surface.

Directivity of the SPL was measured at a distance of 5λ from the nacelle exit plane and plotted against the numerical results of Hamilton and Astley [38] in Figures 5.55 and 5.56. The method used in [38] is a hybrid approach, where quadratic finite elements are

used in the near-field and 5^{th}-order infinite elements in the far-field. Infinite elements use an exponentially decaying shape function to simulate the propagation behavior of waves away from the source [2]. They solved the resulting system of equations, originally written in frequency domain, with a time marching approach by first transforming them using a complex Fourier transform.

Referring to Figures 5.55 and 5.56, numerical values near the duct axis ($-20° \leq \theta \leq 20°$) are meaningless and would diminish if a higher order of approximation was used. Outside this region, however, there is little match between our results and those presented in [38]. To gain a better insight and to better judge the quality of results for the two numerical methods, analytical results corresponding to propagation from a zero-thickness uniform duct at the same frequency and azimuthal mode are also plotted in the figures. By comparison, the spectral method provides very smooth curves and "qualitatively" exhibits a similar trend as the exact solution. The results presented by Hamilton and Astley, on the other hand, are very oscillatory and only weakly resemble the exact data or the present calculation. However, it seems that both numerical methods predict the location of maximum SPL closely. Given the above argument, the accuracy of the method presented in [38] is questionable.

The location of the principal lobe as predicted by equation (5.2) with a cut-off ratio of $\xi_{13,0} = 1.138$ is 61.42°. Due to the asymmetric duct shape and also lip thickness, these values for the scarfed inlet are -70.86°,-63.82°, 67.5° and 72.84°. In section 5.1.3, the effect of finite lip thickness was found to be a reduction in the far-field SPL directivity, especially at large angles away from the duct axis. This explains the deviation in our results from the exact solution for $|\theta| > 70°$ in Figures 5.55 and 5.56. Also, note the asymmetry in directivity curves along both the xy- and xz-planes. This is more pronounced along the xy-plane, where the maximum SPL is about 4.5 dB less in the downward direction. This difference is also clearly visible in Figure 5.53, where the isosurface extends farther in the upper region. The reduction in the SPL is due to the longer lower lip, which shields part of the acoustic energy directed in that direction. For this reason, scarfed inlets were one

No. of Rows (Nodes)	(%)	Nonzero per Row
1 050 840	53.349	216
734 456	37.287	396
216	0.011	546
170 654	8.664	726
280	0.014	906
48	0.002	1001
13 188	0.670	1331
64	0.003	1661

Table 5.3: Sparsity structure of the global coefficient matrix; scarfed inlet test case.

of the early methods studied for reduction of noise propagated from the engine toward the ground.

The spectral grid with 15 328 elements and $N = 5$ contains close to two million (1 969 746) unknowns. The global coefficient matrix, however, is extremely sparse and contains only about 660 million (659 800 016) nonzeros. This means only less than 1% of the matrix is filled (sparsity of 99.983%). A better understanding of the sparsity of the matrix can be achieved by analysing the data in Table 5.3, which shows the number of nonzeros (nnz) per row. The numbers in the first and second rows correspond to the interior and face nodes of the spectral elements (connected only with the nodes within one or two elements), respectively. Note that the entries of the matrix are complex numbers and all variables have a double precision accuracy. So, storing the nonzeros alone requires about 2.5 GB of memory. Given that the bulk of the matrix (90%) contains a very small number of nonzeros per row, it is extremely important to allocate the memory accordingly. Proper memory management is crucial in achieving a high performance, especially when a large system of equations is solved. It will also minimize or eliminate dynamic memory allocation, which could significantly increase matrix assembly time.

Computations were performed in parallel using 48 CPU's. With this number of partitions, 148 137 nodes were generated on interface boundaries (size of the Schur system). This is more than one order of magnitude smaller than the size of the original system (more precisely 7.5%). Calculations were performed on an SGI Origin 3800 machine with MIPS R12000 processors, each with a clock rate of 400 MHz. By comparison, the coefficient matrix in Ref. [38] contained approximately 450 000 unknowns and calculations were done on a single 1GHz processor.

To examine the behavior of iterative methods, two different solvers, i.e., the TFQMR and restarted GMRES were used to solve the system of equations using the Schur complement method and the proposed preconditioner, Eq. (4.46). Figure 5.57 shows the convergence history for the two methods for a 32 CPU run. While restarted GMRES shows a monotonic convergence, TFQMR has an irregular behavior and starts to slow down after close to 1100 iterations. GMRES solver converged in 484 iterations in 2hr and 19 min., while TFQMR required 2758 iterations lasting 21hr and 27 minutes. This obviously points to the fact that GMRES is a much better choice for this kind of problems. In both methods, however, memory usage was very large given that matrices for interior subdomains ($A_{II}^{(i)}$) each of an approximate size of 4×10^5, and boundary matrices ($A_{BB}^{(i)}$) of relative size 7×10^4 were all solved by a direct method. At the peak of the calculations, the total memory usage was roughly 70GB.

It should be pointed out that a slowdown in convergence rate was observed when the restart parameter for the GMRES solver was set to a small value, ($m = 30$). Figure 5.58 shows the effect of the restart parameter on convergence of the Schur system for a 24-cpu run. It seems that stagnation or slowdown in convergence happens when frequency is relatively high, as such a negative trend was not observed for smaller values of $\bar{\omega}$. The computation time with GMRES(600) was about 2hr 7min. while it took 9hr 24min. and four times more iterations to converge to the same tolerance with GMRES(30), as shown in the figure. Therefore, to speed up the convergence all calculations using GMRES solver were performed with a restart parameter of ($m = 600$).

No. of CPU's	8	16	24	32	40	48
Building Matrix	4:25	2:11	1:27	1:04	0:52	0:44
Solving Equations	7:41	3:40	2:07	2:19	2:06	1:05
Total (hr:min)	12:17	5:51	3:34	3:23	2:58	1:49
No. of Iterations	216	288	406	484	586	614

Table 5.4: CPU time for different parallel computations; scarfed inlet test case.

The effect of the number of processors on the convergence rate of the GMRES method is shown in Figure 5.59. The figure shows that a faster convergence is obtained when a smaller number of CPU's is used. However, to make a proper judgment the corresponding computation time should also be taken into account. Table5.4 shows that despite a larger number of iterations, calculations with 48 processors result in the shortest computation time. For the 8- and 16-CPU runs the Jacobi preconditioner (M_J) was used, while the other results were obtained using the proposed preconditioner. By comparison, the results in [38] were obtained using the Bi-CGSTAB algorithm with a diagonal preconditioner in approximately 10 hrs.

Reduction in the number of iterations as a smaller number of CPU's is used is attributed to the fact that the ratio of the number of boundary nodes to the internal nodes decreases with a decrease in the number of subdomains. As a result, the Schur matrix will be smaller and the proposed preconditioner will be a better representation of the global boundary matrix, thus providing a faster convergence.

To show the parallel efficiency of the code, computation time for building and assembling the matrix vs. the number of processors is plotted in Figure 5.60. Given the extremely large size of the problem, it was not possible to run the test on a single CPU, as stipulated in Eq. (3.6). Instead, the 8-processor results were used as the reference data. The parallel performance is very satisfactory, showing a *superlinear* speed up for the majority of the calculations. Such an excellent scaling is attributed to many factors, among which is the

optimality of the algorithms used in the code and also an efficient memory management. For example, after generating the spectral mesh, additional pointers were created to ensure that each CPU would only process the elements it has been assigned to in the partitioning stage, so that interprocessor communication would be minimized. Optimization measurements like this are necessary in order to gain a high parallel efficiency, especially when a large number of equations, say $> 300\,000$, are solved.

Overall, about 80%–85% of the total computation time was spent on building, assembling and solving the system of equations (core calculations) and the rest on pre- and post-processing, such as decomposing the domain, creating the spectral grid and its associated pointers, memory management, etc.

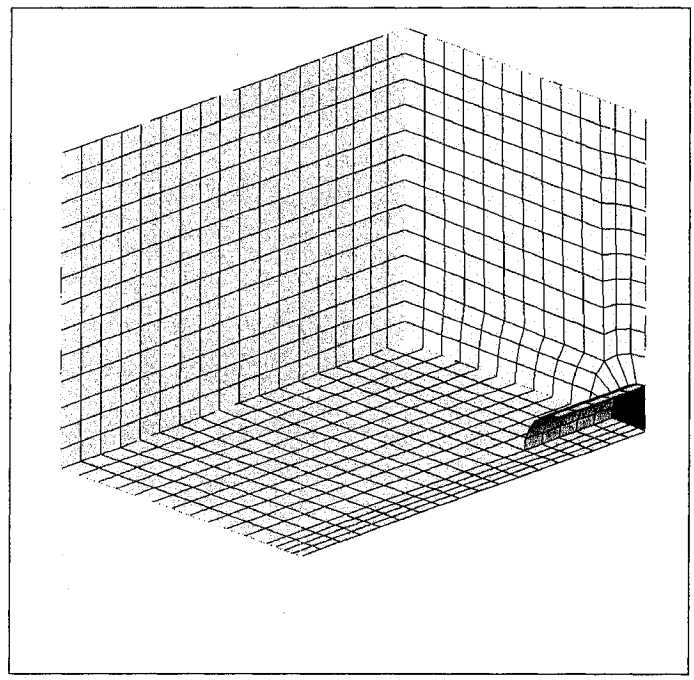

Figure 5.1: Computational domain and background finite element grid for the uniform cylinder.

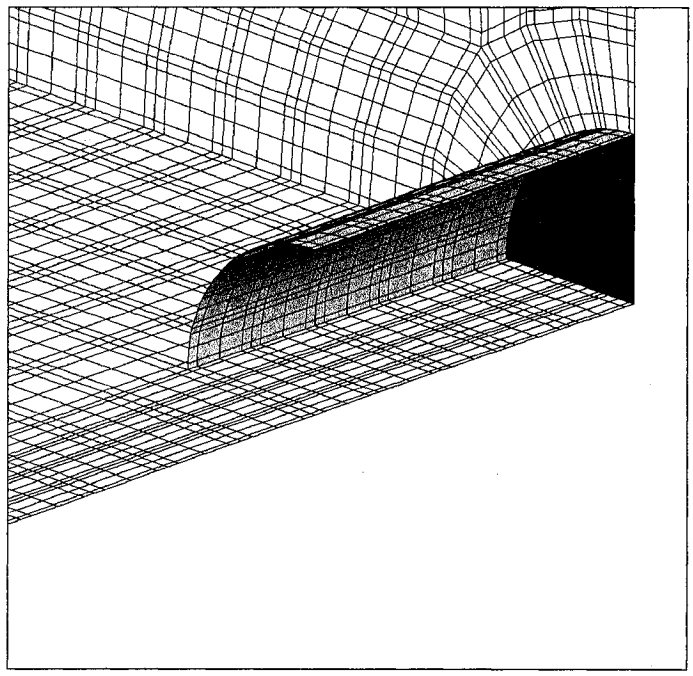

Figure 5.2: Outline of the spectral grid inside the cylinder.

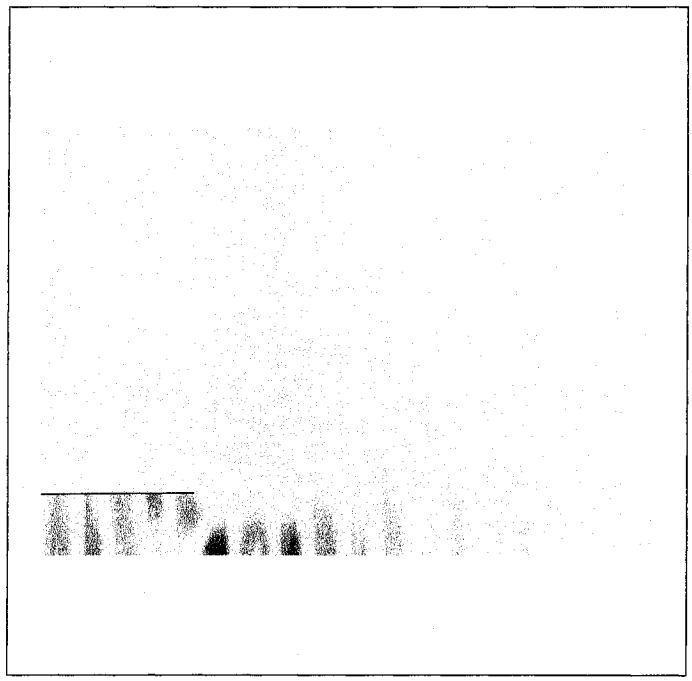

Figure 5.3: Acoustic pressure distribution for the plane wave radiation.

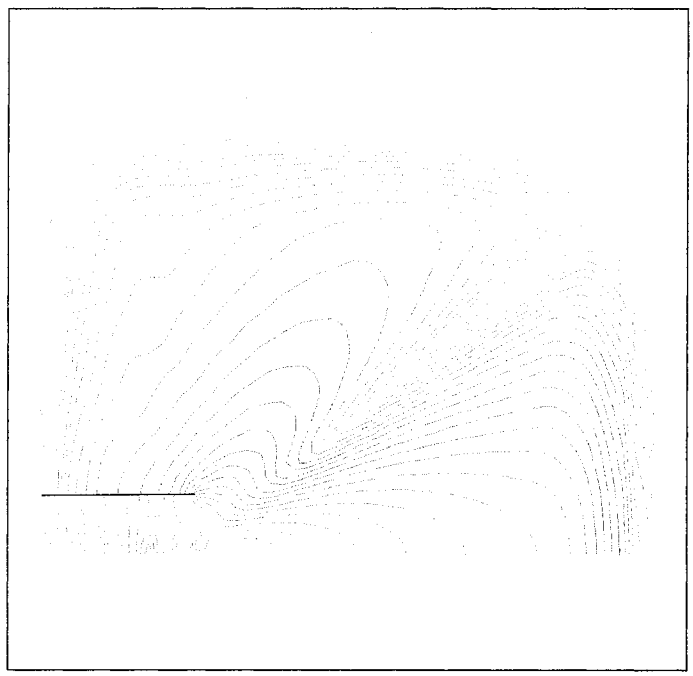

Figure 5.4: SPL contours for the plane wave radiation.

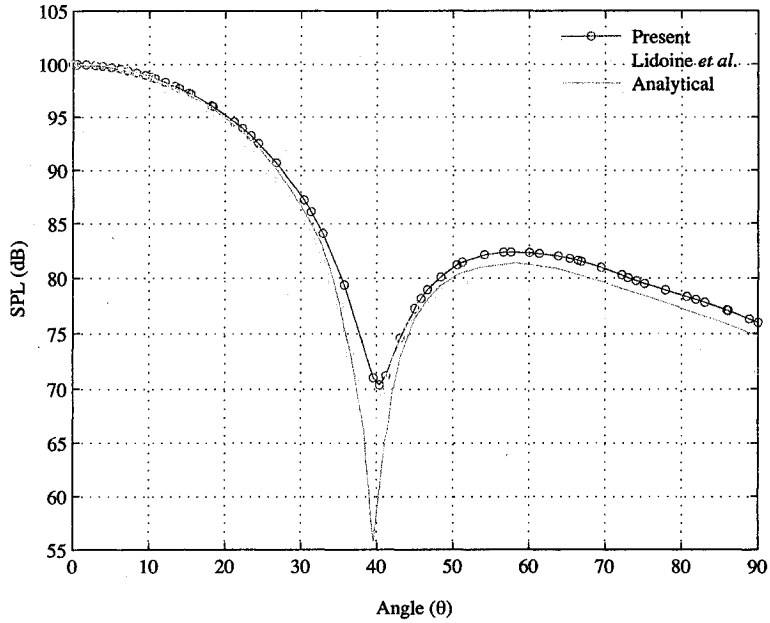

Figure 5.5: Far-field directivity of SPL for the plane wave.

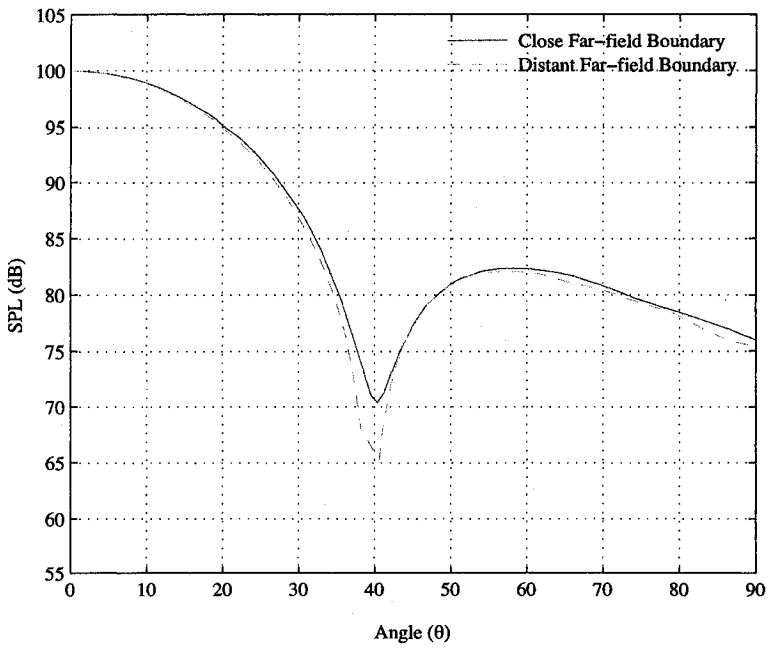

Figure 5.6: Comparison of SPL directivity at the far-field for domains of different size.

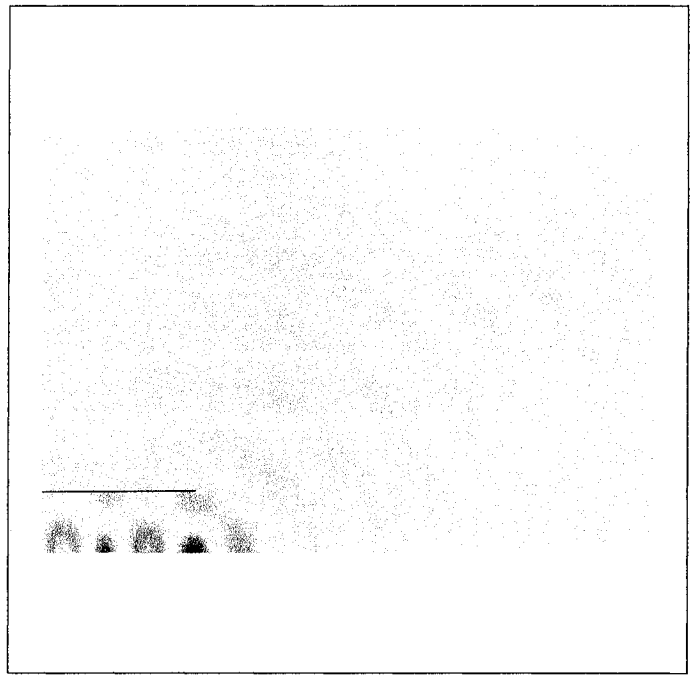

Figure 5.7: Acoustic pressure distribution for the first radial mode; (0,1).

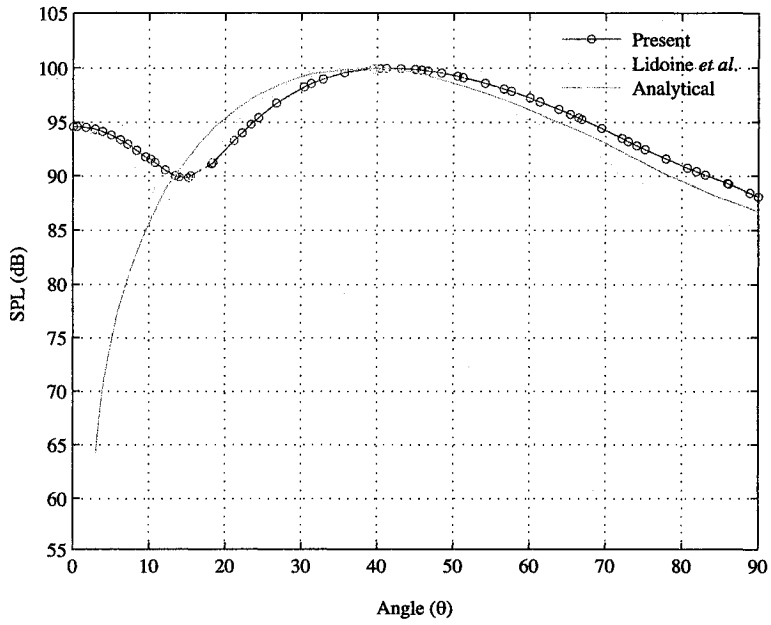

Figure 5.8: Far-field directivity of SPL the first radial mode.

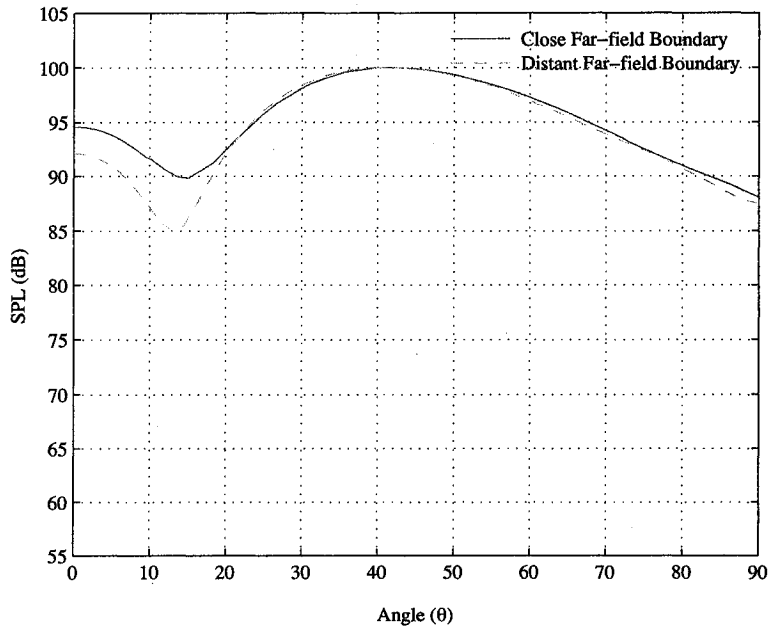

Figure 5.9: Effect of far-field boundary on SPL directivity near the axis for the first radial mode.

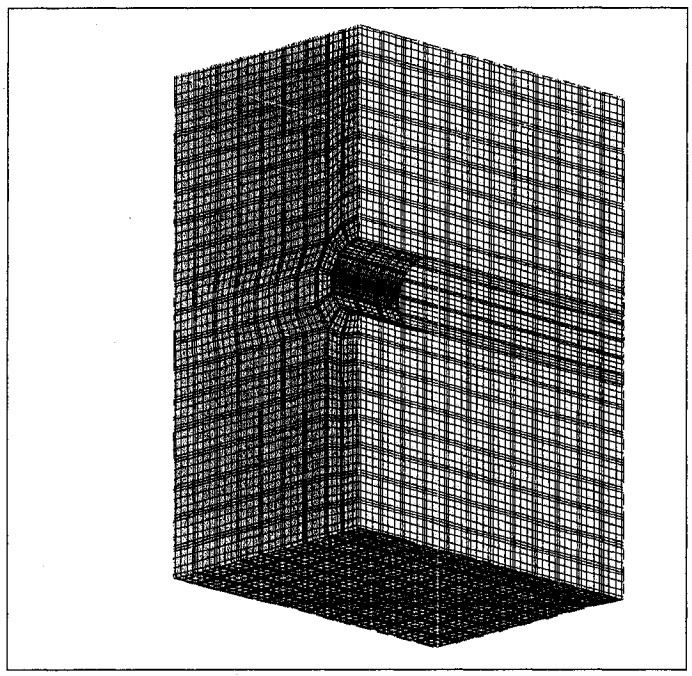

Figure 5.10: Computational domain and spectral grid outline for the first azimuthal mode calculations.

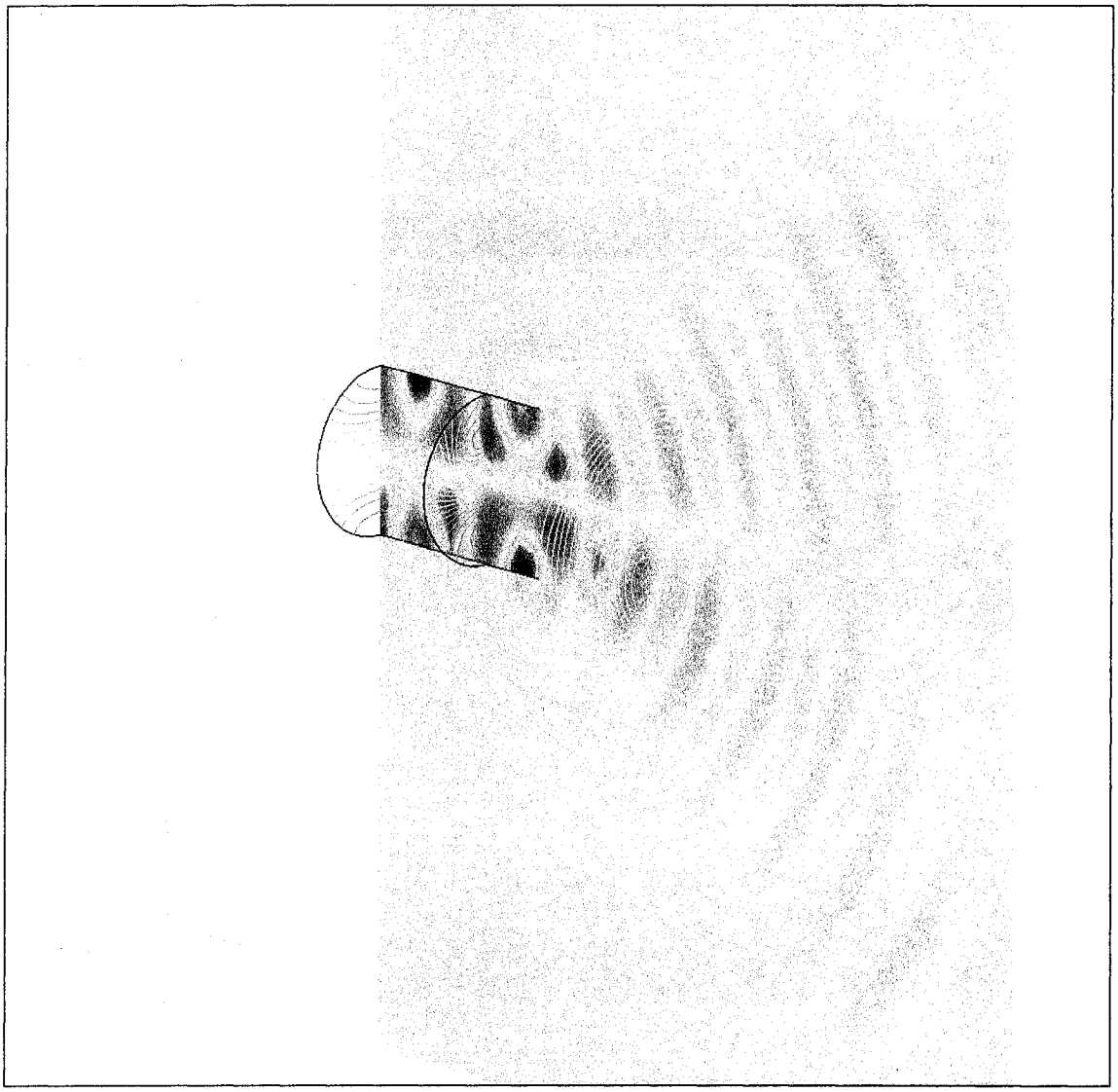

Figure 5.11: Acoustic pressure distribution for the (1,0) mode along the plane of symmetry and at several axial locations.

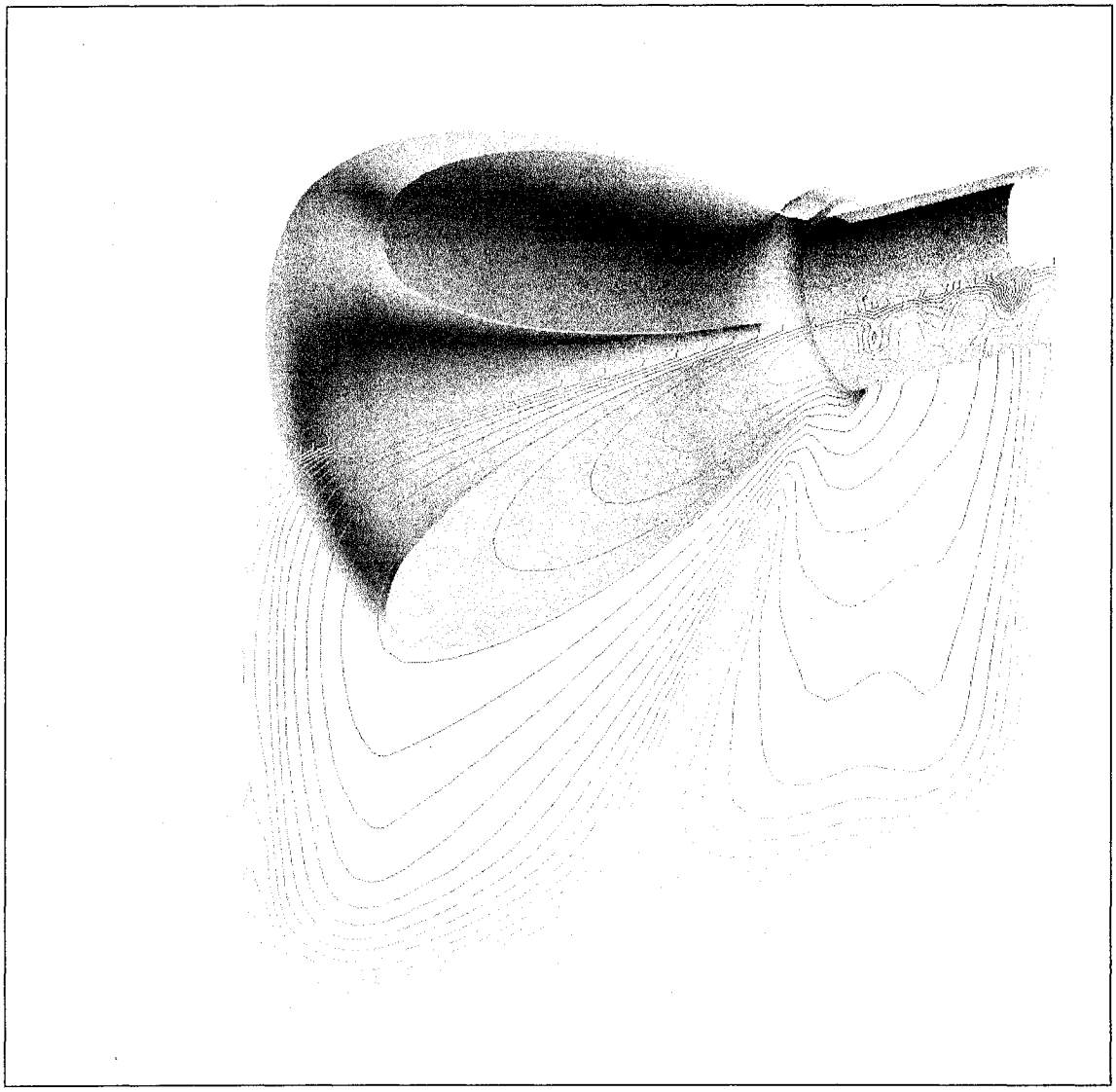

Figure 5.12: SPL isosurface and contour lines for the first azimuthal mode.

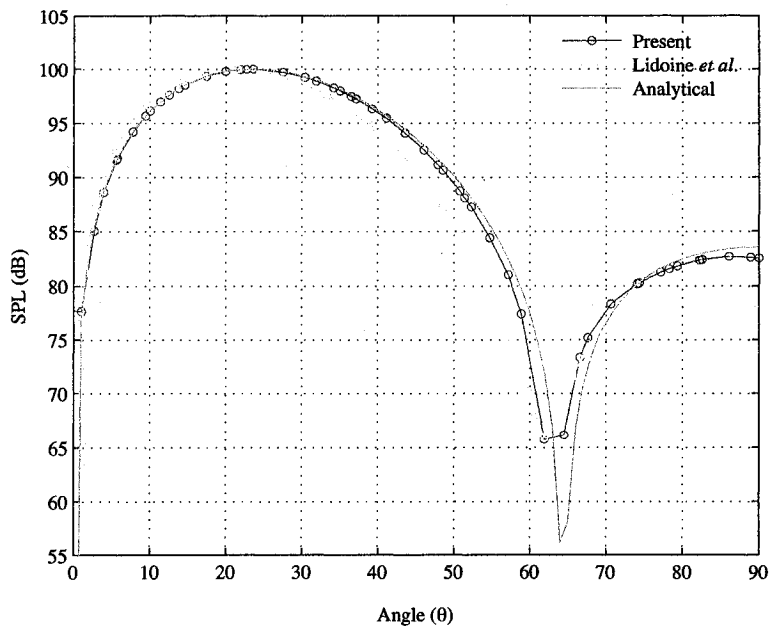

Figure 5.13: Directivity at the far-field for the first azimuthal mode.

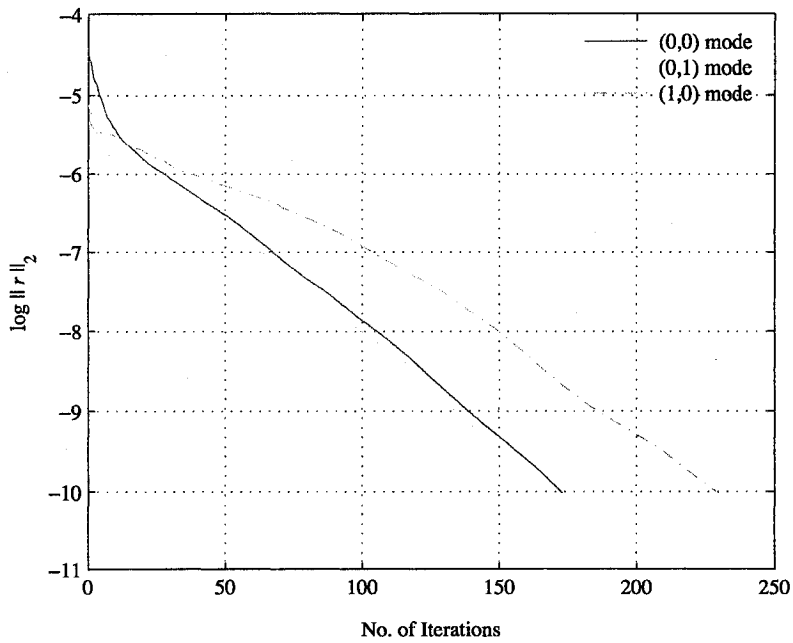

Figure 5.14: Convergence history of the GMRES algorithm for validation tests.

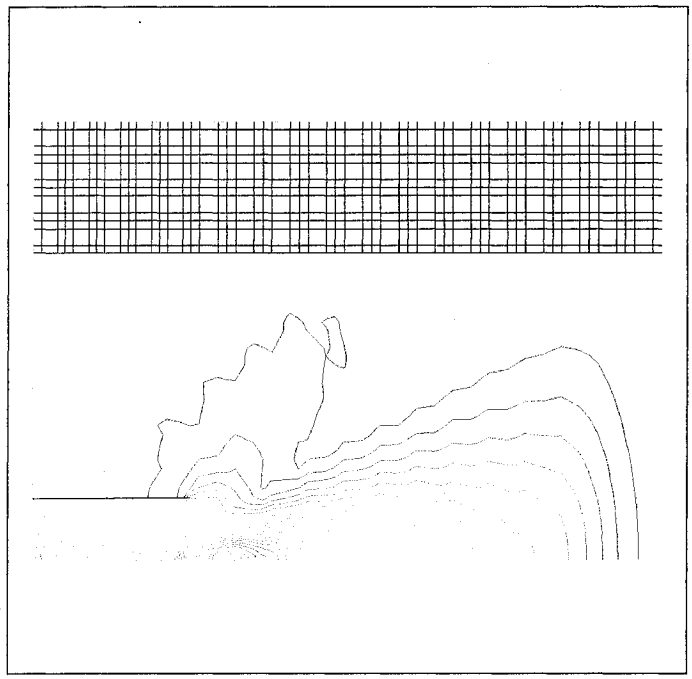

Figure 5.15: Acoustic pressure amplitude contours and spectral grid outline; $N = 3$.

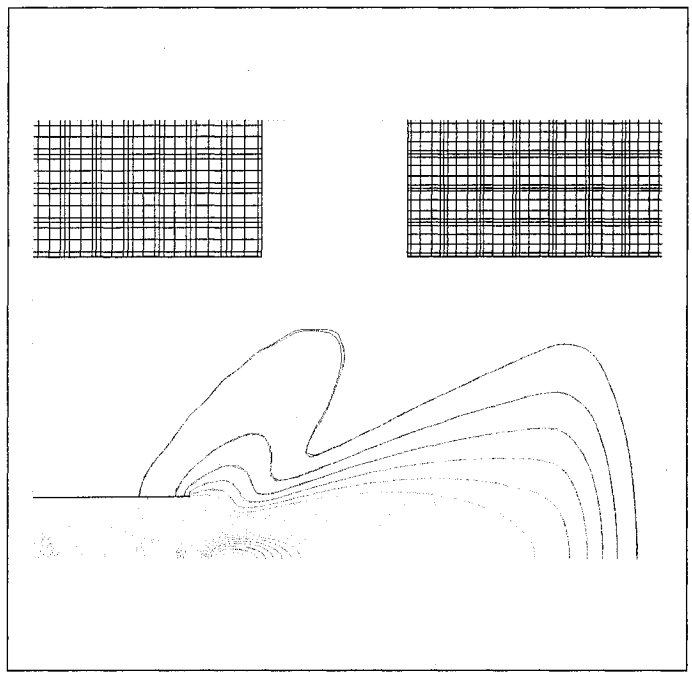

Figure 5.16: Acoustic pressure amplitude contours and spectral grid outline; $N = 4, 5$.

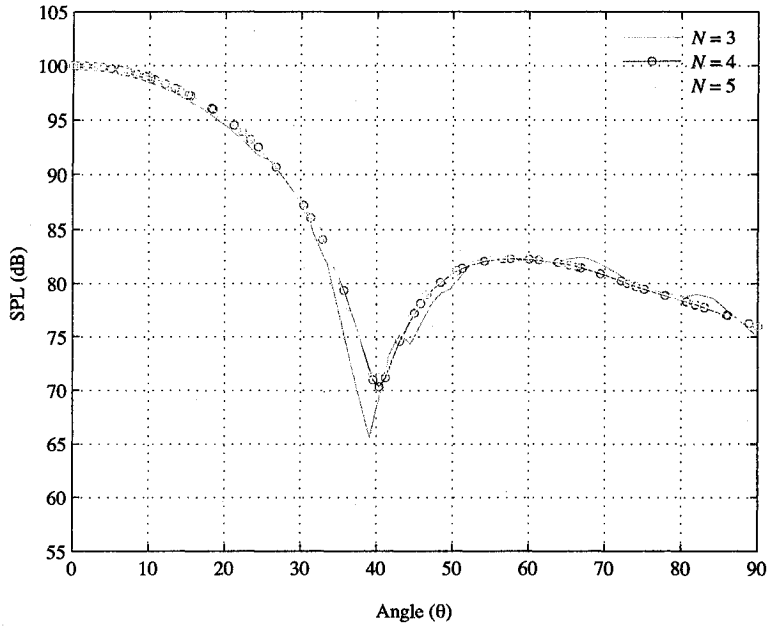

Figure 5.17: Effect of the order of approximation on SPL directivity of the plane wave.

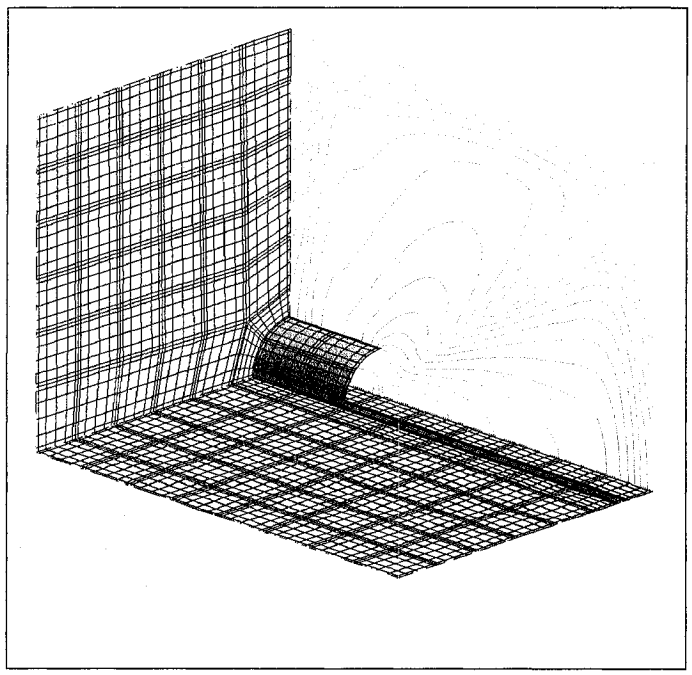

Figure 5.18: Outline of the spectral grid ($N = 6$) and SPL contours for the plane wave radiation.

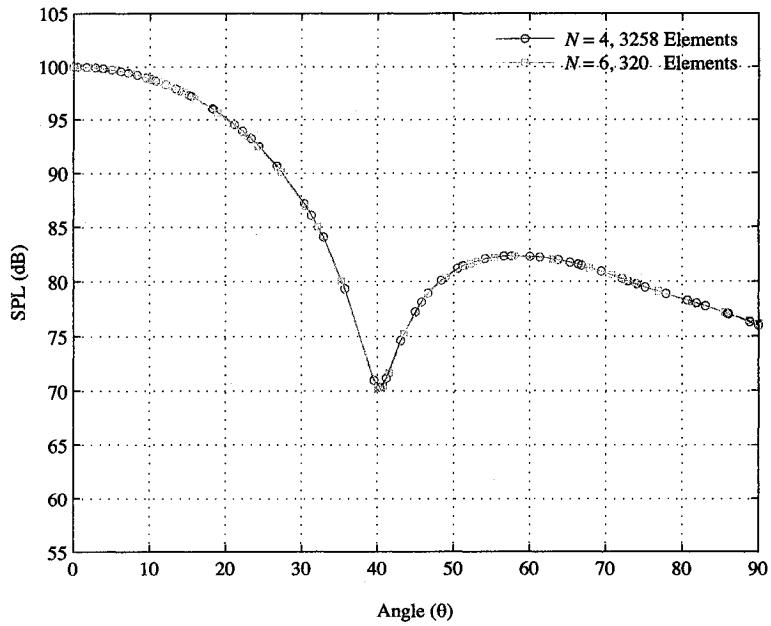

Figure 5.19: Comparison of directivity for a coarse ($N = 6$) and a fine ($N = 4$) spectral grid.

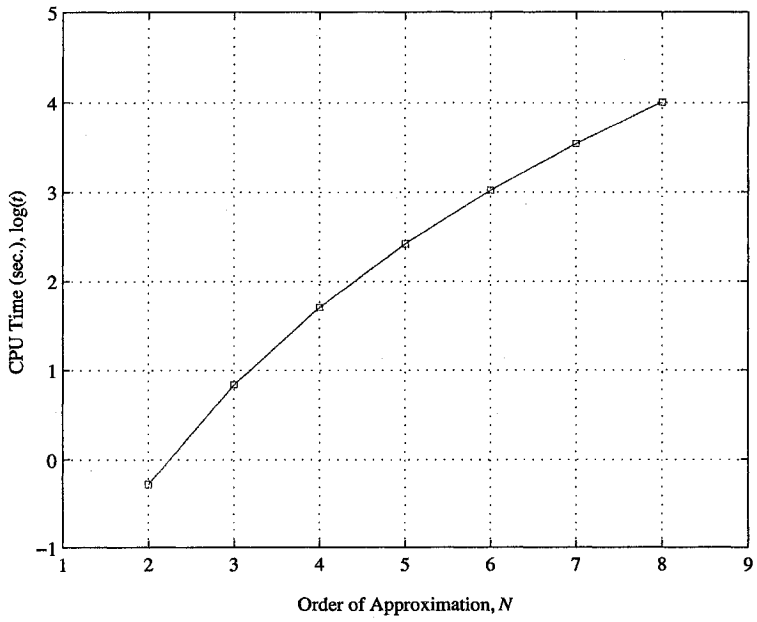

Figure 5.20: Exponential growth of the computation time for building the matrix vs. N.

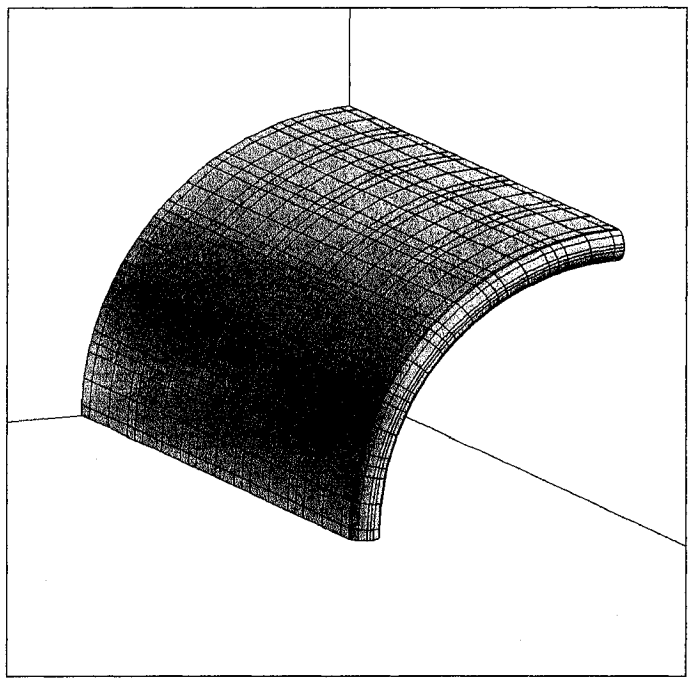

Figure 5.21: Close-up of the cylindrical duct with a lip thickness of $\frac{1}{8}R$.

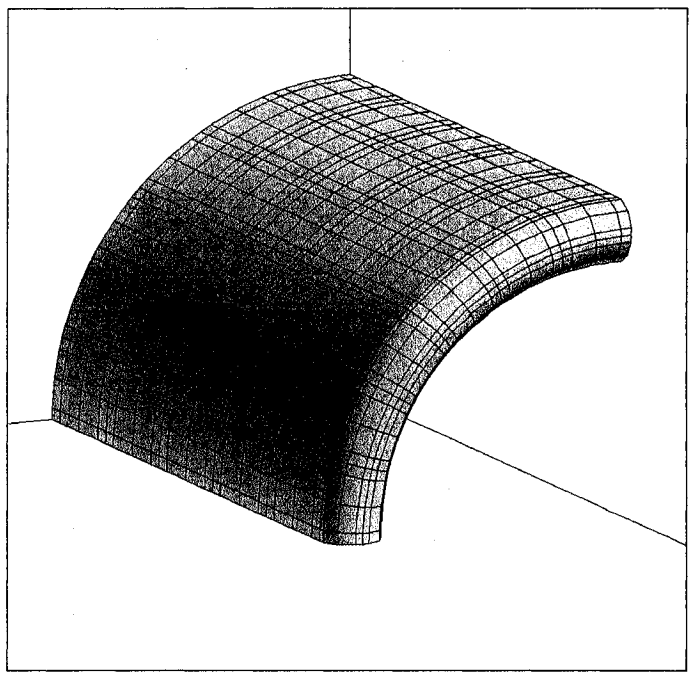

Figure 5.22: Close-up of the cylindrical duct with a lip thickness of $\frac{1}{4}R$.

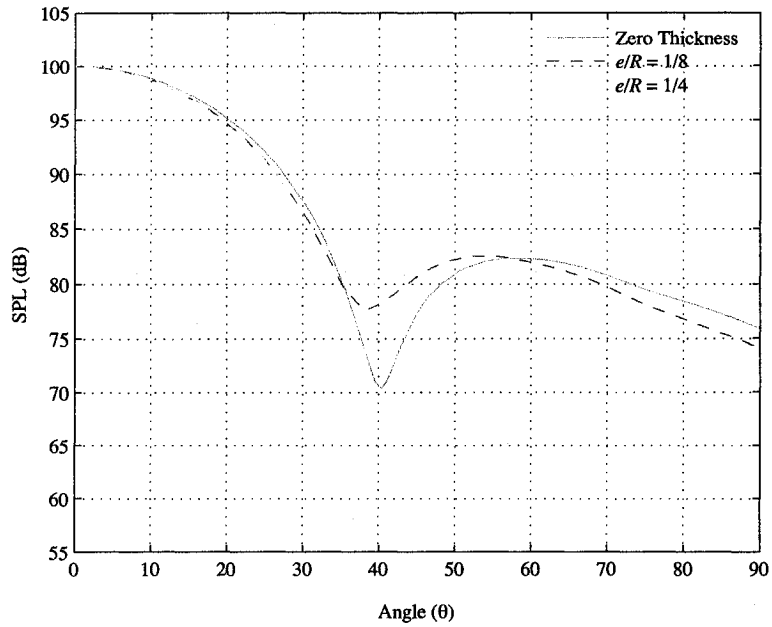

Figure 5.23: Effect of the lip thickness on directivity of the plane wave.

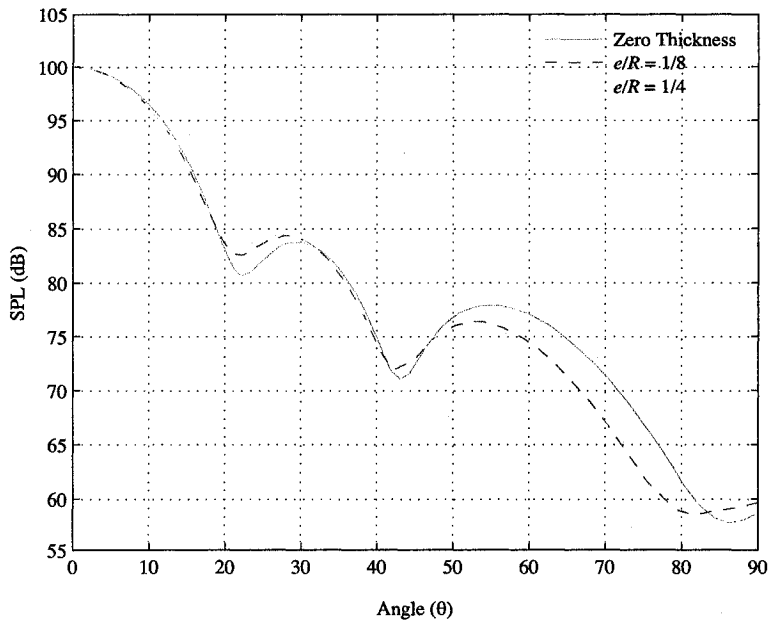

Figure 5.24: Effect of the lip thickness on directivity at a higher frequency.

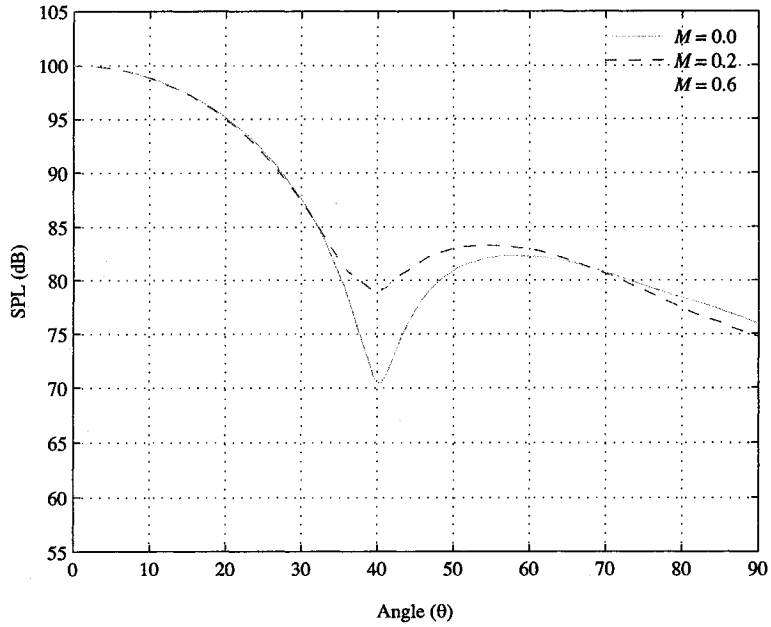

Figure 5.25: Effect of the uniform external flow on SPL directivity.

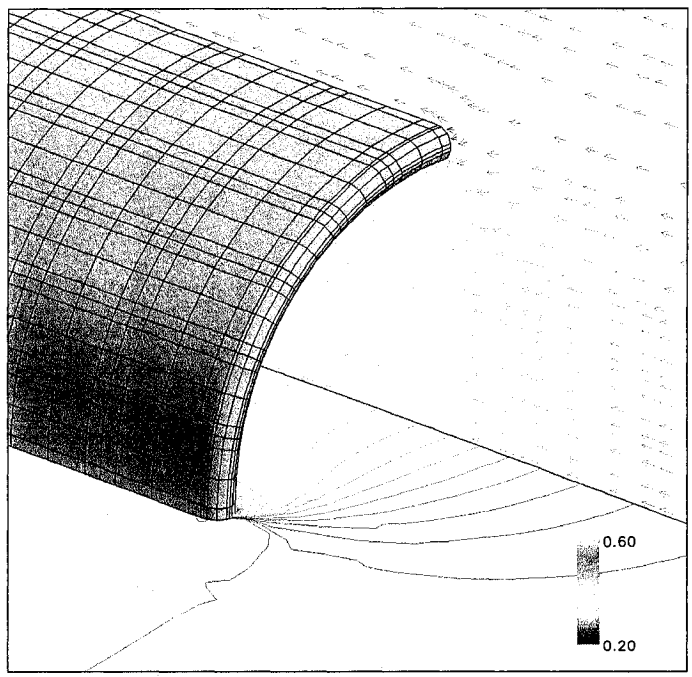

Figure 5.26: Velocity field and Mach contours around a cylinder with a lip thickness of $\frac{1}{8}R$; potential flow solution ($M_\infty = 0.2$).

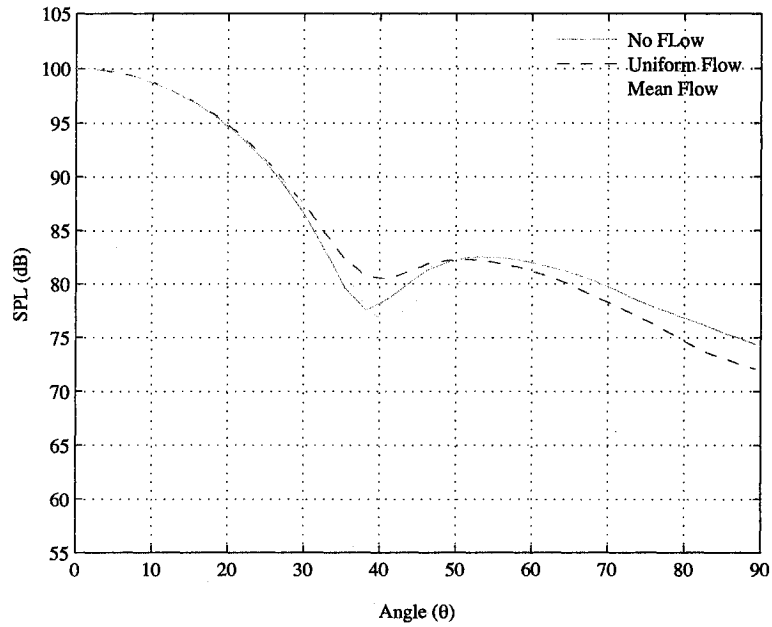

Figure 5.27: Effect of the uniform and mean flow on SPL directivity.

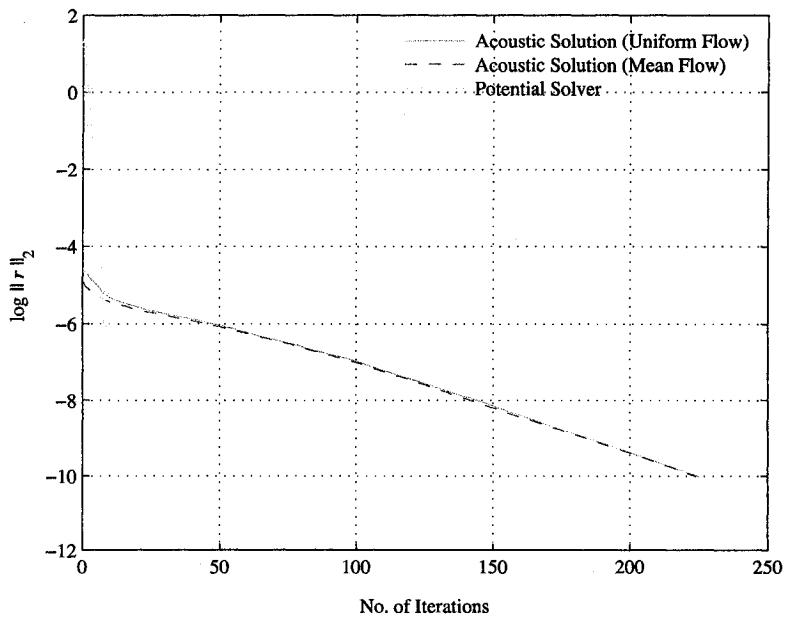

Figure 5.28: Convergence history of the acoustic and mean flow potential solvers.

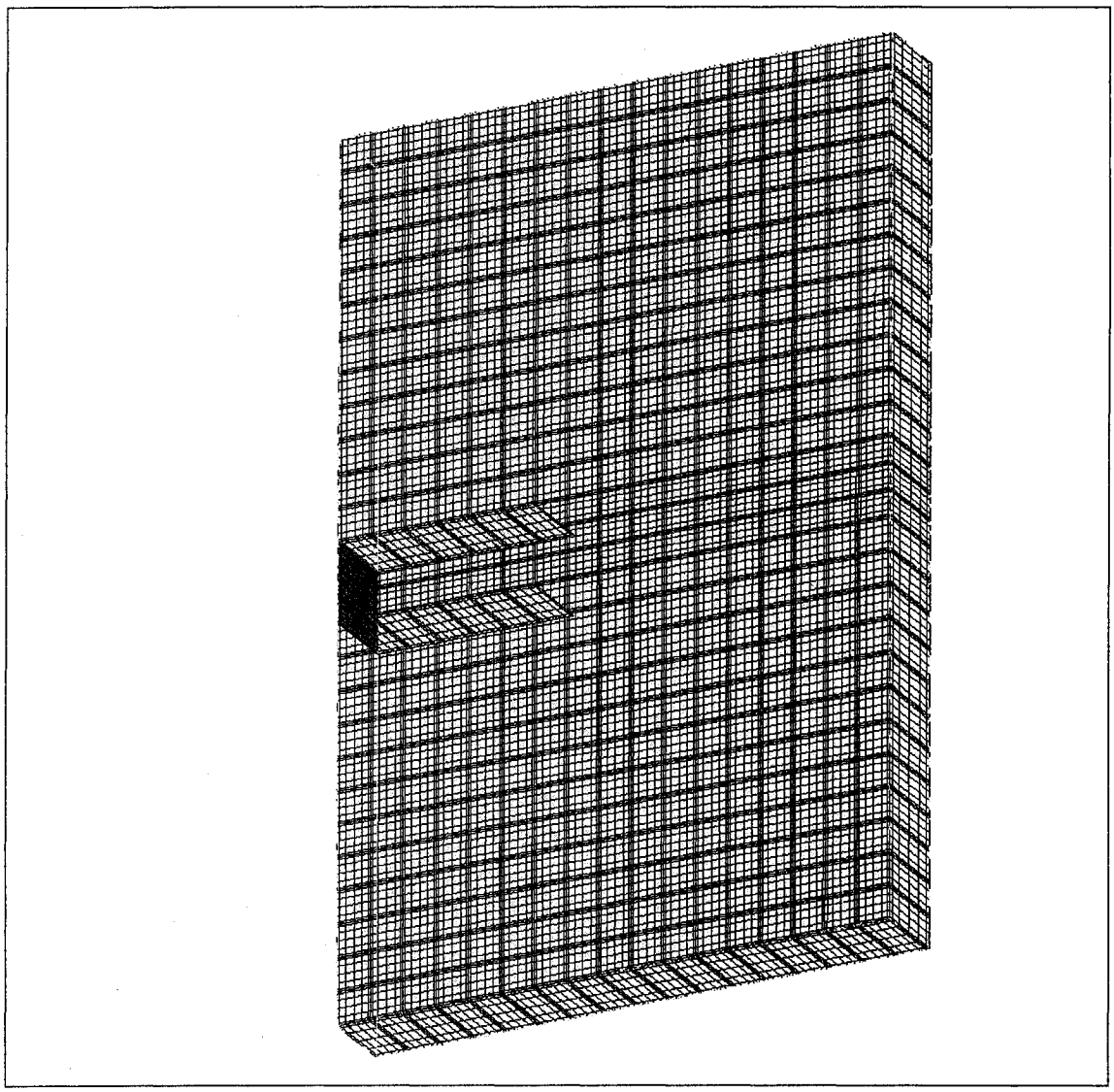

Figure 5.29: Computational domain and spectral grid outline for the semi-infinite duct.

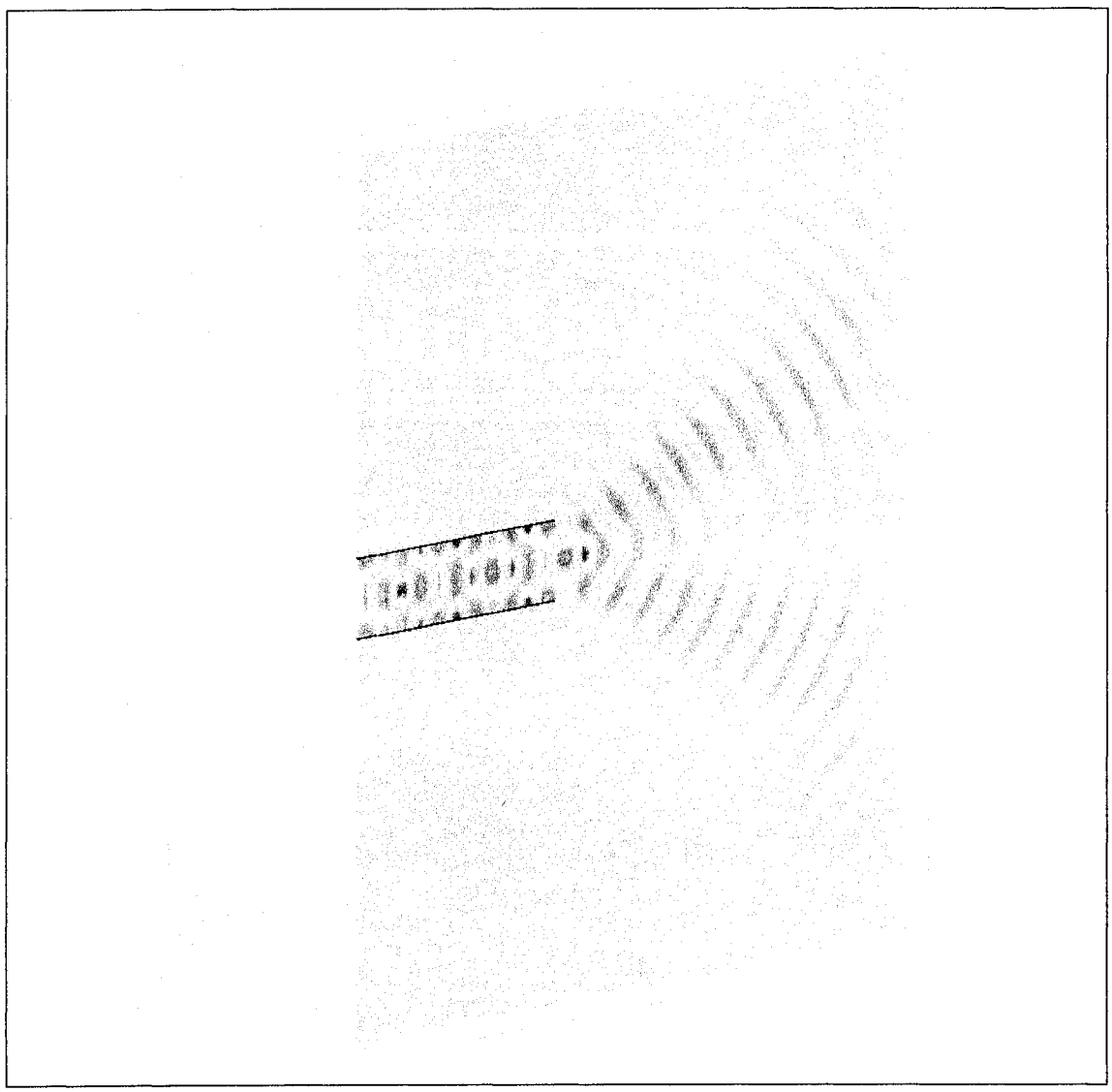

Figure 5.30: Acoustic pressure contours for the 2D duct; second cross mode.

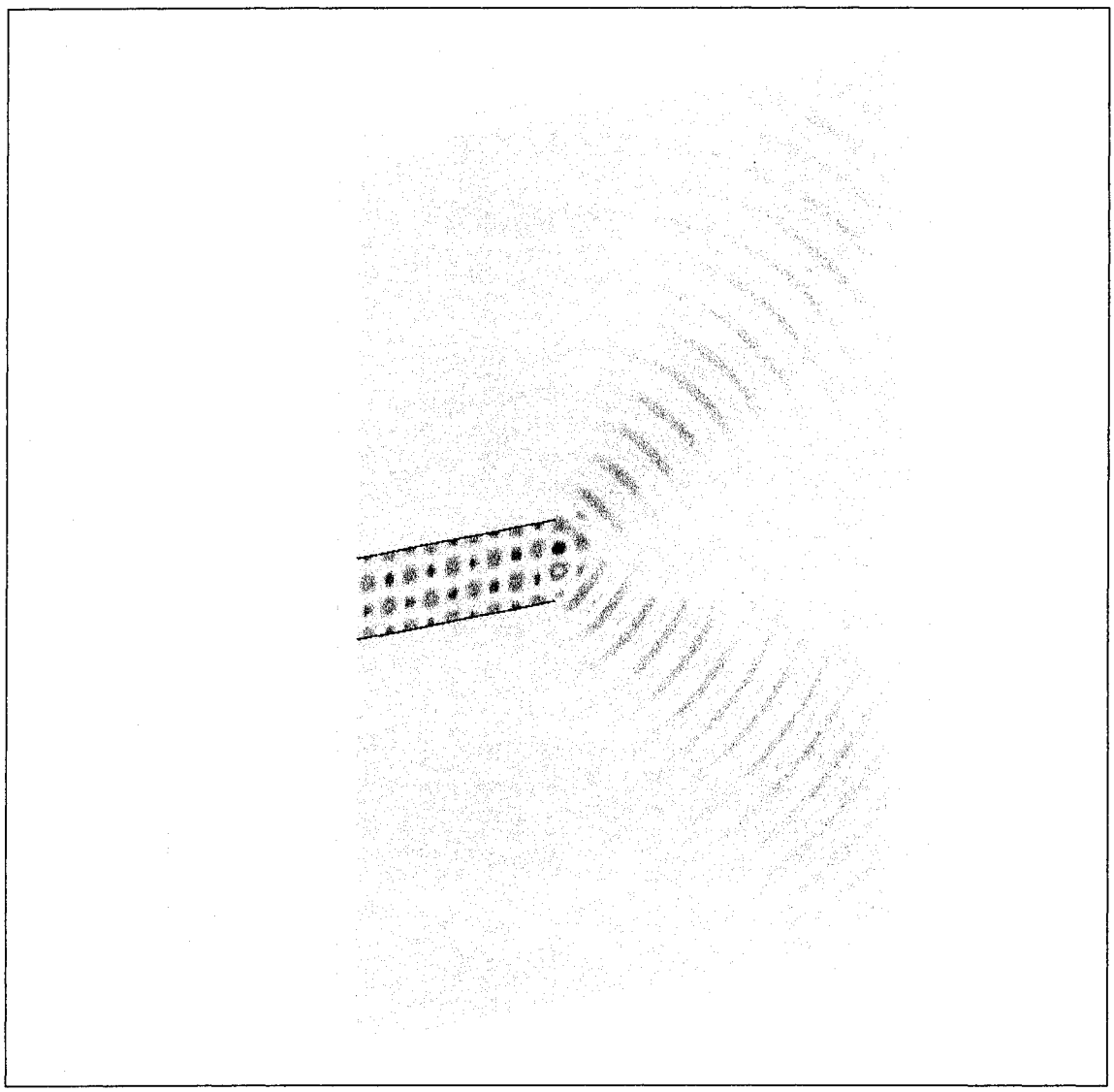

Figure 5.31: Acoustic pressure contours for the 2D duct; third cross mode.

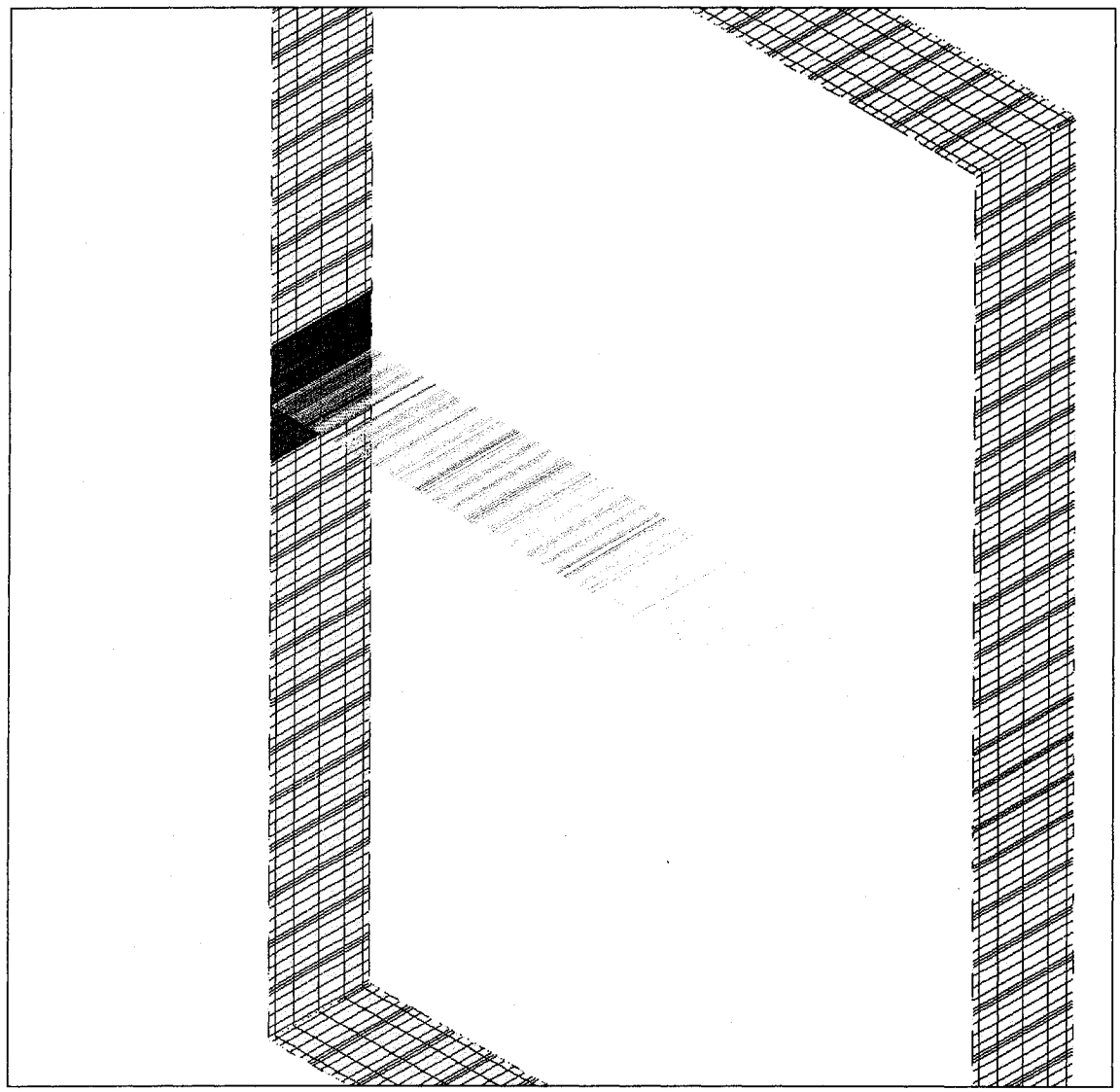

Figure 5.32: Acoustic pressure contours at $y = 0$, in the periodic direction.

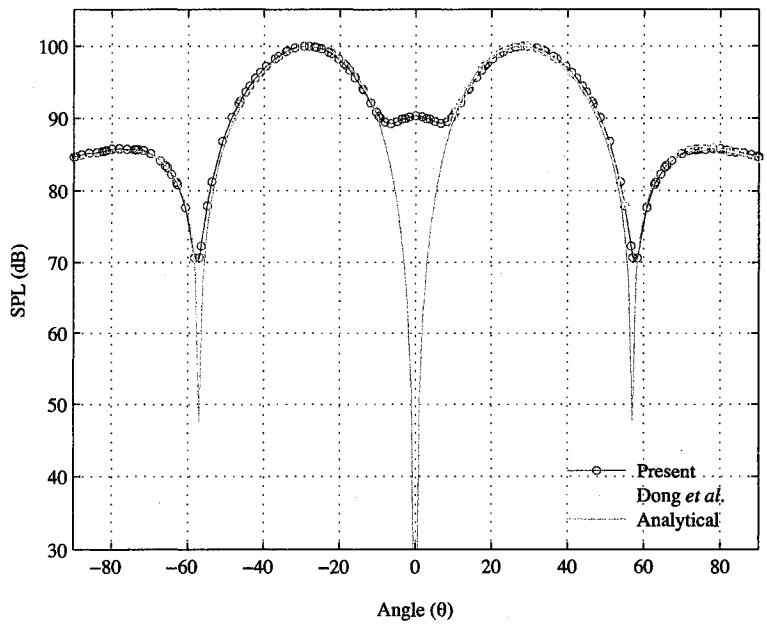

Figure 5.33: Directivity for the second cross mode; comparison with analytical and numerical data.

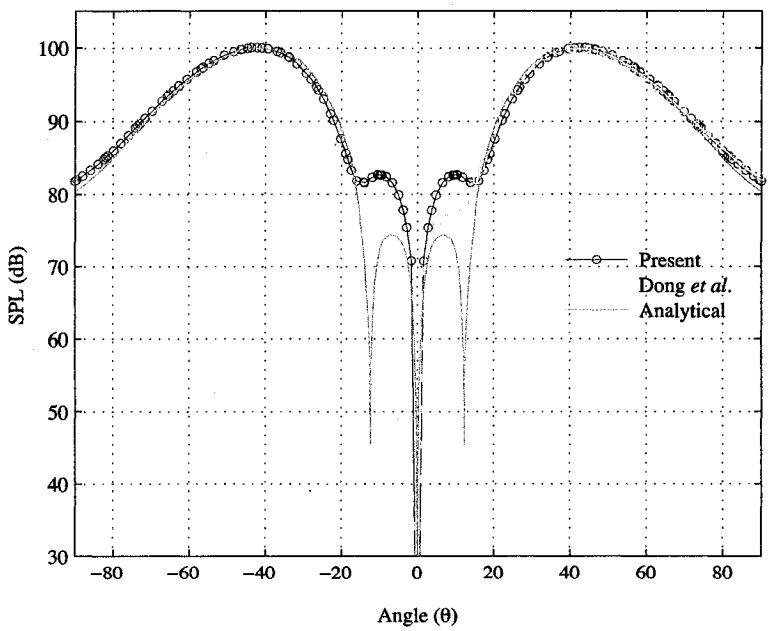

Figure 5.34: Directivity for the third cross mode; comparison with analytical and numerical data.

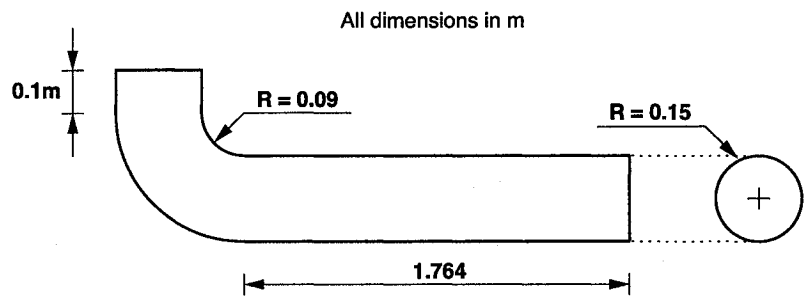

Figure 5.35: Geometry of the curved duct.

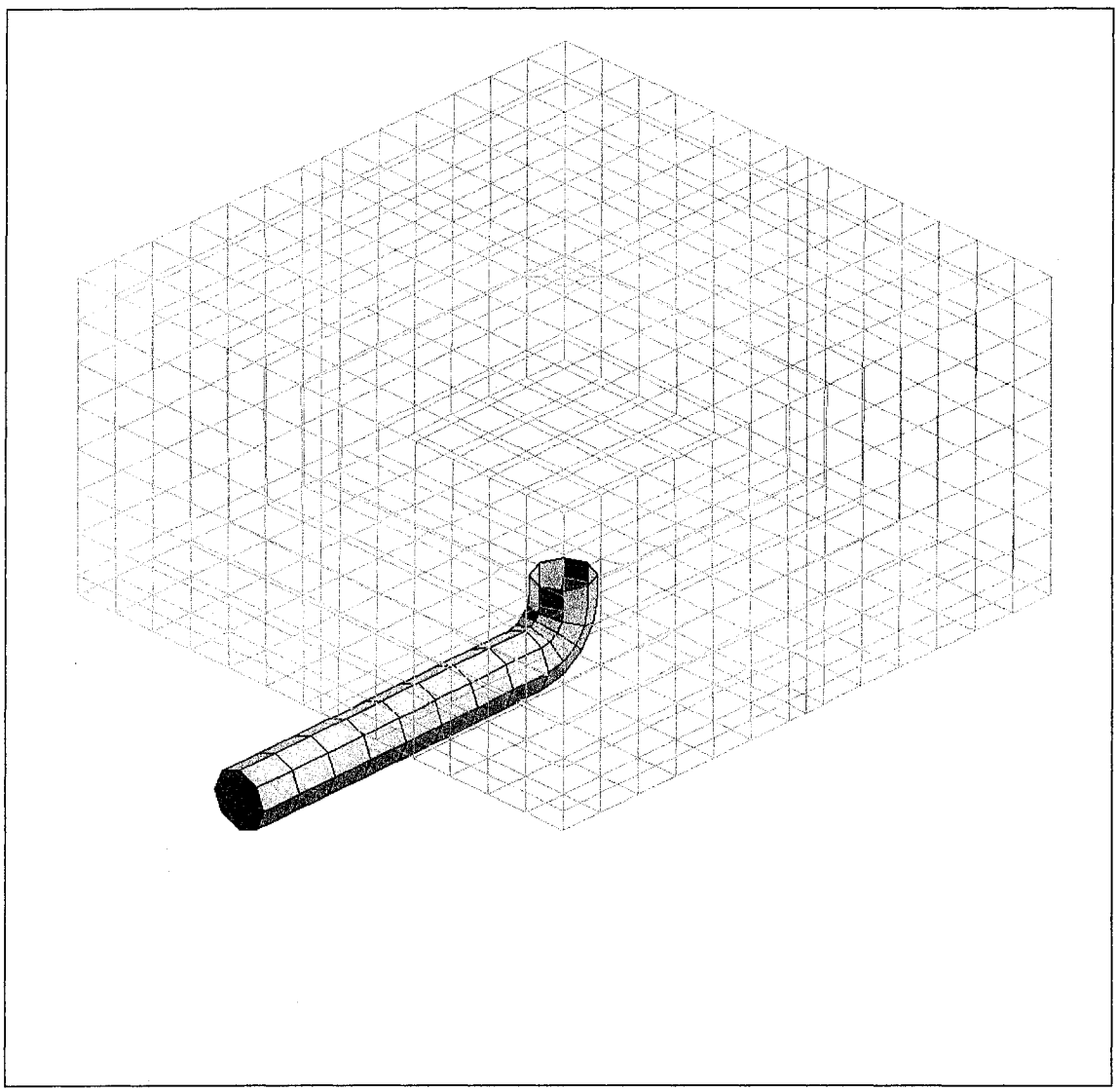

Figure 5.36: Computational domain showing the finite element background grid.

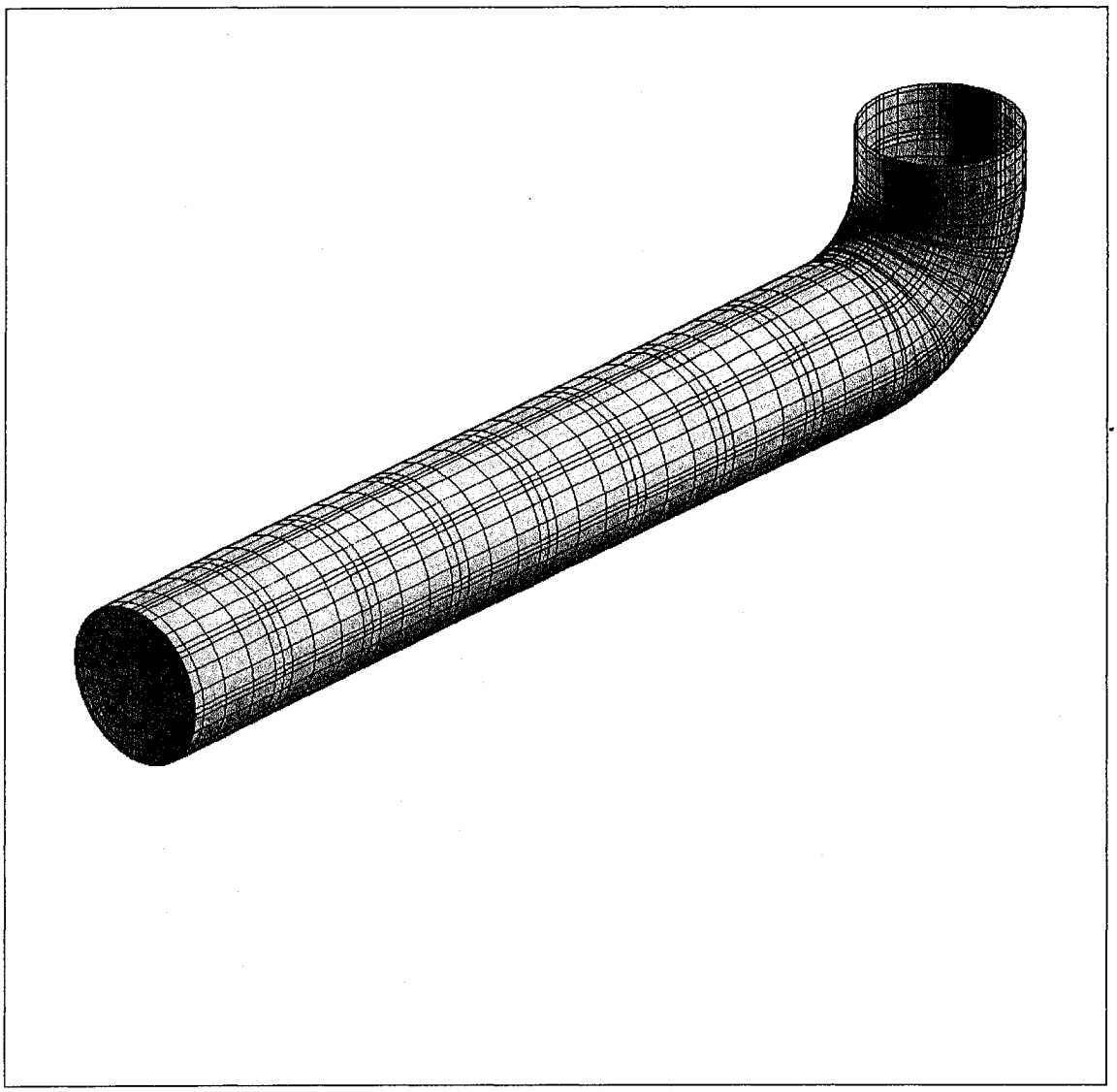

Figure 5.37: Close-up of the spectral grid around the curved duct after CAD projection.

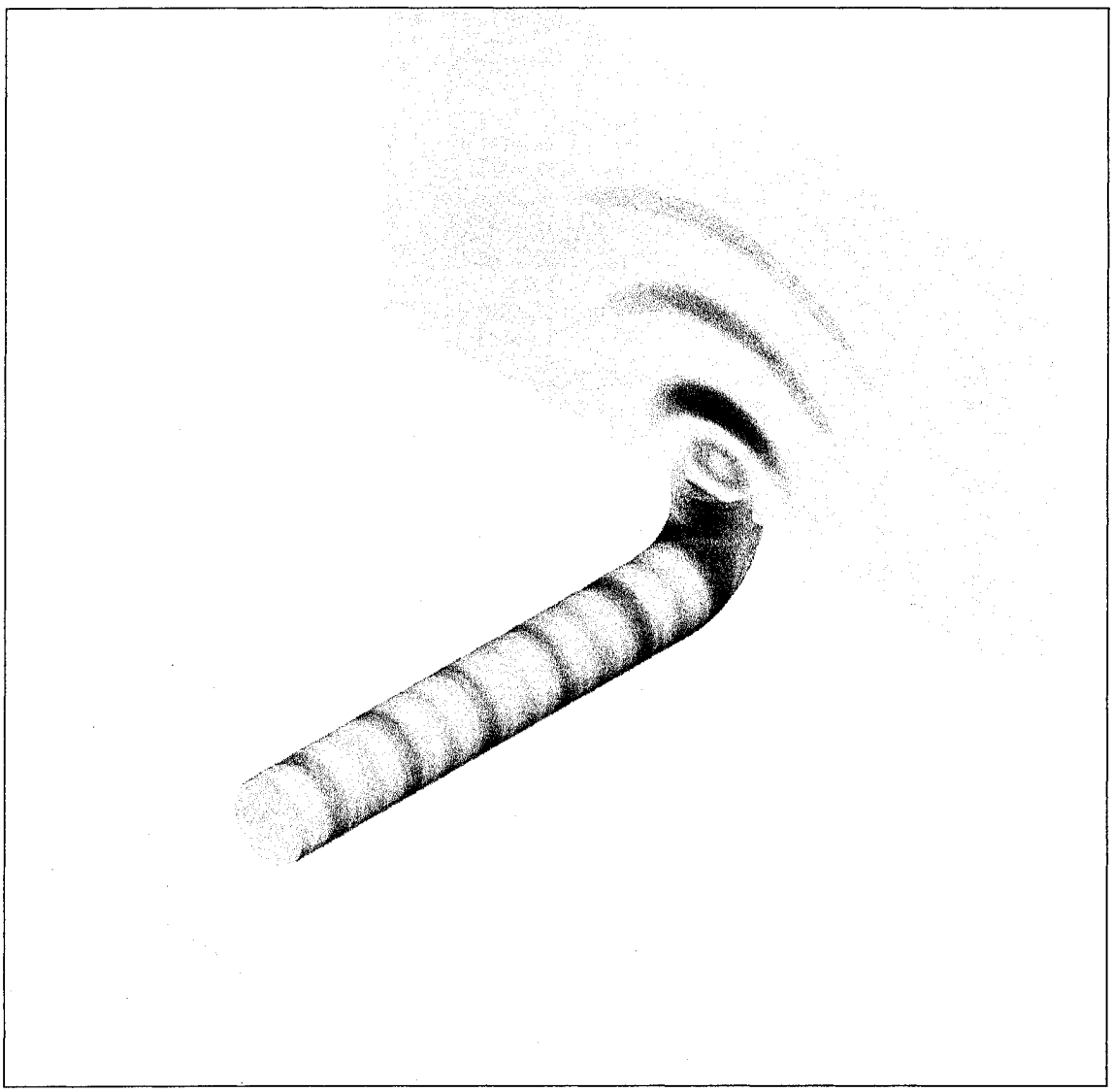

Figure 5.38: Acoustic pressure distribution in the duct and on the exit plane; plane wave radiation.

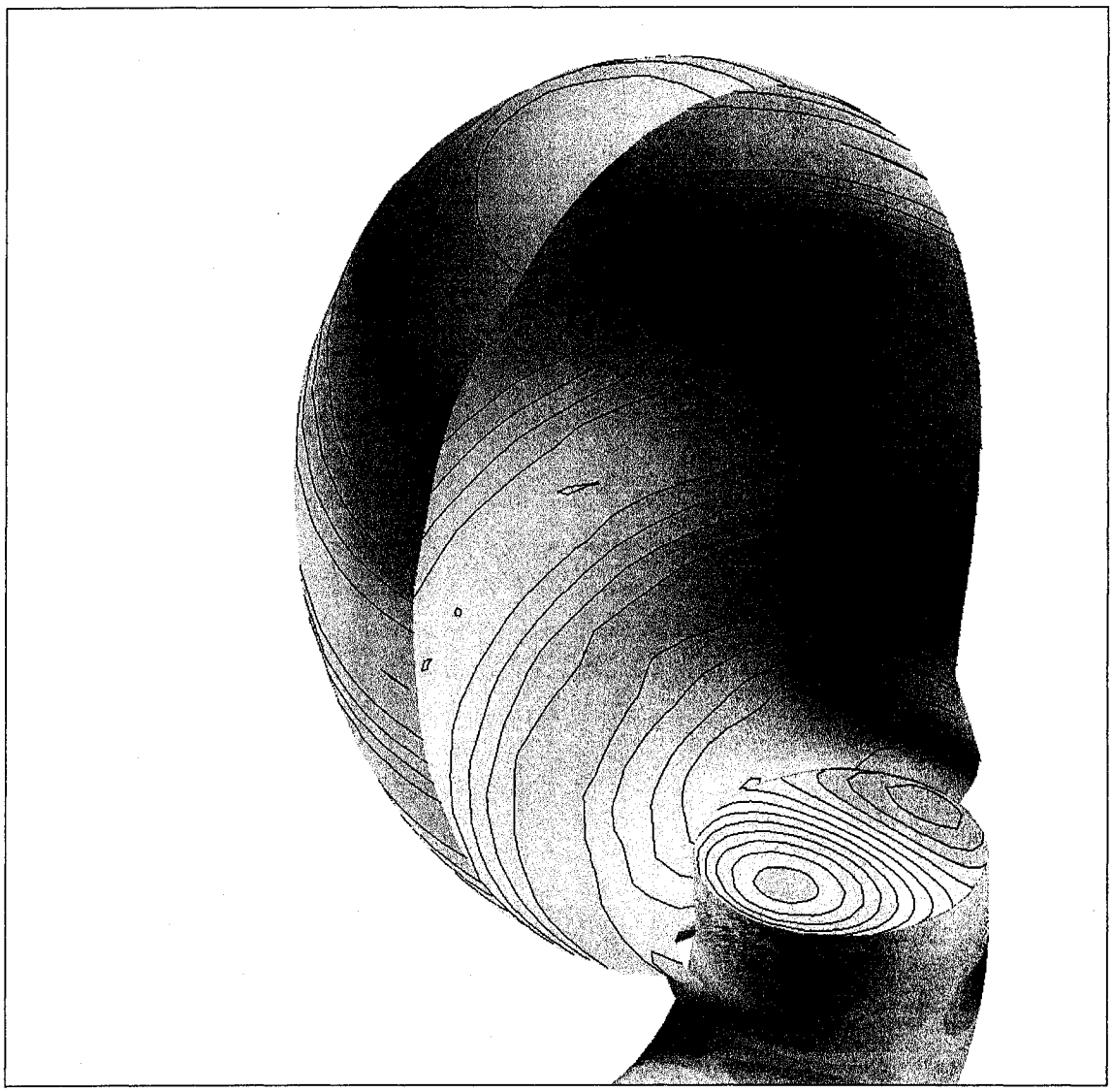

Figure 5.39: Isosurface of SPL and acoustic pressure contour lines at the duct exit; plane wave radiation.

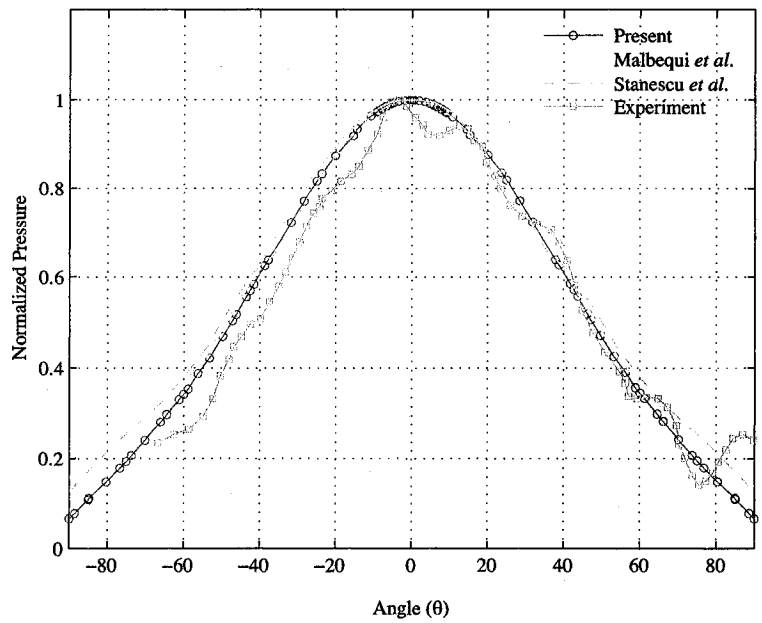

Figure 5.40: Pressure directivity of the plane wave radiation at $x = 2.004$; $f = 1000$Hz.

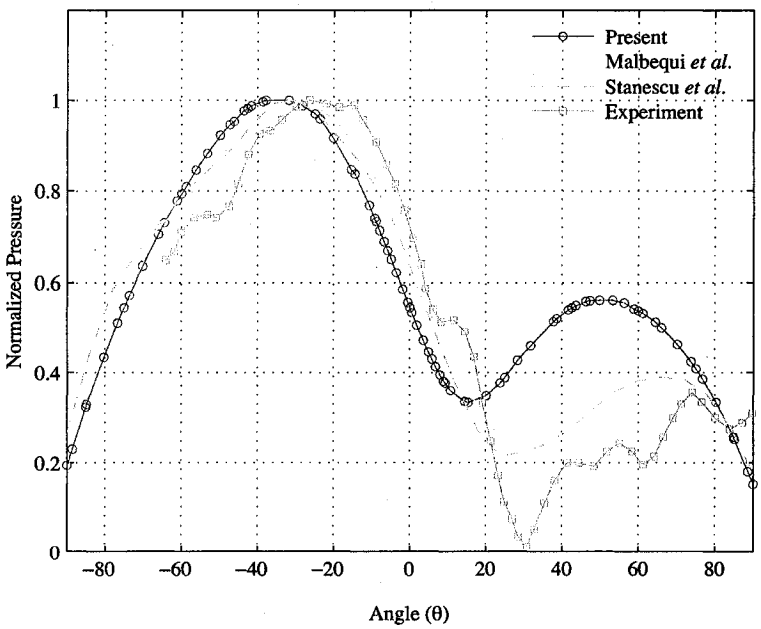

Figure 5.41: Pressure directivity of the $(m, s) = (1, 0)$ mode at $x = 2.004$; $f = 1000$Hz.

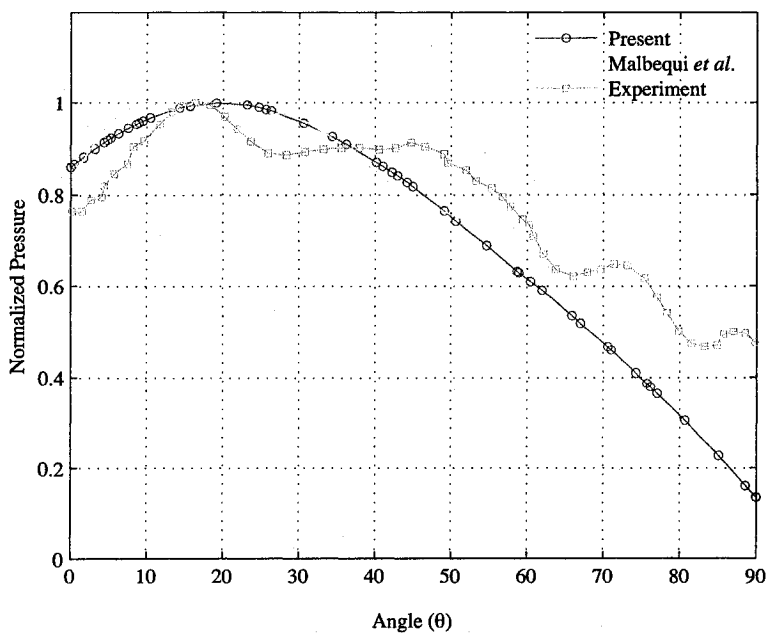

Figure 5.42: Pressure directivity of the $(m, s) = (1, 0)$ mode on the xy-plane; $f = 1000$Hz.

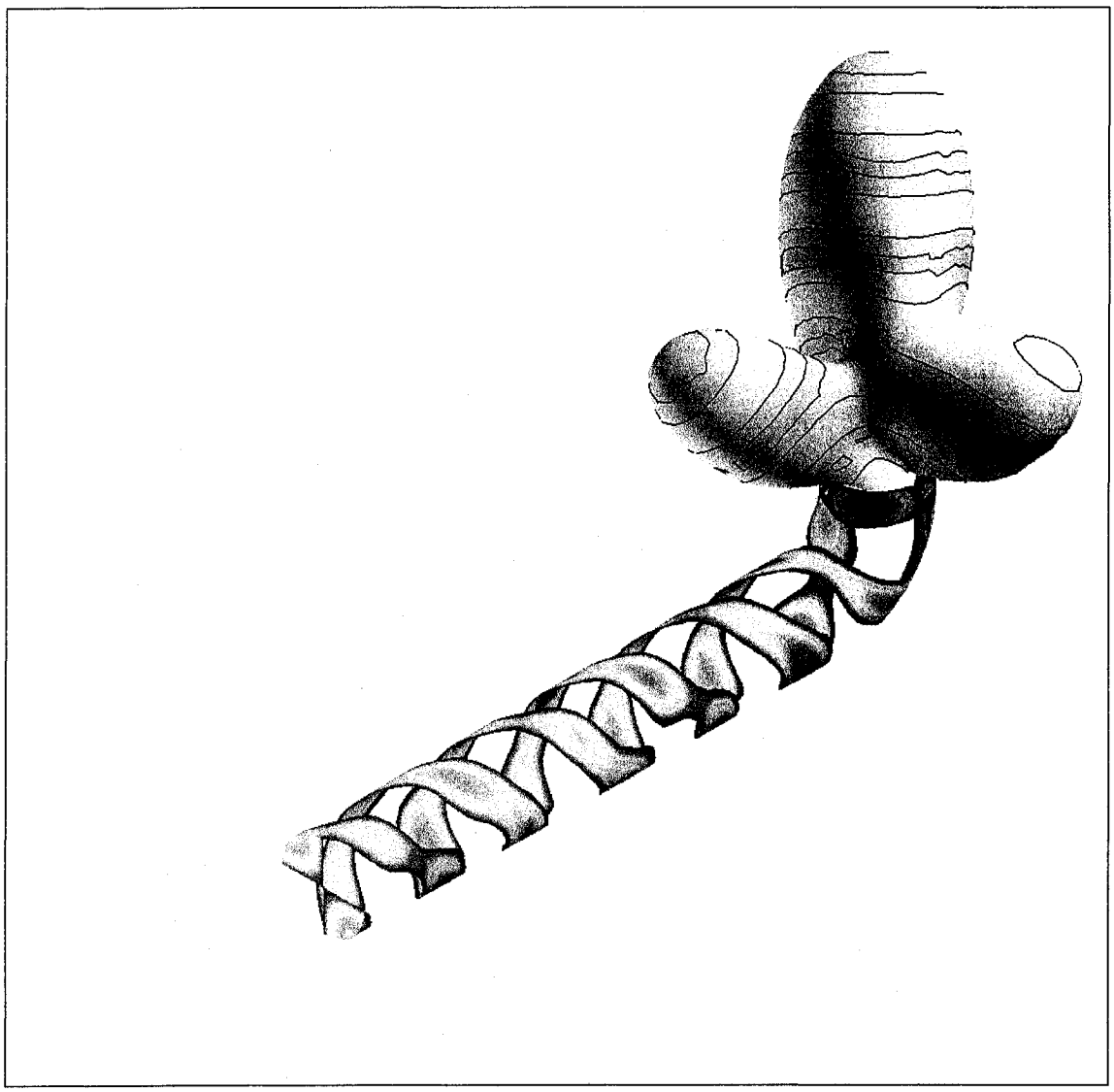

Figure 5.43: Acoustic pressure contours in the duct and SPL isosurface at the exit plane for the second azimuthal mode; $f = 1500$Hz.

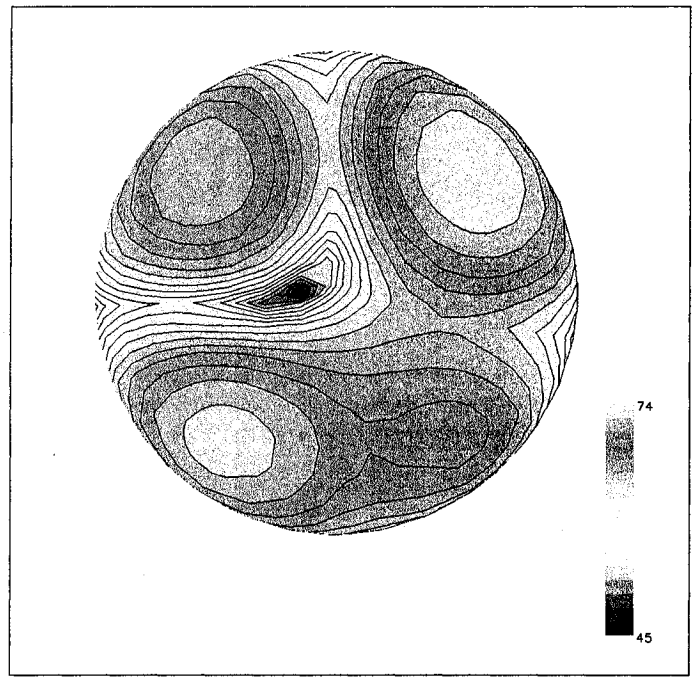

Figure 5.44: SPL contours at the duct exit plane ($y = 0.34$) for the second azimuthal mode; $f = 1500$Hz.

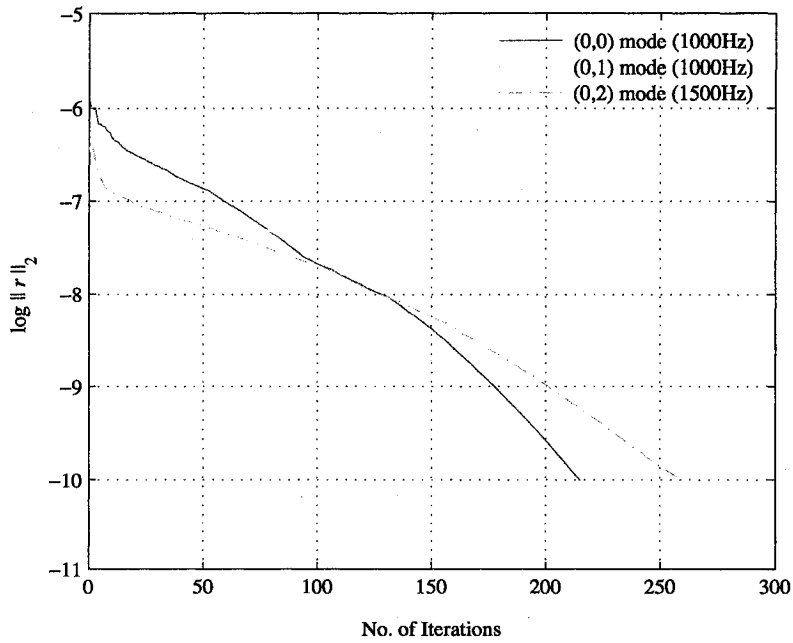

Figure 5.45: Convergence history of the GMRES algorithm for different tests.

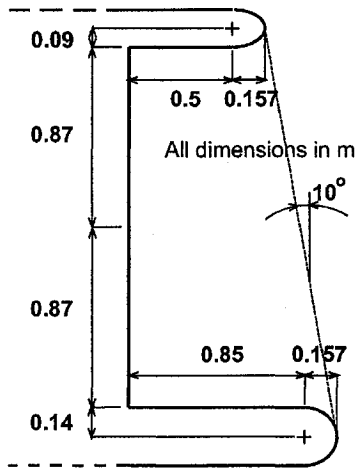

Figure 5.46: Geometry of the scarfed inlet.

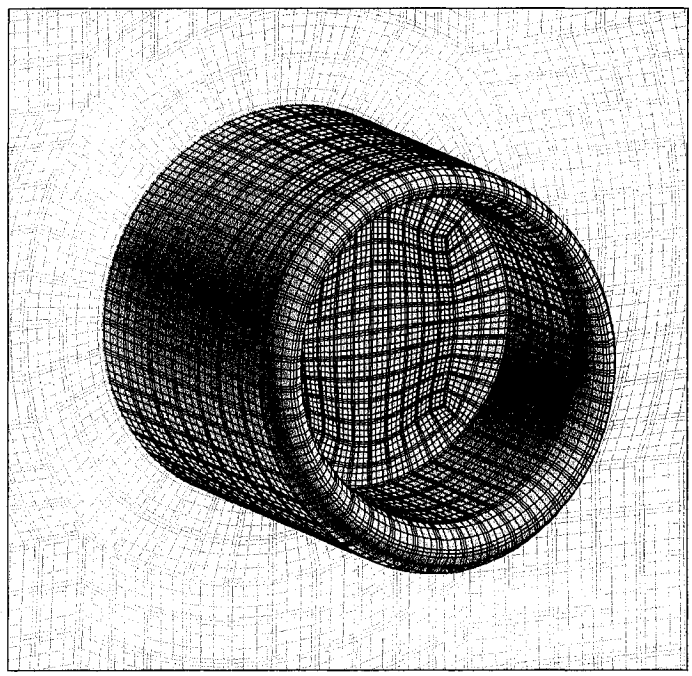

Figure 5.47: Outline of the spectral mesh in and around the nacelle.

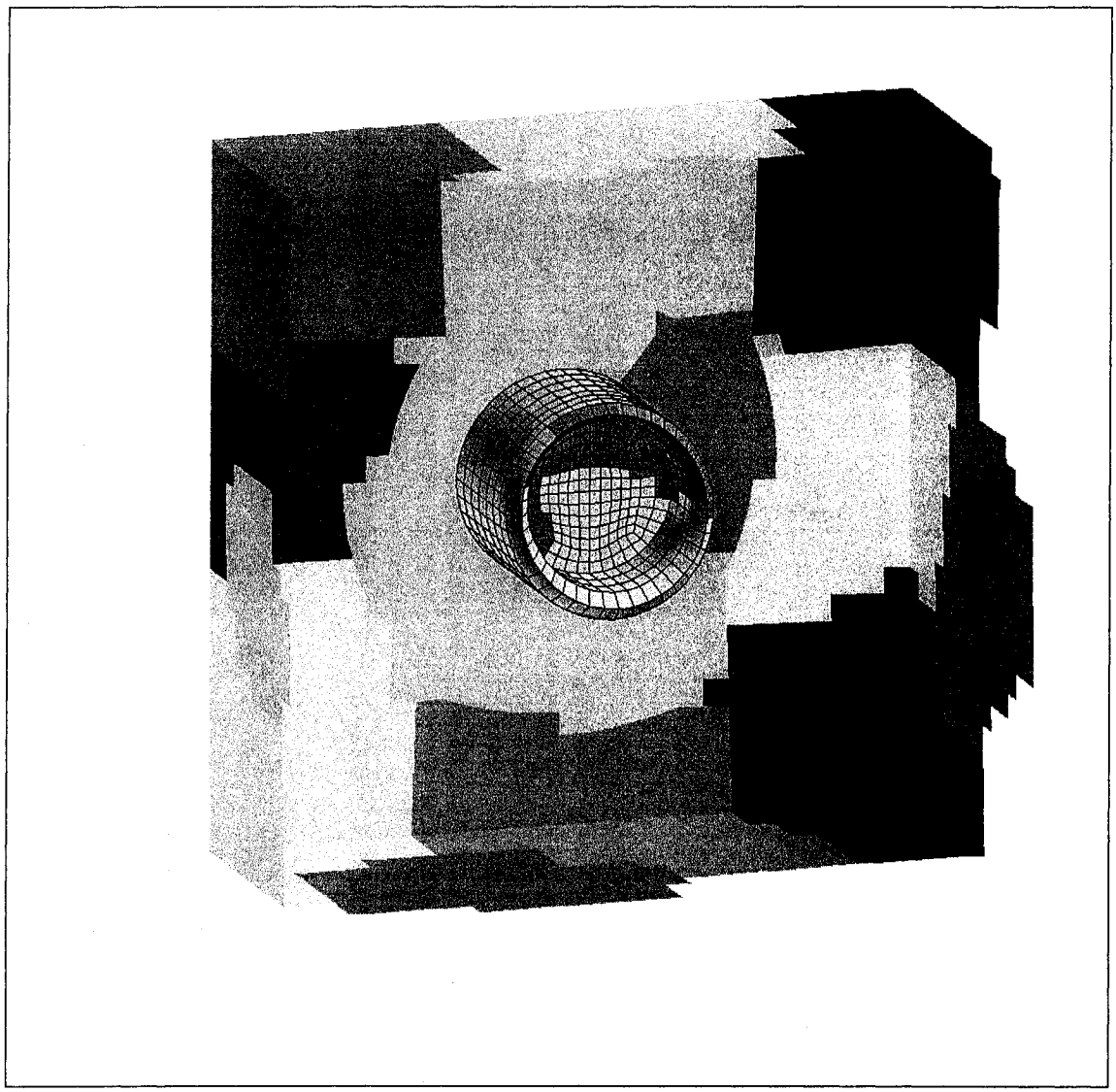

Figure 5.48: Typical nonoverlapping partitioning of the computational domain into smaller subdomains.

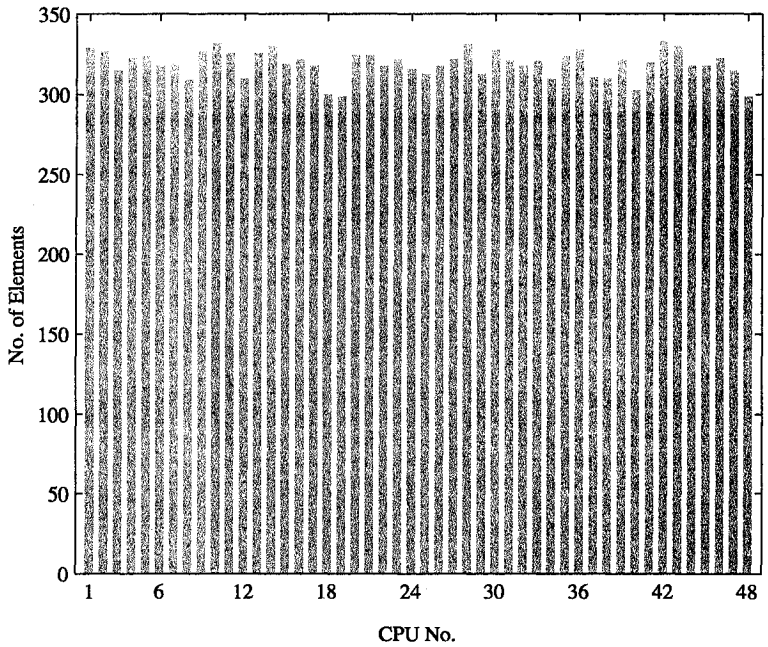

Figure 5.49: Element distribution among subdomains for a 48-CPU partitioning.

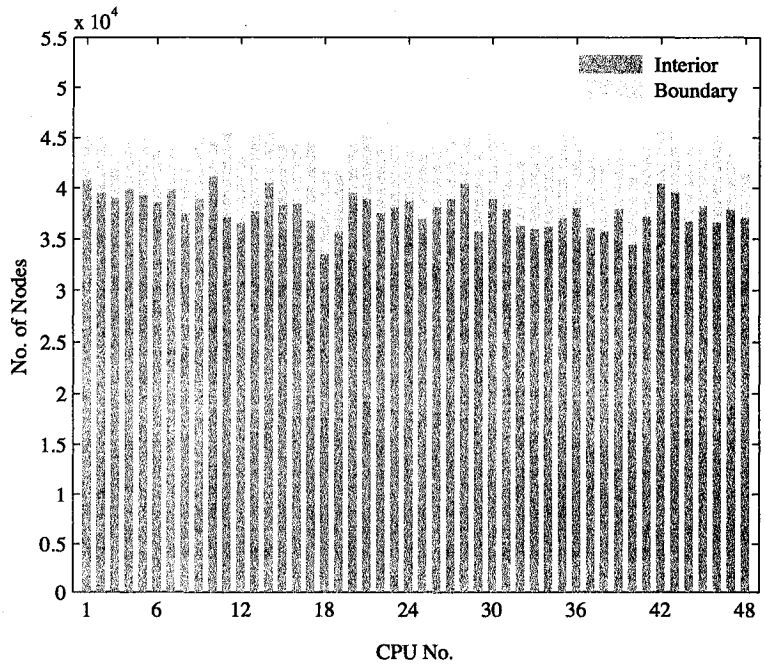

Figure 5.50: Node distribution among subdomains for a 48-CPU partitioning.

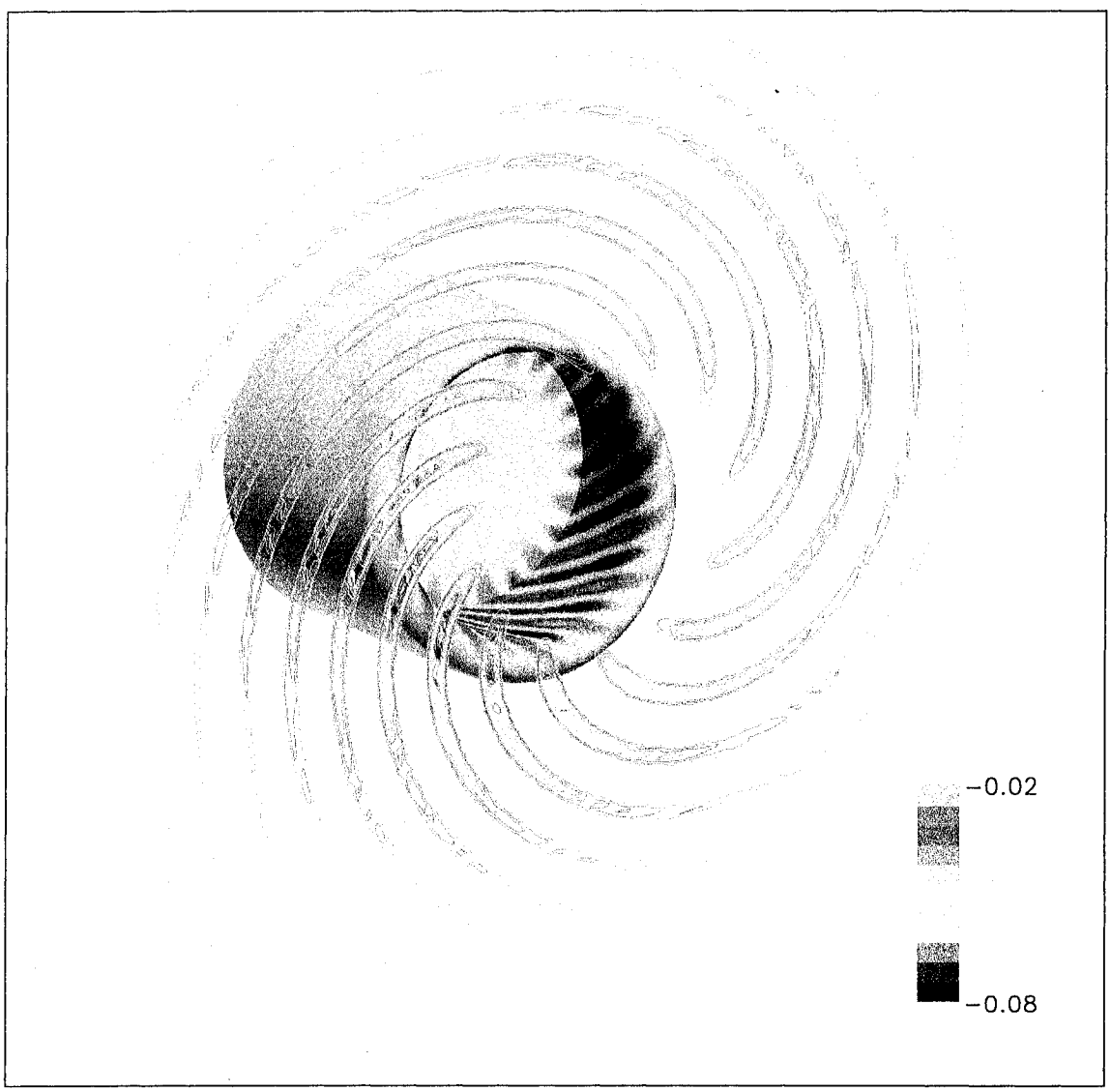

Figure 5.51: Acoustic pressure contours inside the nacelle and at an axial location away from the inlet.

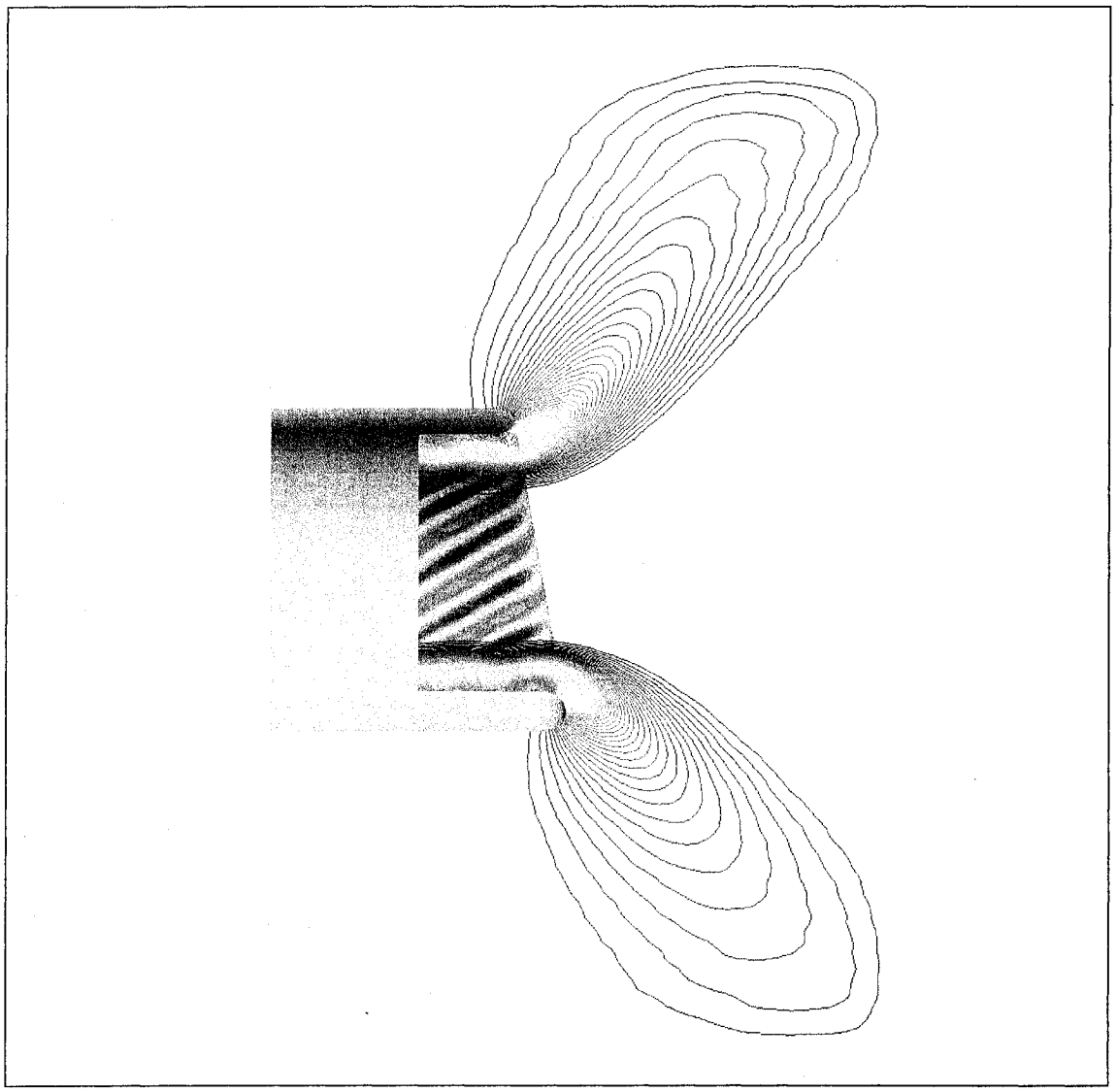

Figure 5.52: Pressure amplitude contours along the duct's plane of symmetry.

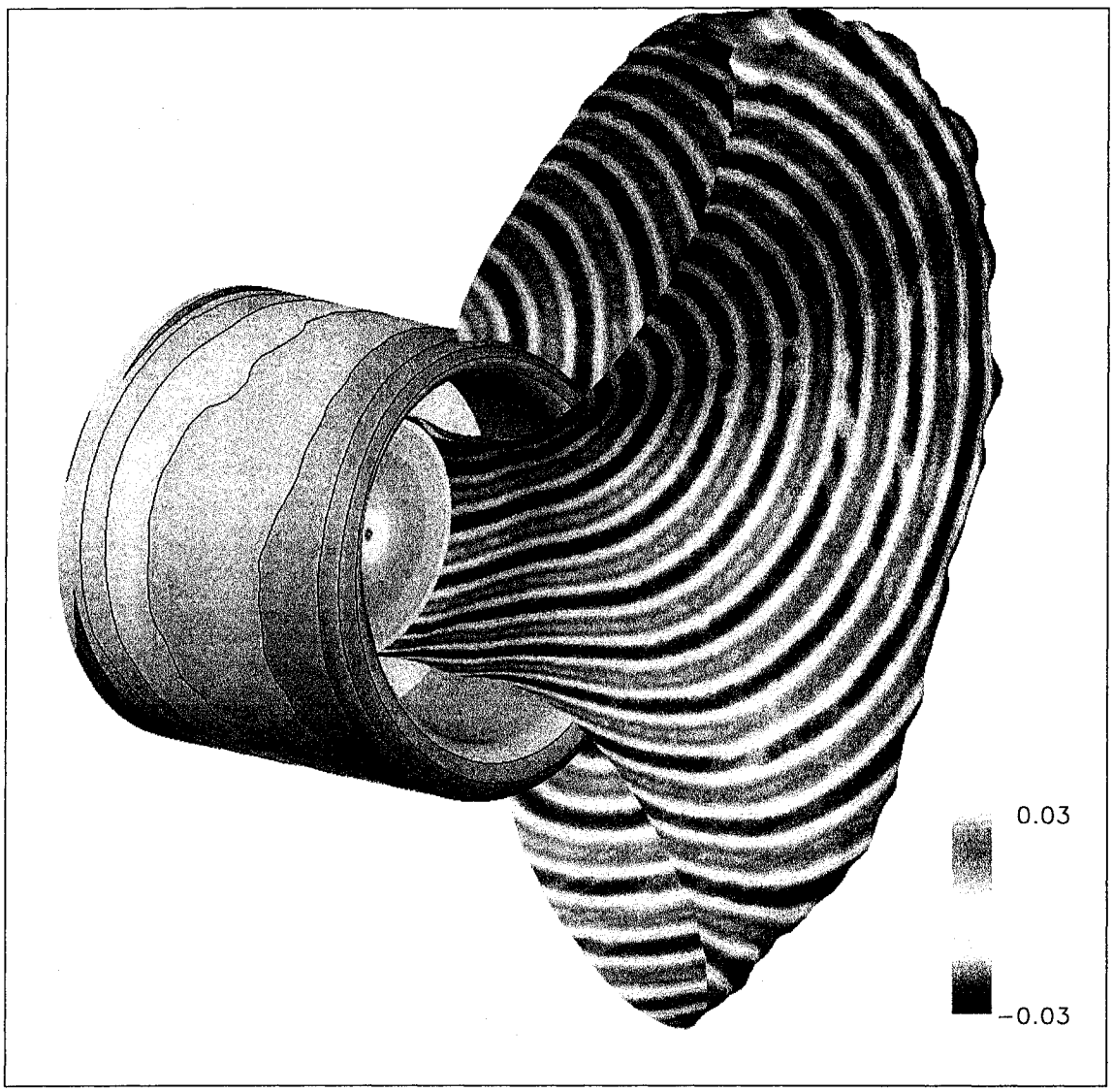

Figure 5.53: Acoustic pressure contours superimposed on the isosurface of SPL at the necelle exit.

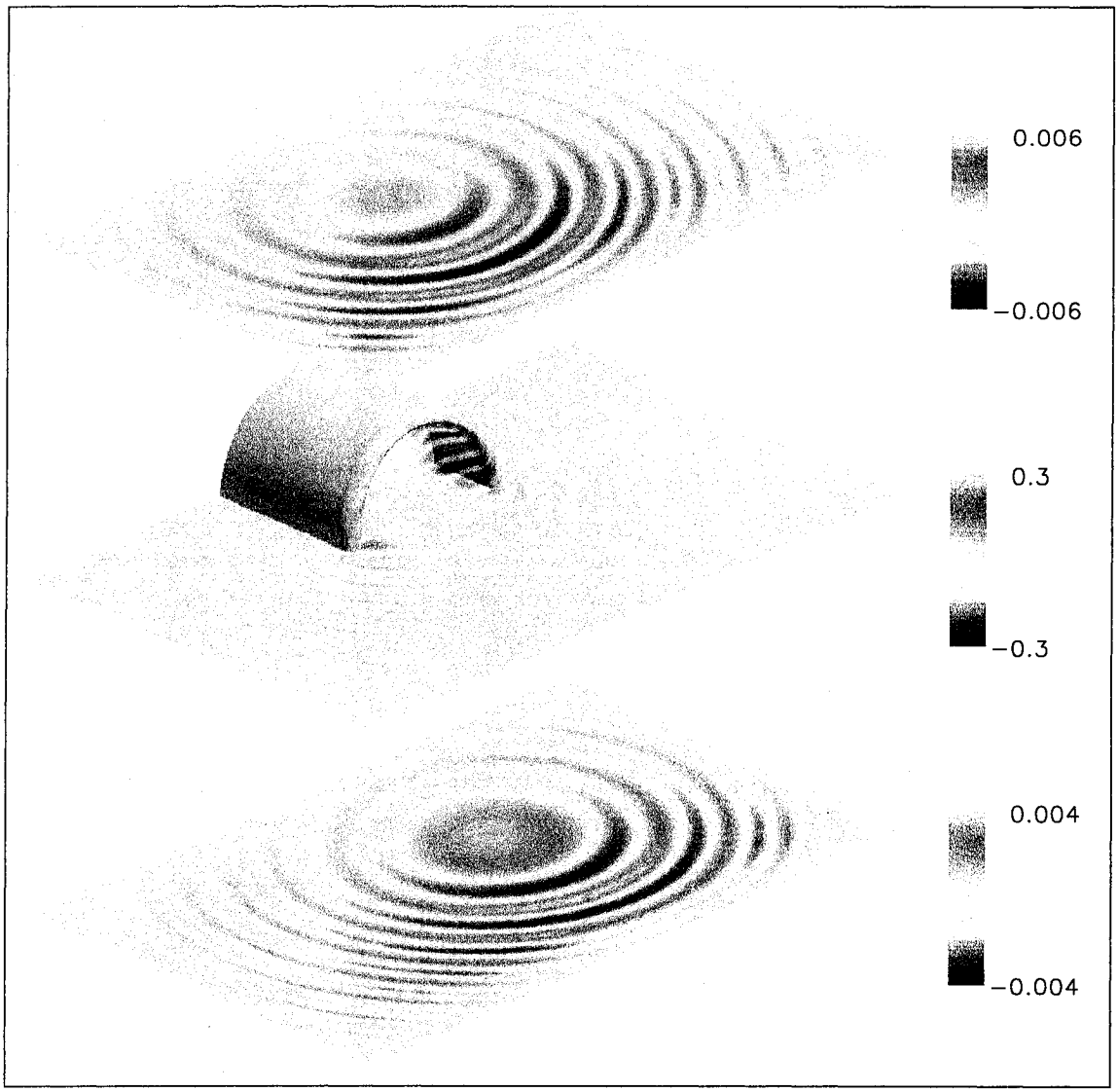

Figure 5.54: Contour maps of the acoustic pressure at both ends of the computational domain and along the $y = 0$ plane.

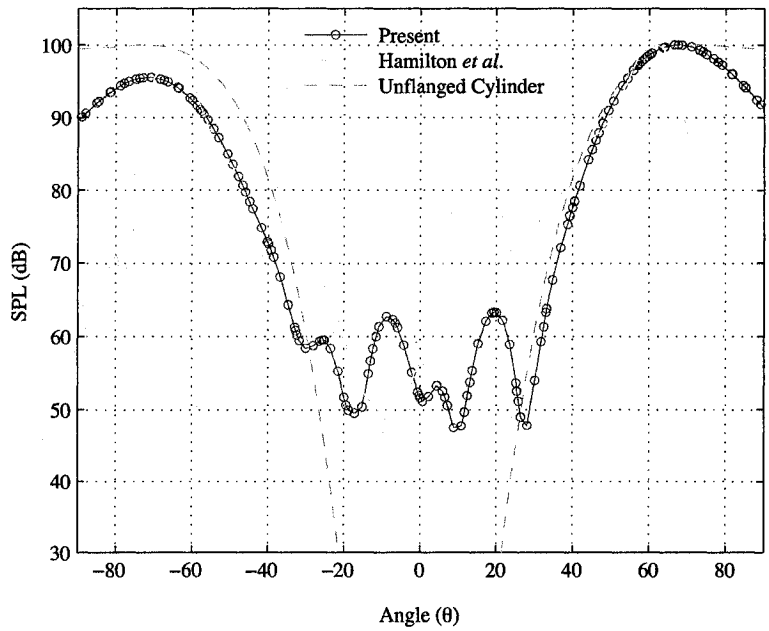

Figure 5.55: SPL directivity of the (13,0) mode on the xy-plane; $\bar{\omega} = 17$.

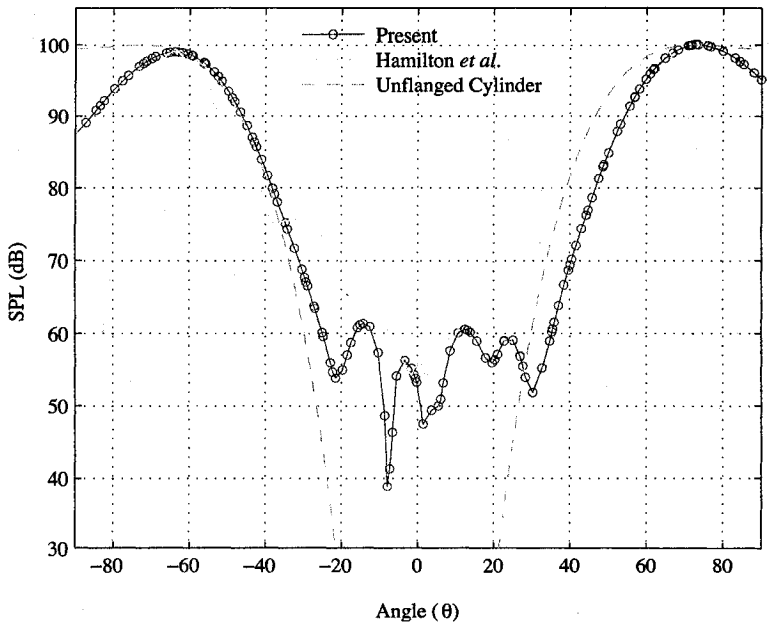

Figure 5.56: SPL directivity of the (13,0) mode on the xz-plane; $\bar{\omega} = 17$.

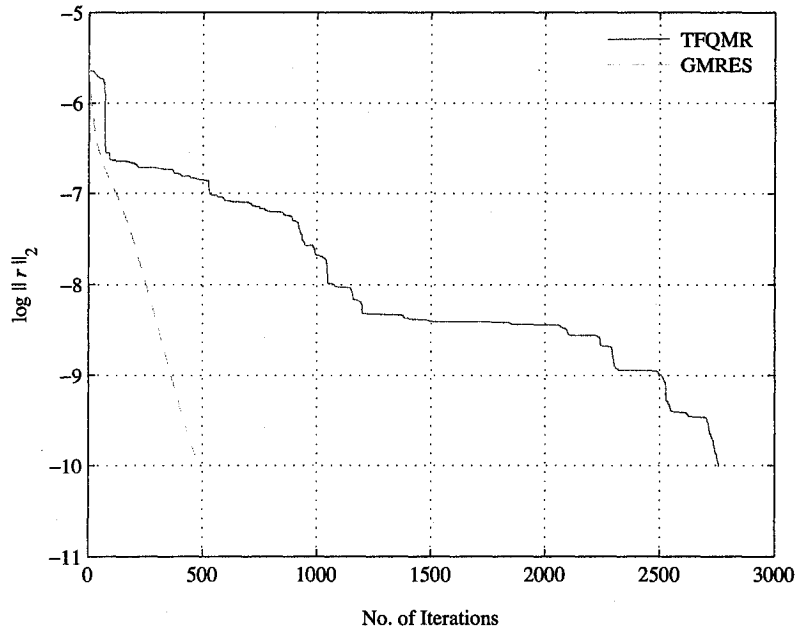

Figure 5.57: Convergence history of the iterative solvers for the Schur matrix.

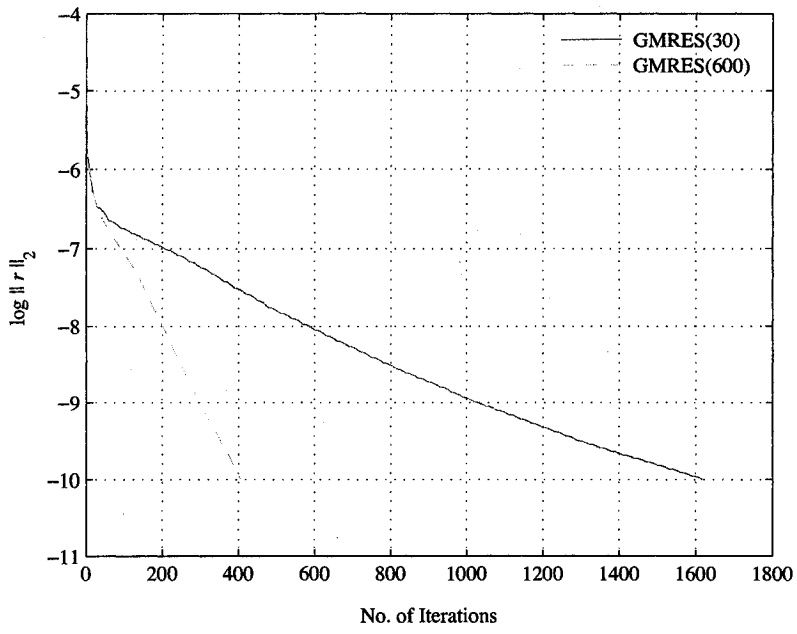

Figure 5.58: Effect of the restart parameter on convergence rate of the GMRES algorithm.

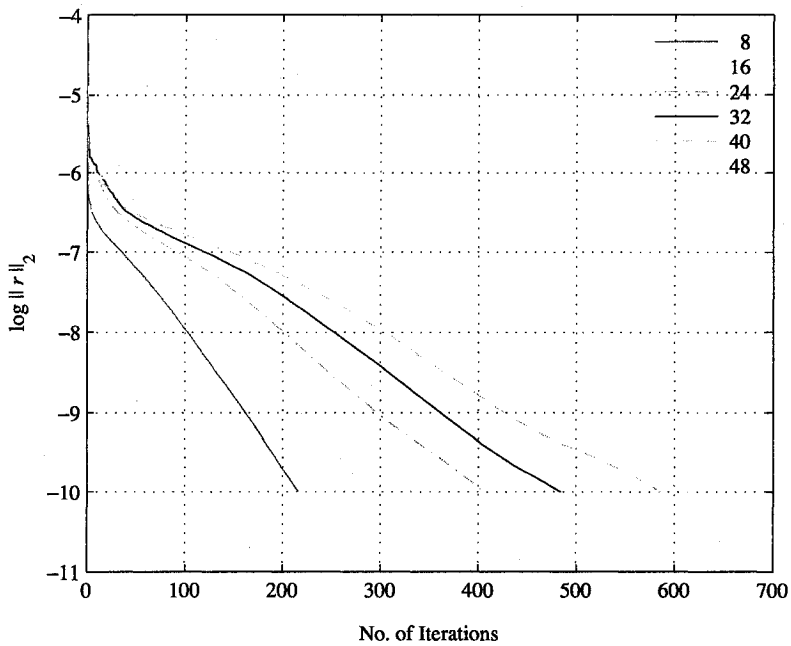

Figure 5.59: Effect of the number of partitions on the convergence rate of the Schur matrix, using the GMRES algorithm.

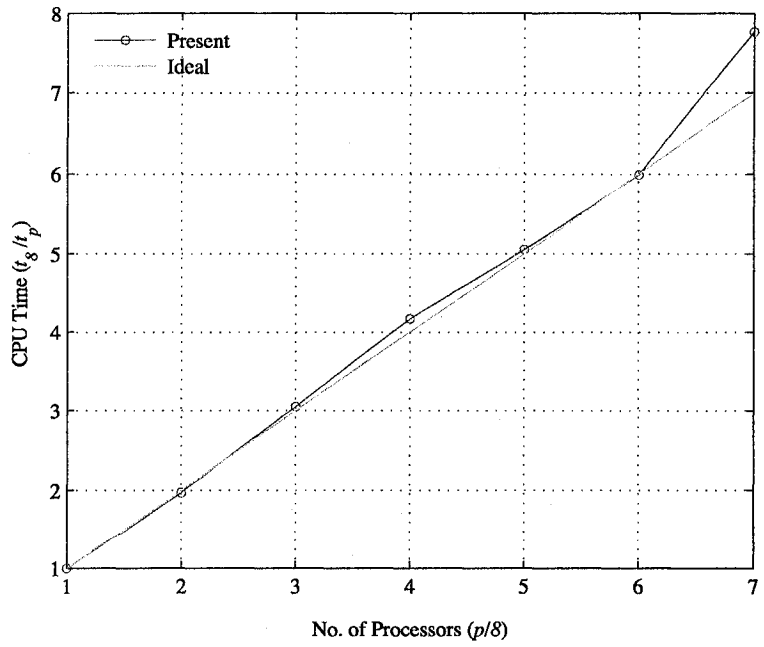

Figure 5.60: Speedup vs. number of CPU's; data normalized with respect to $p = 8$.

Chapter 6

Conclusions and Future Research

6.1 Summary and Conclusions

A 3D code was developed using the spectral element method for the simulation of noise generated by turbofan engine inlets. The applicability of the code, however, is not restricted to fan noise problems. The 3D formulation allows prediction of far-field acoustic radiation from a duct of an arbitrary shape (asymmetric) and geometry with a rectangular or circular source.

The governing equations are based on the linearized Euler equations, which are further simplified to a set of equations in terms of a single variable, i.e. the acoustic potential, using the irrotational flow assumption. Spatial discretization is based on a spectral element method, where the computational domain is subdivided into hexahedral elements within which an N^{th}-order Chebyshev spectral approximation is applied at the nodes of a Gauss-Chebyshev-Lobatto grid. Due to the distinct features of a wave propagation problem, use of a high-order method is essential to minimize the errors associated with dissipation or dispersion as the waves travel within the domain.

Further, given the periodic nature of the fan noise problem, the governing equations were written in the frequency domain. This eliminated the need for a time-marching (explicit) solution method. Instead, a large system of linear equations had to be solved, which

is known to be challenging due to ill-conditioning of the coefficient matrix. For this purpose, a solution method based on the domain decomposition using a Schur complement method was used. The Schur matrix was solved iteratively using a matrix-free approach using a novel local preconditioner. The method was demonstrated to be robust and very efficient for a large number of problems.

Moreover, a mean flow solver based on the full potential equation was added in order to take into account the effects of flow variations around the nacelle on the scattering of the radiated sound field.

A large number of test cases were presented, ranging in size from 100 000-2 000 000 unknowns. In all tests, the results were compared with either analytical or other numerical data (or both), which validated and demonstrated the suitability of the proposed method for the solution of a wide range of acoustic problems.

All aspects of numerical simulations, including building and assembling the coefficient matrices, implementation of the Schur complement method, and solution of the system of equations were performed in parallel. Depending on the size of the problem, between 8-48 CPU's were used. The developed code was shown to have an excellent parallel performance with a linear or superlinear speed-up. For the purpose of parallelization, the MPI standard and PETSc libraries were extensively used.

The method presented in this thesis and the associated computer code were developed to enable a tool that can be applied and used for realistic design problems. Therefore, a very important aspect of this technology, in addition to accuracy, is the amount of time and computer resources required to perform such simulations. It was shown that the current method could solve a real world problem (scarfed inlet) in about 5 hrs on 16 CPU's. This is still considered a very expensive computation by many industrial standards. So, reducing the amount of memory usage and speeding up the calculations are the areas where we will focus and make several recommendations for future work in the context of the current research.

6.2 Future Research

Recall that in applying the Schur complement method, a large amount of memory was used for storing the factorized matrices $A_{II}^{(i)}$ and $A_{BB}^{(i)}$ because a direct method was used for solving systems involving the two at every iteration. An alternative is to use an iterative method that will substantially require much less memory, especially for the $A_{II}^{(i)}$ matrix that is dense. A good preconditioner is the key for a fast convergence, given the ill-conditioning of these matrices, otherwise the large solution time will outweigh the benefits of using less memory. Our experiments indicated that standard preconditioners, such as ILU were not efficient.

One possibility that could be investigated is to build a preconditioner using the fast diagonal method (FDM) [12]. This method uses 1D matrix operators to build the three dimensional differential operator using a matrix tensor product, assuming that the geometry is a cuboid (hexahedral with right angles). For such geometries, it is then possible to build an inverse operator using eigenvalues and eigenvectors of the 1D operators, which are of smaller size [65].

For hexahedral elements with general shapes, one could build an approximate inverse operator in a similar manner by first creating a cuboid to represent the deformed elements using proper average dimensions in each coordinate direction. Such an approximate inverse operator could be used as a preconditioner for the $A_{II}^{(i)}$ matrices. The 1D operators could either be based on the N^{th} degree polynomials of the spectral grid or simpler linear or quadratic approximations. Such an approach has been used recently in the context of domain decomposition method for the solution of the Poisson equation [63]. In addition, one can simplify the process by building the inverse operator only for the elliptic terms in the weak form, Eq. (2.28). Ref. [13] provides the formulas for differential operators for this case.

One area that could be considered is mesh adaptation, which to our best knowledge, has not been examined for the fan noise problem before. Given the directionality of acoustic radiation from the engine inlet, such a measure is justified and could reduce the size of the

problem. The issues to be considered include the type of the *a posteriori* error estimator to be used, which is somewhat different for acoustic problems [11, 94], and reconstruction of the spectral grid and associated data structure for the newly added or deformed elements. Also, to estimate the error, it would be easier to use the information on the spectral grid but apply the adaptation to the background finite element grid using some kind of lumping or averaging strategy. The literature on the subject of the mesh adaptation for acoustic problems is not vast, but it seems that the majority of approaches favor an h-refinement.

In addition to being ill-conditioned, the coefficient matrix of the discretized acoustic problem is also nonsymmetric and indefinite. If the governing equations are recast in the form of a system of first order differential equations, then one could use a least-squares method to build the coefficient matrix [97]. In this case, the coefficient matrix would be symmetric and positive definite (SPD), for which efficient preconditioners are available, not to mention the savings in computer memory due to storing only half the matrix. Such an approach was applied for the solution of unsteady sound propagation problems in one and two dimensions [61]. The current acoustic problem could be solved with this method either in the frequency domain, or by introducing a pseudo-time term, in the time domain. Obviously, there will be a large savings in memory if the latter approach is adopted. However, in a very recent work by Özyörük *et al.*, the solution of a fan noise problem using a pseudo-time on a grid of size 241×81 (i.e., 19 521 unknowns) required 60 000 time steps and 4.3 hrs of CPU time on 12 processors to converge. This is a large computation time, given that using the Schur complement method, we solved a problem of slightly larger size on 8 processors in only 20 minutes.

The current approach based on solving the acoustic equations in the entire domain has limitations for problems where the observer is located at a very large distance from the source, simply because a high-order solution for such a large domain will be extremely expensive and not practical. An alternative is to use a hybrid method, where the CAA is used only in the near-field region, and the far-field domain is resolved using a boundary integral method. This approach relies on the acoustic analogy and could lead to savings in

computation time of more than an order of magnitude [107]. The integral solution in the far-field is usually based on the Kirchhoff formulation. The assumption involved, however, is that the finite volume surface surrounding the near-field must be located at a region where the linear wave equation is valid. This restriction, however, could be lifted if one uses a permeable surface form of the Ffowcs Williams-Hawkings (FW-H) equation [8], which is completely based on the continuity and momentum equations and is valid in regions of the domain where nonlinearities exist. This latter form, however, is in terms of the primitive flow variables, i.e., ρ, u, v, w and p. So, if a potential formulation is used in the near-field, the integral equations should be modified to be in terms of the acoustic potential.

Bibliography

[1] C. Ashcraft and R. Grimes, "SPOOLES: An Object-Oriented Sparse Matrix Library," *Proceedings of the SIAM Conference on Parallel Processing for Scientific Computing*, March 22-27, 1999.

[2] R.J. Astley, "Wave Envelope and Infinite Elements for Acoustical Radiation," *International Journal for Numerical Methods in Fluids*, Vol. 3, pp. 507-526, 1983.

[3] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, Cambridge, 1994.

[4] S. Balay, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, and H. Zhang, "PETSc Users Manual," *ANL-95/11 - Revision 2.1.5*, Argonne National Laboratory, 2004.

[5] P. Beckmann, *Orthogonal Polynomials for Engineers and Physicists*, Golem Press, 1973.

[6] J-P. Berenger, "A Perfectly Matched Layer for the Absorption of Electromagnetic Waves," *Journal of Computational Physics*, Vol. 114, pp. 185-200, 1994.

[7] R.T. Biedron, C.L. Rumsey, G.G. Podboy, and M.N. Dunn, "Predicting the Rotor-Stator Interaction Acoustics of a Ducted Fan Engine," *AIAA Paper 2001-0664*, 2001.

[8] K.S. Brentner and F. Farassat, "Analytical Comparison of the Acoustic Analogy and Kirchhoff Formulation for Moving Surfaces," *AIAA Journal*, Vol. 36, pp. 1379-1386, 1998.

[9] F. Collino and P. Monk, "The Perfectly Matched Layer in Curvilinear Coordinates," *SIAM Journal of Scientific Computing*, Vol. 19, pp. 2016-2090, 1998.

[10] T.F. Chan and D. Goovaerts, "Schur Complement Domain Decomposition Algorithms for Spectral Methods," *Applied Numerical Mathematics*, Vol. 6, pp. 53-64, 1989.

[11] J.T. Chen, K.H. Chen and C.T. Chen, "Adaptive Boundary Element Method of Time-harmonic Exterior Acoustics in Two Dimensions," *Computer Methods in Applied Mechanics and Engineering*, Vol. 191, pp. 3331-3345, 2002.

[12] W. Couzy and M.O. Deville, "A Fast Schur Complement Method for the Spectral Element Discretization of the Incompressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 116, pp. 135-142, 1995.

[13] M.O. Deville, P.F. Fischer, and E.H. Mund, *High-Order Methods for Incompressible Fluid Flow*, Cambridge University Press, 2002.

[14] J-F. Dietiker, K.A. Hoffmann, M. Papadakis, and R.K. Agarwal, "Development of Three-dimensional PML Boundary Conditions for Aeroacoustics Applications," *International Journal of Aeroacoustics*, Vol. 1, pp. 307-327, 2002.

[15] T.Z. dong, S.H. Shih, R.R. Mankbadi, and L.A. Povinelli, "A Numerical Study of Duct Geometry Effect," *AIAA Paper 1997-1604*, 1997.

[16] M. Dryja and O.B. Widlund, "An Additive Variant of the Schwarz Alternating Method for the Case of Many Subregions," *Technical Report 339*, Courant Institute of Mathematical Sciences, New York, 1987.

[17] B. Enquist and A. Majda, "Radiation Boundary Conditions for Acoustic and Elastic Wave Calculations," *Communications in Pure and Applied Mathematics*, Vol. 32, pp. 313-357, 1979.

[18] G. Evans, *Practical Numerical Integration*, John Wiley & Sons, 1993.

[19] W. Eversman and I.D. Roy, "Ducted Fan Acoustic Radiation Including the Effects of Nonuniform Mean Flow and Acoustic Treatment," *AIAA Paper 93-4424.*

[20] W. Eversman, "Theoretical Models for Duct Acoustic Propagation and Radiation," in *Aeroacoustics of Flight Vehicles, Theory and Practice, Vol. 2: Noise Control*, Acoustical Society of America, 1995.

[21] E. Envia, "Fan Noise Prediction: An Overview," *AIAA Paper 2001-0661*, 2001.

[22] E. Envia, A.G. Wilson, and D.L. Huff, "Fan Noise: A Challenge to CAA," *International Journal of Computational Fluid Dynamics*, Vol. 18, pp. 471-480, 2004.

[23] F. Farassat, "Linear Acoustic Formulas for Calculation of Rotating Blade Noise," *AIAA Journal*, Vol. 19, pp. 1122-1130, 1980.

[24] F. Farassat and M.K. Myers, "Extension of Kirchhoff's Formula to Radiation from Moving Surfaces," *Journal of Sound and Vibration*, Vol. 123, pp. 451-460, 1988.

[25] F. Farassat and M.K. Myers, "A Graphical Approach to Wave Propagation in a Rigid Duct," *Journal of Sound and Vibration*, Vol. 200, pp. 729-735, 1997.

[26] J.E. Ffowcs Williams and D.L. Hawkings, "Sound Generation by Turbulence and Surfaces in Arbitrary Motion," *Philosophical Transactions of the Royal Society*, Series A, Vol. 264, pp. 321-342, 1969.

[27] P.F. Fischer, "An Overlapping Schwarz Method for Spectral Element Solution of the Incompressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 133, pp. 135-101, 1997.

[28] J.B. Freund, "Proposed Inflow/Outflow Boundary Condition for Direct Computation of Aerodynamic Sound," *AIAA Journal*, Vol. 35, pp.740-742, 1997.

[29] D. Funaro, *Polynomial Approximation of Differential Equations*, Springer-Verlag, 1992.

[30] J.M. Gallman, M.K. Myers, and F. Farassat, "Boundary Integral Approach to the Scattering of Nonplanar Acoustic Waves by Rigid Bodies," *AIAA Journal.* Vol. 29, pp. 2038-2046, 1991.

[31] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. S. Sunderam, *PVM: Parallel Virtual Machine - A Users' Guide and Tutorial for Network Parallel Computing*, MIT Press, 1994.

[32] M.B. Giles, "Non-reflecting Boundary Conditions for Euler Equation Calculations," *AIAA Journal*, Vol. 28, pp. 2050-2058, 1990.

[33] D. Givoli, "Non-reflecting Boundary Conditions," *Journal of Computational Physics*, Vol. 94, pp. 1-29, 1991.

[34] W.J. Gordon and C.A. Hall, "Construction of Curvilinear Co-ordinate Systems and Applications to Mesh Generation," *International Journal for Numerical Methods in Engineering*, Vol. 7, pp. 461-477, 1973.

[35] W.J. Gordon and C.A. Hall, "Transfinite Element Methods: Blending-Function Interpolation Over Arbitrary Curved Element Domains," *Numerical Mathematics*, Vol. 21, pp. 109-129, 1973.

[36] D. Gottlieb and S.A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Application*, SIAM Publications, 1977.

[37] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI - Portable Parallel Programming with the Message Passing Interface, 2/e*, MIT Press, 1999.

[38] J.A. Hamilton and R.J. Astley, "Acoustic Propagation on Irrotational Mean Flows Using Transient Finite and Infinite Elements," *AIAA Journal*, Vol. 43, pp. 124-134, 2005.

[39] D.L. Hawkings, "Multiple Tone Generation by Transonic Compressors," *Journal of Sound and Vibration*, Vol. 17, pp. 241-250, 1971.

[40] Sh. Hirsch, G. Ghorbaniasl, and J. Ramboer, "Fan Noise Simulation in the Time Domain: Validation Test Cases," *Proceedings of Fan Noise 2003 International Symposium*, Senlis, France, September 2003.

[41] G.F. Homicz and J.A. Lordi, "A Note on Radiative Directivity Patterns of Duct Acoustic Modes," *Journal of Sound and Vibration*, Vol. 41, pp. 283-290, 1975.

[42] S.J. Horowitz, R.K. Sigman, and B.T. Zinn, "An Iterative Finite Element-Integral Technique for Predicting Sound Radiation from Turbofan Inlets," *AIAA Paper 81-1987*, 1981.

[43] F.Q. Hu, "Absorbing Boundary Conditions," *International Journal of Computational Fluid Dynamics*, Vol. 18, pp. 513-522, 2004.

[44] D.L. Huff, "Fan Noise Prediction: Status and Needs," *AIAA Paper 98-0177*, 1998.

[45] M. Israeli and S.A. Orszag, "Approximation of Radiation Boundary Conditions," *Journal of Computational Physics*, Vol. 41, pp. 115-135, 1981.

[46] G. Karniadakis and S.J. Sherwin, *Spectral/hp Element Methods For Computational Fluid Dynamics*, Oxford University Press, 2005.

[47] G. Karypis, V. Kumar, and K. Schloegel, "Graph Partitioning for High-performance Scientific Simulations," *Computing Reviews*, Vol. 45, pp. 110, 2004.

[48] D.E. Keyes and W.D. Gropp, "Domain Decomposition Techniques for the Parallel Solution of Nonsymmetric Systems of Elliptic Boundary Value Problems," *Applied Numerical Mathematics*, Vol. 6, pp. 281-301, 1990.

[49] P. Knupp and S. Steinberg, *Fundamentals of Grid Generation*, CRC Press, 1993.

[50] D.A. Kopriva, "A Spectral Multidomain Method for the Solution of Hyperbolic Systems," *Applied Numerical Mathematics*, Vol. 2, pp. 221-241, 1986.

[51] D.A. Kopriva and J.H. Kolias, "A Conservative Staggered-Grid Chebyshev Multidomain Method for Compressible Flows," *Journal of Computational Physics*, Vol. 125, pp. 244-261, 1996.

[52] K.Z. Korczak and A.T. Patera, "An Isoparametric Spectral Element Method for Solution of the Navier-Stokes Equations in Complex Geometry," *Journal of Computational Physics*, Vol. 62, pp. 361-382, 1986.

[53] R. Kosloff and D. Kosloff, "Absorbing Boundary Conditions for Wave Propagation Problems," *Journal of Computational Physics*, Vol. 63, pp. 363-376, 1986.

[54] J.H. Lan, Y. Guo, and C. Breard, "Validation and Application of Two Frequency Domain Acoustic Propagation Codes," *AIAA Paper 2003-3244*, 2003.

[55] S.K. Lele, "Compact Finite Difference Schemes with Spectral-like Resolution," *Journal of Computational Physics*, Vol. 103, 16-42, 1992.

[56] S.J. Lele, "Computational Aeroacoustics: A Review," *AIAA Paper 97-0018*, 1997.

[57] P. Le Tallec, "Neumann-Neumann Domain Decomposition Algorithms for solving 2D Elliptic Problems with Non-matching Grids," *East-West Journal of Mathematics*, Vol. 1, pp. 129-146, 1993.

[58] Z. Li, Y. Saad, and M. Sosonkina, "pARMS : A Parallel Version of the Algebraic Recursive Multilevel Solver," *Report umsi-2001-100*, Minnesota Supercomputer Institute, University of Minnesota, 2001.

[59] S. Lidoine, H. Batard, S. Troyes, A. Delnevo, and M. Roger, "Acoustic Radiation Modeling of Aeroengine Intake - Comparison between Analytical and Numerical Methods," *AIAA Paper 2001-2140*, 2001.

[60] M.J. Lighthill, "On Sound Generated Aerodynamically. I. General Theory," *Proceedings of the Royal Society of London*, Series A, Vol. 211, pp. 564-587, 1952.

[61] W.H. Lin, "Least-Squares Legendre Spectral Element Solutions to Sound Propagation Problems," *Journal of Acoustical Society of America*, Vol. 109, pp. 465-474, 2001.

[62] E. Listerud and W. Eversman, "Finite Element Modeling of Acoustics Using Higher Order Elements. Part II: Turbofan Acoustic Radiation," *Journal of Computational Acoustics*, Vol. 3, pp. 431-446, 2004

[63] J.W. Lottes and P.F. Fischer, "Hybrid Multigrid/Schwarz Algorithms for the Spectral Element Method," *SIAM Journal of Scientific Computing*, Vol. 24, pp. 613-646, 2005.

[64] J.K.C. Low, "Ultra-High Bypass Ratio Jet Noise," *NASA CR-195394*, 1994.

[65] R.E. Lynch, J.R. Rice, and D.H. Thomas, "Direct Solution of Partial Differential Equations by Tensor Product Methods," *Numerical Mathematics*, Vol. 6, 185-199, 1964.

[66] A.S. Lyrintzis, "Review: The Use of Kirchhoff's Method in Computational Aeroacoustics," *ASME Journal of Fluids Engineering*, Vol. 116, pp. 665-675, 1994.

[67] P. Malbéqui and C. Glandier, "Sound Propagation and Radiation in a Curved Duct," *AIAA Journal*, Vol. 34, pp. 1778-1784, 1996.

[68] J. Mandel, "Balancing Domain Decomposition," *Communications in Numerical Methods in Engineering*, Vol. 9, pp 233-241, 1993.

[69] R. Mani, "Refraction of Acoustic Duct Waveguide Modes by Exhaust Jets," *Quarterly of Applied Mathematics*, Vol. 30, pp 501-517, 1973.

[70] A. McAlpine and M.J. Fisher, "On the Prediction of Buzz-saw Noise in Aero-engine Inlet Ducts," *Journal of Sound and Vibration*, Vol. 248, pp. 123-149, 2001.

[71] P. Moin, *Fundamentals of Engineering Numerical Analysis*, Cambridge University Press, 2001.

[72] G. Moore, "Lithography and the Future of the Moore's Law," *Optical/Laser Microlithography VIII: Proceedings of the SPIE*, Vol. 2240, pp 2-17, 1995.

[73] R.E. Owens, "Energy Efficient Engine Performance System - Aircraft Integration Evaluation," *NASA CR-159488*, 1979.

[74] Y. Özyörük and L.N. Long, "A Navier-Stokes/Kirchhoff Method for Noise Radiation from Ducted Fans," *AIAA Paper 94-0462*, 1994.

[75] Y. Özyörük and L.N. Long, "Time-Domain Calculation of Sound Propagation in Lined Ducts with Sheared Flows," *AIAA Journal*, Vol. 38, pp. 768-773, 2000.

[76] Y. Özyörük, "Parallel Computation of Forward Radiated Noise of Ducted Fans including Acoustic Treatment," *AIAA Journal*, Vol. 40, pp. 450-455, 2002.

[77] A.T. Patera, "A Spectral Element Method for Fluid Dynamics: Laminar Flow in a Channel Expansion," *Journal of Computational Physics*, Vol. 54, pp. 468-488, 1984.

[78] P. Petropoulos, "Reflectionless Sponge Layers as Absorbing Boundary Conditions for the Numerical Solution of Maxwell's Equations in Rectangular, Cylindrical, and Spherical Coordinates," *SIAM Journal of Applied Mathematics*, Vol. 60, pp. 1037-1065, 2000.

[79] A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford University Press, 1999.

[80] J.W.S. Rayleigh, *Theory of Sound*, 2 Volumes, Macmillan, 1877, Reprinted by Dover, 1945.

[81] R.A. Renault, "Absorbing Boundary Conditions, Difference Operators, and Stability," *Journal of Computational Physics*, Vol. 102, pp. 236-251, 1992.

[82] E.J. Rice, M.F. Heidmann and T.G. Sofrin, "Modal Propagation Angles in a Cylindrical Duct with Flow and Their Relation to Sound Radiation," *AIAA Paper 79-0183*, 1979.

[83] E.M. Ronquist and A.T. Patera, "A Legendre Spectral Element Method for the Stefan Problem," *International Journal for Numerical Methods in Engineering*, Vol. 24, pp. 2273-2299, 1987.

[84] I.D. Roy, W. Eversman, and H.D. Meyer, "Improved Finite Element Modeling of the Turbofan Engine Inlet Radiation Problem," *NASA-CR-204341*, 1993.

[85] C.L. Rumsey, R.T. Biedron, F. Farassat, and P.L. Spence, "Ducted-fan Engine Acoustic Predictions Using a Navier-Stokes Code," *Journal of Sound and Vibration*, Vol. 213, pp. 643-664, 1998.

[86] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM Publications, 2003.

[87] H.M. Blackburn and S.J. Sherwin, "Formulation of a Galerkin Spectral Element-Fourier Method for Three-dimensional Incompressible Flows in Cylindrical Geometries," *Journal of Computational Physics*, Vol. 197, pp. 759-778, 2004.

[88] R.J. Silcox, "Experimental Investigation of Geometry and Flow Effects on Acoustic Radiation from Duct Inlets," *AIAA Paper 83-0713*, 1983.

[89] B. Smith, P. Bjorstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.

[90] A. Solomonoff, "A Fast Algorithm for Spectral Differentiation," *Journal of Computational Physics*, Vol. 98, pp. 174-177, 1992.

[91] D. Stanescu, D. Ait-Ali-Yahia, W.G. Habashi, and M.P. Robichaud, "Multidomain Spectral Computations of Sound Radiation from Ducted Fans," *AIAA Journal*, Vol. 37, pp. 296-302, 1999.

[92] D. Stanescu, M.Y. Hussaini, and F. Farassat, "Large Scale Frequency Domain Numerical Simulation of Aircraft Engine Tone Noise Radiation and Scattering," *AIAA Paper 2002-2586*, 2002. 1997.

[93] D. Stanescu, D. Ait-Ali-Yahia, W.G. Habashi, and M.P. Robichaud, "Spectral Element Formulation of the Linearized Fan Tone Noise Radiation Problem," *AIAA Journal*, Vol. 42, pp. 696-705, 2004.

[94] J.R. Stewart and T.J.R. Hughes, "h-Adaptive Finite Element Computation of Time-harmonic Exterior Acoustics Problems in Two Dimensions," *Computer Methods in Applied Mechanics and Engineering*, Vol. 146, pp. 65-89, 1997.

[95] D.L. Sutliff, Z. Hu, F.G. Pla, and L.J. Heidelberg, "Active Noise Control of Low Speed Fan Rotor-Stator Modes," *AIAA Paper 97-1641*, 1997.

[96] S. Ta'asan and D.M. Nark, "An Absorbing Buffer Zone Technique for Acoustic Wave Propagation," *AIAA Paper 95-0164*, 1995.

[97] F. Taghaddosi, W.G. Habashi, G. Guèvremont, and D. Ait-Ali-Yahia, "An Adaptive Least-Squares Method for the Compressible Euler Equations," *International Journal for Numerical Methods in Fluids*, Vol. 31, pp. 1121-1139, 1999.

[98] F. Taghaddosi and W.G. Habashi, "3D Computation of Noise Propagation from Ducted Fans Using a Spectral Element Method," *AIAA Paper 2004-0520*, 2004.

[99] C.K.W. Tam and J.C. Webb, "Dispersion-Relation-Preserving Finite Difference Schemes for Computational Acoustics," *Journal of Computational Physics*, Vol. 107, pp. 262-281, 1993.

[100] C.K.W. Tam, "Advances in Numerical Boundary Conditions for Computational Aeroacoustics," *Journal of Computational Acoustics*, Vol. 6, 377-402, 1998.

[101] C.K.W. Tam, "Computational Aeroacoustics: An Overview of Computational Challenges and Applications," *International Journal of Computational Fluid Dynamics*, Vol. 18, pp. 547-567, 2004.

[102] S. Temkin, *Elements of Acoustics*, John Wiley & Sons, 1981.

[103] A. Toselli and O. Widlund, *Domain Decomposition Methods - Algorithms and Theory*, Springer Series in Computational Mathematics, Vol. 34, 2005.

[104] N. Tsuchiya, Y. Nakamura, A. Yamagata, and H. Kodama, "Fan Noise Prediction Using Unsteady CFD Analysis," *AIAA Paper 2002-2491*, 2002.

[105] J.M. Tyler and T.G. Sofrin, "Axial Flow Compressor Noise Studies," *SAE Transactions*, Vol. 70, pp. 309-332, 1962.

[106] H.A. Van Der Vorst, "Bi-CGSTAB: A Fast and Smoothly Convergent Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems," *SIAM Journal of Scientific and Statistical Computing*, Vol. 13, pp. 631-644, 1992.

[107] D.A. Venditti, D. Ait-Ali-Yahia, M.P. Robichaud, and G. Girard, "Spectral Element/Kirchhoff Method for Fan-Tone Directivity Calculations," *AIAA Paper 2005-2926*, 2005.

[108] J. Xu, D. Stanescu, M.Y. Hussaini, and F. Farassat, "Computation of Engine Noise Propagation and Scattering Off an Aircraft," *AIAA Paper 2003-0542*, 2003.

[109] B. Yang, D. Gottlieb, and J.S. Hesthaven, "Spectral Simulations of Electromagnetic Wave Scattering," *Journal of Computational Physics*, Vol. 134, pp. 216-230, 1997.

[110] X. Zhang, X.X. Chen, C.L. Morfey, and P.A. Nelson, "Computation of Spinning Modal Radiation from an Unflanged Duct," *AIAA Journal*, Vol. 42, pp. 1795-1801, 2004.

Appendix A

Derivation of Momentum Equation in Terms of Density

We start the derivation with the momentum equation written in the vector form, equation (2.8),

$$\frac{\partial \vec{V}}{\partial t} + \vec{V} \cdot \nabla \vec{V} = -\frac{\nabla p}{\rho}, \tag{A.1}$$

where we have dropped the superscript "$*$" for the sake of simplicity. Since the flow is assumed irrotational, i.e., $\nabla \times \vec{V} = 0$, one can introduce a potential, Φ, such that, $\vec{V} = \nabla \Phi$. So, using the vector identity

$$\vec{A} \cdot \nabla \vec{A} = \nabla\Big(\frac{\vec{A}^{\,2}}{2}\Big) + (\nabla \times \vec{A}) \times \vec{A}, \tag{A.2}$$

we can write

$$\vec{V} \cdot \nabla \vec{V} = \nabla\Big(\frac{\vec{V}^{\,2}}{2}\Big) = \nabla\Big(\frac{\vec{V} \cdot \vec{V}}{2}\Big), \tag{A.3}$$

and the left-hand side of the momentum equation becomes

$$\frac{\partial \vec{V}}{\partial t} + \vec{V} \cdot \nabla \vec{V} = \frac{\partial \nabla \Phi}{\partial t} + \nabla\Big(\frac{\nabla \Phi \cdot \nabla \Phi}{2}\Big) = \nabla \left[\frac{\partial \Phi}{\partial t} + \frac{1}{2}(\nabla \Phi \cdot \nabla \Phi)\right]. \tag{A.4}$$

To write the right-hand side of the momentum equation in terms of acoustic potential, Φ, we first use the equation of state, (Eq. 2.6), and find its gradient

$$p = \frac{1}{\gamma}\rho^{\gamma}, \qquad \rightarrow \qquad \nabla p = \frac{1}{\gamma}(\gamma\, \rho^{\gamma-1}\, \nabla \rho) = \rho^{\gamma-1}\, \nabla \rho. \tag{A.5}$$

In a similar way, the equation for the speed of sound, Eq. (2.7), becomes

$$c^2 = \rho^{\gamma-1}, \qquad \rightarrow \qquad \nabla(c^2) = (\gamma-1)\,\rho^{\gamma-2}\,\nabla\rho = (\gamma-1)\,\frac{\rho^{\gamma-1}\,\nabla\rho}{\rho}. \tag{A.6}$$

Combining these two equations, we get

$$\nabla(c^2) = (\gamma-1)\frac{\nabla p}{\rho}, \tag{A.7}$$

or,

$$-\frac{\nabla p}{\rho} = -\frac{\nabla(c^2)}{\gamma-1}. \tag{A.8}$$

So, the momentum equation can be written as

$$\nabla\left[\frac{\partial\Phi}{\partial t} + \frac{1}{2}(\nabla\Phi\cdot\nabla\Phi)\right] = -\frac{\nabla(c^2)}{\gamma-1}, \tag{A.9}$$

or along a streamline

$$\left[\frac{\partial\Phi}{\partial t} + \frac{1}{2}(\nabla\Phi\cdot\nabla\Phi)\right] + \frac{c^2}{\gamma-1} = \text{const.} \tag{A.10}$$

For a point at the far-field, we have

$$(\partial/\partial t)_\infty = 0, \qquad (\nabla\Phi\cdot\nabla\Phi)_\infty = |\nabla\Phi|^2_\infty = M^2_\infty, \qquad \text{and} \qquad c_\infty = 1, \tag{A.11}$$

leading to

$$\frac{M^2_\infty}{2} + \frac{1}{\gamma-1} = \text{const.} \tag{A.12}$$

So, writing equation (A.10) along a streamline between the two points, one in the near field and one in the far-field, gives

$$\left[\frac{\partial\Phi}{\partial t} + \frac{1}{2}(\nabla\Phi\cdot\nabla\Phi)\right] + \frac{c^2}{\gamma-1} = \frac{M^2_\infty}{2} + \frac{1}{\gamma-1}, \tag{A.13}$$

or

$$\frac{1-c^2}{\gamma-1} = \left[\frac{\partial\Phi}{\partial t} + \frac{1}{2}(\nabla\Phi\cdot\nabla\Phi) - \frac{M^2_\infty}{2}\right], \tag{A.14}$$

which can also be written in terms of density using the equation for the speed of sound

$$\rho = \left[1-(\gamma-1)\left(\frac{\partial\Phi}{\partial t} + \frac{\nabla\Phi\cdot\nabla\Phi - M^2_\infty}{2}\right)\right]^{1/(\gamma-1)}. \tag{A.15}$$

Appendix B

Integrals Involving Chebyshev Functions

Here, we derive exact formulas for several integrals involving Chebyshev polynomials and/or their derivatives. These integrals appear frequently in the derivation of coefficient matrices, as discussed in the following appendices. We start with the definition of the n^{th} order Chebyshev polynomial, $T_n(\xi)$,

$$T_n(\xi) = \cos(n \cos^{-1} \xi) = \cos n\theta, \qquad \text{where} \quad \xi = \cos\theta. \tag{B.1}$$

Derivative of the Chebyshev polynomial can be easily obtained by applying the chain rule

$$\frac{dT_n(\xi)}{d\xi} = \frac{dT_n}{d\theta}\frac{d\theta}{d\xi} = \frac{dT_n/d\theta}{d\xi/d\theta}, \tag{B.2}$$

or

$$\frac{dT_n(\xi)}{d\xi} = \frac{n \sin n\theta}{\sin\theta}. \tag{B.3}$$

In the course of derivation, we will repeatedly make use of the following results

$$\int_0^\pi \sin p\theta \sin q\theta \, d\theta = \begin{cases} 0 & p+q \quad \text{even}, \\ \dfrac{2p}{p^2 - q^2} & p+q \quad \text{odd}, \end{cases} \tag{B.4}$$

$$\int_0^\pi \frac{\sin p\theta \, \sin q\theta}{\sin\theta} \, d\theta = \begin{cases} 0 & p+q \quad \text{odd}, \\ \frac{1}{2}\left[J_{|(p+q)/2|} - J_{|(p-q)/2|}\right] & p+q \quad \text{even}, \end{cases} \tag{B.5}$$

in which

$$J_k = \begin{cases} 0 & k = 0, \\ 4\sum_{i=1}^{k} \dfrac{1}{2i-1} & k \geq 1. \end{cases} \tag{B.6}$$

We now proceed by finding the closed form solutions for the following integrals.

▶ $a_{mn} = \int_{-1}^{+1} T_m(\xi)\, T_n(\xi)\, \xi\, d\xi$

$$\begin{aligned} a_{mn} &= -\int_{\pi}^{0} \cos m\theta\, \cos n\theta\, \cos\theta\, \sin\theta\, d\theta, \\ &= \frac{1}{2}\int_{0}^{\pi} \sin 2\theta\, \cos m\theta\, \cos n\theta\, d\theta. \end{aligned} \tag{B.7}$$

Using the trigonometric identity

$$\cos m\theta\, \cos n\theta = \frac{1}{2}[\cos(m+n)\theta + \cos(m-n)\theta], \tag{B.8}$$

we have

$$a_{mn} = \frac{1}{4}\int_{0}^{\pi} \sin 2\theta\, \cos(m+n)\theta\, d\theta + \frac{1}{4}\int_{0}^{\pi} \sin 2\theta\, \cos(m-n)\theta\, d\theta, \tag{B.9}$$

which can be written as

$$a_{mn} = \begin{cases} 0 & m+n \ \text{ even}, \\ \frac{1}{4-(m+n)^2} + \frac{1}{4-(m-n)^2} & m+n \ \text{ odd}, \end{cases} \tag{B.10}$$

using (B.4).

▶ $b_{mn} = \int_{-1}^{+1} T_m(\xi)\, T_n(\xi)\, d\xi$

In a similar way, one can obtain

$$b_{mn} = \begin{cases} 0 & m+n \ \text{ odd}, \\ \frac{1}{1-(m+n)^2} + \frac{1}{1-(m-n)^2} & m+n \ \text{ even}. \end{cases} \tag{B.11}$$

► $c_{mn} = \int_{-1}^{+1} \frac{dT_m}{d\xi} \frac{dT_n}{d\xi} \xi \, d\xi$

$$c_{mn} = -\int_{\pi}^{0} \frac{m \sin m\theta}{\sin\theta} \frac{n \sin n\theta}{\sin\theta} \cos\theta \, \sin\theta \, d\theta,$$

$$= mn \int_{0}^{\pi} \frac{\sin m\theta \, \sin n\theta \, \cos\theta}{\sin\theta} d\theta. \tag{B.12}$$

Using the trigonometric identity

$$\sin m\theta \, \cos n\theta = \frac{1}{2}[\sin(m+n)\theta + \sin(m-n)\theta], \tag{B.13}$$

we can write

$$c_{mn} = \frac{mn}{2} \int_{0}^{\pi} \frac{\sin m\theta \, \sin(n+1)\theta}{\sin\theta} d\theta + \frac{mn}{2} \int_{0}^{\pi} \frac{\sin m\theta \, \sin(n-1)\theta}{\sin\theta} d\theta. \tag{B.14}$$

Or, using (B.5)

$$c_{mn} = \begin{cases} 0 & m+n \text{ even}, \\ \frac{mn}{4}\left[J_{|(m+n+1)/2|} - J_{|(m-n-1)/2|} + J_{|(m+n-1)/2|} - J_{|(m-n+1)/2|}\right] & m+n \text{ odd}. \end{cases} \tag{B.15}$$

Note that when $(m+n)$ is odd, so will $(m-n)$, and so on.

► $d_{mn} = \int_{-1}^{+1} \frac{dT_m}{d\xi} \frac{dT_n}{d\xi} d\xi$

Following a similar procedure, one finds

$$d_{mn} = \begin{cases} 0 & m+n \text{ odd}, \\ \frac{mn}{2}\left[J_{|(m+n)/2|} - J_{|(m-n)/2|}\right] & m+n \text{ even}. \end{cases} \tag{B.16}$$

► $e_{lmn} = \int_{-1}^{+1} T_l(\xi) \, T_m(\xi) \, T_n(\xi) \, d\xi$

$$e_{lmn} = -\int_{\pi}^{0} \cos l\theta \, \cos m\theta \, \cos n\theta \, \sin\theta \, d\theta,$$

$$= \int_{0}^{\pi} \cos l\theta \, \cos m\theta \, \cos n\theta \, \sin\theta \, d\theta. \tag{B.17}$$

Repeated application of (B.8), leads to

$$e_{lmn} = \frac{1}{4}\int_0^{\pi} \sin\theta\,[\cos(l+m+n)\theta+\cos(l+m-n)\theta+\cos(l-m+n)\theta+\cos(l-m-n)\theta]\,d\theta, \tag{B.18}$$

or, after using (B.4)

$$e_{lmn} = \begin{cases} 0 & l+m+n \text{ odd}, \\ \frac{1}{2}\left[\frac{1}{1-(l+m+n)^2}+\frac{1}{1-(l+m-n)^2}+\frac{1}{1-(l-m+n)^2}+\frac{1}{1-(l-m-n)^2}\right] & l+m+n \text{ even}. \end{cases} \tag{B.19}$$

► $f_{lmn} = \displaystyle\int_{-1}^{+1} \frac{dT_l}{d\xi}\,T_m(\xi)\,T_n(\xi)\,d\xi$

$$\begin{aligned} f_{lmn} &= -\int_{\pi}^{0} \frac{l\sin l\theta}{\sin\theta}\cos m\theta\,\cos n\theta\,\sin\theta\,d\theta, \\ &= l\int_0^{\pi} \sin l\theta\cos m\theta\,\cos n\theta\,d\theta, \\ &= \frac{l}{2}\int_0^{\pi} \sin l\theta\,[\cos(m+n)\theta+\cos(m-n)\theta]\,d\theta, \end{aligned} \tag{B.20}$$

or

$$f_{lmn} = \begin{cases} 0 & l+m+n \text{ even}, \\ \frac{l^2}{l^2-(m+n)^2}+\frac{l^2}{l^2-(m-n)^2} & l+m+n \text{ odd}. \end{cases} \tag{B.21}$$

► $g_{lmn} = \displaystyle\int_{-1}^{+1} \frac{dT_l}{d\xi}\,\frac{dT_m}{d\xi}\,T_n(\xi)\,d\xi$

$$\begin{aligned} g_{lmn} &= -\int_{\pi}^{0} \frac{l\sin l\theta}{\sin\theta}\,\frac{m\sin m\theta}{\sin\theta}\cos n\theta\,\sin\theta\,d\theta, \\ &= lm\int_0^{\pi} \frac{\sin l\theta\sin m\theta\cos n\theta}{\sin\theta}\,d\theta, \\ &= \frac{lm}{2}\int_0^{\pi} \frac{\sin l\theta}{\sin\theta}\,[\sin(m+n)\theta+\sin(m-n)\theta]\,d\theta. \end{aligned} \tag{B.22}$$

Using (B.5), gives

$$g_{lmn} = \begin{cases} 0 & l+m+n \quad \text{odd}, \\ \frac{lm}{4}\left[J_{|(l+m+n)/2|} - J_{|(l-m-n)/2|} + J_{|(l+m-n)/2|} - J_{|(l-m+n)/2|}\right] & l+m+n \quad \text{even}. \end{cases} \tag{B.23}$$

Appendix C

Evaluation of Submatrices

Let us define

$$E_{ijk} = \int_{-1}^{+1} h_i(\xi)\, h_j(\xi)\, h_k(\xi)\, d\xi, \tag{C.1}$$

$$F_{ijk} = \int_{-1}^{+1} \frac{dh_i(\xi)}{d\xi}\, h_j(\xi)\, h_k(\xi)\, d\xi, \tag{C.2}$$

$$G_{ijk} = \int_{-1}^{+1} \frac{dh_i(\xi)}{d\xi}\, \frac{dh_j(\xi)}{d\xi}\, h_k(\xi)\, d\xi. \tag{C.3}$$

If we write $h_i(\xi)$, the Lagrange interpolating polynomial, as

$$h_i(\xi) = \mu_{in} T_n(\xi), \qquad \text{where} \quad \mu_{in} = \frac{2}{N c_i} \sum_{n=0}^{N} \frac{1}{c_n} T_n(\xi_i), \tag{C.4}$$

integrals (C.1)-(C.3) can be recast in the following compact form

$$E_{ijk} = \mu_{il}\, \mu_{jm}\, \mu_{kn} \int_{-1}^{+1} T_l(\xi)\, T_m(\xi)\, T_n(\xi)\, d\xi = \mu_{il}\, \mu_{jm}\, \mu_{kn}\, e_{ijk}, \tag{C.5}$$

$$F_{ijk} = \mu_{il}\, \mu_{jm}\, \mu_{kn} \int_{-1}^{+1} \frac{dT_l}{d\xi}\, T_m(\xi)\, T_n(\xi)\, d\xi = \mu_{il}\, \mu_{jm}\, \mu_{kn}\, f_{ijk}, \tag{C.6}$$

$$G_{ijk} = \mu_{il}\, \mu_{jm}\, \mu_{kn} \int_{-1}^{+1} \frac{dT_l}{d\xi}\, \frac{dT_m}{d\xi}\, T_n(\xi)\, d\xi = \mu_{il}\, \mu_{jm}\, \mu_{kn}\, g_{ijk}, \tag{C.7}$$

in which e_{ijk}, f_{ijk}, and g_{ijk} are given in Appendix B.

We can now proceed to derive formulas for the evaluation of submatrices (2.55)-(2.67). Note that the integrals over the elements are in fact volume integrals over the master element, i.e.,

$$\int_{\Omega^e} = \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1}. \tag{C.8}$$

Starting with (2.55)

$$\mathcal{A}_{ijk,lmn} = \int_{\Omega^e} (\rho_0/c_0^2)\, h_i(\xi)\, h_j(\eta)\, h_k(\zeta)\, h_l(\xi)\, h_m(\eta)\, h_n(\zeta)\, |J|\, d\xi\, d\eta\, d\zeta, \tag{C.9}$$

the integral becomes

$$\mathcal{A}_{ijk,lmn} = \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} (\rho_0/c_0^2)\, h_i(\xi)\, h_j(\eta)\, h_k(\zeta)\, h_l(\xi)\, h_m(\eta)\, h_n(\zeta)\, |J|\, d\xi\, d\eta\, d\zeta. \tag{C.10}$$

Using the Gauss-Chebyshev quadrature to evaluate the above integral, we have

$$\mathcal{A}_{ijk,lmn} = \sum_{pqr} (\rho_0/c_0^2)_{pqr} \left(\int_{-1}^{+1} h_i(\xi)\, h_l(\xi)\, h_p(\xi)\, d\xi\right)\left(\int_{-1}^{+1} h_j(\eta)\, h_m(\eta)\, h_q(\eta)\, d\eta\right)$$

$$\left(\int_{-1}^{+1} h_k(\zeta)\, h_n(\zeta)\, h_r(\zeta)\, d\zeta\right) |J|_{pqr}. \tag{C.11}$$

Or, using (C.5)

$$\mathcal{A}_{ijk,lmn} = \sum_{pqr} (\rho_0/c_0^2)_{pqr}\, E_{ilp}\, E_{jmq}\, E_{knr}\, |J|_{pqr}. \tag{C.12}$$

Submatrices of the form $\mathcal{A}^{xx}_{ijk,lmn}$, $\mathcal{A}^{xy}_{ijk,lmn}$, $\mathcal{A}^{xz}_{ijk,lmn}$, ..., which involve double derivatives, can be evaluated in a similar fashion. We show this for $\mathcal{A}^{xy}_{ijk,lmn}$ in detail and then provide the general formula in matrix form for the evaluation of other integrals.

Recall that

$$\mathcal{A}^{xy}_{ijk,lmn} = \int_{\Omega^e} (\rho_0 u_0 v_0/c_0^2)\, \mathcal{D}^x_{ijk}\, \mathcal{D}^y_{lmn}\, |J|^{-1}\, d\xi\, d\eta\, d\zeta. \tag{C.13}$$

From (2.72) and (2.73), we have

$$\mathcal{D}^x_{ijk} = \mathcal{D}^x\left[h_i(\xi)\, h_j(\eta)\, h_k(\zeta)\right],$$

$$= \left(\mathcal{J}_{11}\frac{dh_i}{d\xi}\, h_j\, h_k\right) + \left(\mathcal{J}_{12}\, h_i \frac{dh_j}{d\eta}\, h_k\right) + \left(\mathcal{J}_{13}\, h_i\, h_j \frac{dh_k}{d\zeta}\right), \tag{C.14}$$

and

$$\mathcal{D}^y_{lmn} = \mathcal{D}^y\,[h_l(\xi)\,h_m(\eta)\,h_n(\zeta)],$$
$$= \Big(\mathcal{J}_{21}\,\frac{dh_l}{d\xi}\,h_m\,h_n\Big) + \Big(\mathcal{J}_{22}\,h_l\,\frac{dh_m}{d\eta}\,h_n\Big) + \Big(\mathcal{J}_{23}\,h_l\,h_m\,\frac{dh_n}{d\zeta}\Big). \qquad \text{(C.15)}$$

So, the product, $\mathcal{D}^x_{ijk}\,\mathcal{D}^y_{lmn}$, becomes

$$\mathcal{D}^x_{ijk}\,\mathcal{D}^y_{lmn} = \Big(\mathcal{J}_{11}\,\frac{dh_i}{d\xi}\,h_j\,h_k\Big)\Big(\mathcal{J}_{21}\,\frac{dh_l}{d\xi}\,h_m\,h_n\Big) + \Big(\mathcal{J}_{11}\,\frac{dh_i}{d\xi}\,h_j\,h_k\Big)\Big(\mathcal{J}_{22}\,h_l\,\frac{dh_m}{d\eta}\,h_n\Big)$$
$$+ \Big(\mathcal{J}_{11}\,\frac{dh_i}{d\xi}\,h_j\,h_k\Big)\Big(\mathcal{J}_{23}\,h_l\,h_m\,\frac{dh_n}{d\zeta}\Big) + \ldots, \qquad \text{(C.16)}$$

and the integral

$$\mathcal{A}^{xy}_{ijk,lmn} = \int_{\Omega^e} (\rho_0 u_0 v_0/c_0^2)\,\mathcal{D}^x_{ijk}\,\mathcal{D}^y_{lmn}\,|J|^{-1}\,d\xi\,d\eta\,d\zeta,$$
$$= \int_{\Omega^e} (\rho_0 u_0 v_0/c_0^2)\,\Big(\frac{\mathcal{J}_{11}\mathcal{J}_{21}}{|J|}\Big)(\frac{dh_i}{d\xi}\frac{dh_l}{d\xi})(h_j h_m)(h_k h_n)\,d\xi\,d\eta\,d\zeta$$
$$+ \int_{\Omega^e} (\rho_0 u_0 v_0/c_0^2)\,\Big(\frac{\mathcal{J}_{11}\mathcal{J}_{22}}{|J|}\Big)(\frac{dh_i}{d\xi}h_l)(h_j\frac{dh_m}{d\eta})(h_k h_n)\,d\xi\,d\eta\,d\zeta$$
$$+ \int_{\Omega^e} (\rho_0 u_0 v_0/c_0^2)\,\Big(\frac{\mathcal{J}_{11}\mathcal{J}_{23}}{|J|}\Big)(\frac{dh_i}{d\xi}h_l)(h_j h_m)(h_k\frac{dh_n}{d\zeta})\,d\xi\,d\eta\,d\zeta$$
$$+ \ldots. \qquad \text{(C.17)}$$

Using the Gauss-Chebyshev quadrature to evaluate the terms involving mean flow variables and Jacobian terms, we have

$$\mathcal{A}^{xy}_{ijk,lmn} = \sum_{pqr} (\rho_0 u_0 v_0/c_0^2)_{pqr}$$
$$\Big\{ \Big(\frac{\mathcal{J}_{11}\mathcal{J}_{21}}{|J|}\Big)_{pqr}\Big(\int_{-1}^{+1}\frac{dh_i}{d\xi}\frac{dh_l}{d\xi}h_p\,d\xi\Big)\Big(\int_{-1}^{+1}h_j h_m h_q\,d\eta\Big)\Big(\int_{-1}^{+1}h_k h_n h_r\,d\zeta\Big)$$
$$+ \Big(\frac{\mathcal{J}_{11}\mathcal{J}_{22}}{|J|}\Big)_{pqr}\Big(\int_{-1}^{+1}\frac{dh_i}{d\xi}h_l h_p\,d\xi\Big)\Big(\int_{-1}^{+1}h_j\frac{dh_m}{d\eta}h_q\,d\eta\Big)\Big(\int_{-1}^{+1}h_k h_n h_r\,d\zeta\Big)$$
$$+ \Big(\frac{\mathcal{J}_{11}\mathcal{J}_{23}}{|J|}\Big)_{pqr}\Big(\int_{-1}^{+1}\frac{dh_i}{d\xi}h_l h_p\,d\xi\Big)\Big(\int_{-1}^{+1}h_j h_m h_q\,d\eta\Big)\Big(\int_{-1}^{+1}h_k\frac{dh_n}{d\zeta}h_r\,d\zeta\Big) + \ldots\Big\}. \qquad \text{(C.18)}$$

Simplifying the integral using (C.1)-(C.3) leads to

$$\begin{aligned}\mathcal{A}^{xy}_{ijk,lmn} = \Big\{ &\Big[(\mathcal{J}_{21})_{pqr}(G_{ilp}\,E_{jmq}\,E_{knr}) + (\mathcal{J}_{22})_{pqr}(F_{ilp}\,F_{mjq}\,E_{knr}) + \\ &(\mathcal{J}_{23})_{pqr}(F_{ilp}\,E_{jmq}\,F_{nkr})\Big](\mathcal{J}_{11})_{pqr} \\ +&\Big[(\mathcal{J}_{21})_{pqr}(F_{lip}\,F_{jmq}\,E_{knr}) + (\mathcal{J}_{22})_{pqr}(E_{ilp}\,G_{jmq}\,E_{knr}) + \\ &(\mathcal{J}_{23})_{pqr}(E_{ilp}\,F_{jmq}\,F_{nkr})\Big](\mathcal{J}_{12})_{pqr} \\ +&\Big[(\mathcal{J}_{21})_{pqr}(F_{lip}\,E_{jmq}\,F_{knr}) + (\mathcal{J}_{22})_{pqr}(E_{ilp}\,F_{mjq}\,F_{knr}) + \\ &(\mathcal{J}_{23})_{pqr}(E_{ilp}\,E_{jmq}\,G_{knr})\Big](\mathcal{J}_{13})_{pqr}\Big\}\left(\frac{\rho_0 u_0 v_0/c_0^2}{|J|}\right)_{pqr}, \end{aligned} \tag{C.19}$$

where summation over repeated indices, pqr, is assumed. Other submatrices can be obtained in a similar way. They can be written in a compact form using the matrix notation as

$$\mathcal{A} = \mathcal{B}_{pqr} \odot \Big([\mathcal{J}]_{pqr}\,\mathcal{C}\,[\mathcal{J}]^T_{pqr}\Big), \tag{C.20}$$

in which $[\mathcal{J}]$ is given by (2.72). $\mathcal{A}$ is the matrix of submatrices

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}^{xx}_{ijk,lmn} & \mathcal{A}^{xy}_{ijk,lmn} & \mathcal{A}^{xz}_{ijk,lmn} \\ \mathcal{A}^{yx}_{ijk,lmn} & \mathcal{A}^{yy}_{ijk,lmn} & \mathcal{A}^{yz}_{ijk,lmn} \\ \mathcal{A}^{zx}_{ijk,lmn} & \mathcal{A}^{zy}_{ijk,lmn} & \mathcal{A}^{zz}_{ijk,lmn} \end{bmatrix}, \tag{C.21}$$

$\mathcal{B}$ is the matrix containing mean flow values

$$\mathcal{B} = \left(\frac{\rho_0/c_0^2}{|J|}\right)\begin{bmatrix} (u_0^2 - c_0^2) & u_0 v_0 & u_0 w_0 \\ v_0 u_0 & (v_0^2 - c_0^2) & v_0 w_0 \\ w_0 u_0 & w_0 v_0 & (w_0^2 - c_0^2) \end{bmatrix}, \tag{C.22}$$

with $\odot$ representing pointwise matrix multiplication, and $\mathcal{C}$ is given by

$$\mathcal{C} = \begin{bmatrix} G_{ilp}\,E_{jmq}\,E_{knr} & F_{lip}\,F_{jmq}\,E_{knr} & F_{lip}\,E_{jmq}\,F_{knr} \\ F_{ilp}\,F_{mjq}\,E_{knr} & E_{ilp}\,G_{jmq}\,E_{knr} & E_{ilp}\,F_{mjq}\,F_{knr} \\ F_{ilp}\,E_{jmq}\,F_{nkr} & E_{ilp}\,F_{jmq}\,F_{nkr} & E_{ilp}\,E_{jmq}\,G_{knr} \end{bmatrix}. \tag{C.23}$$

Similarly, integrals involving single derivatives can be written in the compact matrix form

$$\begin{bmatrix} \mathcal{A}^{x\star}_{ijk,lmn} & \mathcal{A}^{\star x}_{ijk,lmn} \\ \mathcal{A}^{y\star}_{ijk,lmn} & \mathcal{A}^{\star y}_{ijk,lmn} \\ \mathcal{A}^{z\star}_{ijk,lmn} & \mathcal{A}^{\star z}_{ijk,lmn} \end{bmatrix} = \mathcal{D}_{pqr}\,[\mathcal{J}]_{pqr}\,\mathcal{E}_{pqr}, \tag{C.24}$$

where $\mathcal{D}$ is the diagonal matrix of mean flow variables

$$\mathcal{D} = \left(\frac{\rho_0}{c_0^2}\right) \begin{bmatrix} u_0 & 0 & 0 \\ 0 & v_0 & 0 \\ 0 & 0 & w_0 \end{bmatrix}, \tag{C.25}$$

and

$$\mathcal{E} = \begin{bmatrix} F_{ilp}\,E_{jmq}\,E_{knr} & F_{lip}\,E_{jmq}\,E_{knr} \\ E_{ilp}\,F_{jmq}\,E_{knr} & E_{ilp}\,F_{mjq}\,E_{knr} \\ E_{ilp}\,E_{jmq}\,F_{knr} & E_{ilp}\,E_{jmq}\,F_{nkr} \end{bmatrix}. \tag{C.26}$$

Appendix D

Discretization of the Eigenvalue Problem

Here, we present the details regarding discretization of the Bessel equation, Eq. (2.93), in the radial direction. Derivation begins using the weak form, Eq. (2.102), rewritten below

$$\int_{\Sigma} \left(\frac{d\varphi}{dr}\frac{dw}{dr} + \frac{m^2}{r^2}\varphi w \right) r\, dr - k_{ms}^2 \int_{\Sigma} \varphi w r\, dr = 0. \tag{D.1}$$

In this equation, Σ is the one-dimensional region in the radial direction, which is divided into E elements, each of size L^e. As shown in Figure D.1, a mapping is defined between

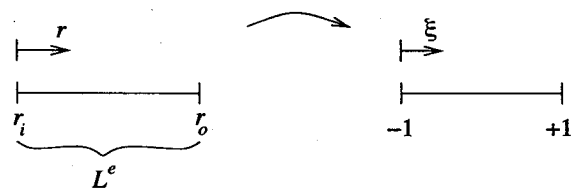

Figure D.1: One-dimensional mapping in the radial direction.

the elements in the physical plane and the reference element in the computational plane, $\hat{\Omega} = [-1, +1]$, given by

$$r = \frac{L^e}{2}\xi + \frac{(r_i + r_o)}{2}, \qquad \rightarrow \qquad dr = \frac{L^e}{2}d\xi, \tag{D.2}$$

in which r_i and r_o are the coordinates of the initial and end points of an element in the r-direction, respectively, and $L^e = (r_o - r_i)$. Let us represent the acoustic potential as the dis-

crete sum of Lagrange interpolation functions, $h_j(\xi)$, collocated at the Gauss-Chebyshev-Lobatto points

$$\varphi^e(\xi) = \sum_{j=0}^{N} h_j(\xi)\,\varphi_j, \tag{D.3}$$

where N is the order of spectral approximation. $h_j(\xi)$, given by (2.47), can also be written in the following compact form

$$h_j(\xi) = \mu_{jn} T_n(\xi), \qquad \text{where} \quad \mu_{jn} = \frac{2}{N c_j} \sum_{n=0}^{N} \frac{1}{c_n} T_n(\xi_j), \tag{D.4}$$

in which summation over repeated indices is assumed. Taking the weight function to be $w = h_i(\xi)$, and integrating over an element, Σ^e, the integrals in (D.1) become

$$\int_{\Sigma^e} \left(\frac{d\varphi}{dr}\frac{dw}{dr} + \frac{m^2}{r^2}\varphi w \right) r\,dr = \sum_{j=0}^{N} \left(\int_{\Sigma^e} \frac{dh_i}{dr}\frac{dh_j}{dr}\, r\,dr + \int_{\Sigma^e} \frac{m^2}{r} h_i\, h_j\, dr \right) \varphi_j,$$

$$= \sum_{j=0}^{N} K^e_{ij}\,\varphi_j, \tag{D.5}$$

and

$$\int_{\Sigma^e} \varphi w r\,dr = \sum_{j=0}^{N} \left(\int_{\Sigma^e} h_i\, h_j\, r\,dr \right) \varphi_j,$$

$$= \sum_{j=0}^{N} M^e_{ij}\,\varphi_j. \tag{D.6}$$

Coefficient matrices K^e_{ij} and M^e_{ij} are evaluated using the transformation defined in (D.2). Starting with K^e_{ij}, we have

$$K^e_{ij} = \int_{\Sigma^e} \frac{dh_i}{dr}\frac{dh_j}{dr}\, r\,dr + \int_{\Sigma^e} \frac{m^2}{r} h_i\, h_j\, dr,$$

$$= \int_{-1}^{+1} \frac{dh_i}{d\xi}\frac{dh_j}{d\xi}\,(\xi+\alpha)\,d\xi + m^2 \int_{-1}^{+1} \frac{h_i\, h_j}{(\xi+\alpha)}\,d\xi, \tag{D.7}$$

in which $\alpha = (r_i + r_o)/L^e$. To evaluate the integrals, we use the definition of the Chebyshev polynomials, given by (D.4) and (2.35), to obtain

$$K^e_{ij} = \mu_{il}\,\mu_{jn} \Big[\int_{-1}^{+1} \frac{dT_l}{d\xi}\frac{dT_n}{d\xi}\,\xi\,d\xi + \alpha \int_{-1}^{+1} \frac{dT_l}{d\xi}\frac{dT_n}{d\xi}\,d\xi + m^2 \int_{-1}^{+1} \frac{T_l\, T_n}{(\xi+\alpha)}\,d\xi \Big]. \tag{D.8}$$

or, simply

$$K_{ij}^{e} = \mu_{il}\,\mu_{jn}\Big[\,(c_{ln} + \alpha\, d_{ln}) + m^2 \int_0^{\pi} \frac{\cos l\theta\, \cos n\theta\, \sin\theta}{(\cos\theta + \alpha)}\, d\theta\,\Big], \tag{D.9}$$

where c_{ln} and d_{ln} are defined in Appendix B, and the last integral on the right-hand side is evaluated using Gauss-Legendre quadrature, because it does not have a closed-form solution.

Similarly, we can write for M_{ij}^{e}

$$\begin{aligned} M_{ij}^{e} &= \int_{\Sigma^e} h_i\, h_j\, r\, dr \\ &= \frac{(L^e)^2}{4} \int_{-1}^{+1} h_i\, h_j\, (\xi + \alpha)\, d\xi, \\ &= \frac{(L^e)^2}{4}\, \mu_{il}\, \mu_{jn} \Big[\int_{-1}^{+1} T_l\, T_n\, \xi\, d\xi + \alpha \int_{-1}^{+1} T_l\, T_n\, d\xi \Big], \end{aligned} \tag{D.10}$$

or

$$M_{ij}^{e} = \frac{(L^e)^2}{4}\, \mu_{il}\, \mu_{jn}\, (\, a_{ln} + \alpha\, b_{ln}\,). \tag{D.11}$$

Matrices a_{ln} and b_{ln} are given in Appendix B. Once the coefficient matrices are created for all elements, their assembly gives the global eigenvalue problem, Equation (2.103)

$$([K] - k_{ms}^2[M])\{\varphi\} = 0, \tag{D.12}$$

which is solved using LAPACK routines to obtain the eigenvalue k_{ms}.