University Of Southampton

Faculty of Engineering and Physical Sciences

PhD Thesis

Acoustics Group

NON-REFLECTIVE BOUNDARY CONDITIONS FOR PREDICTION OF ACOUSTIC TONES IN NON-UNIFORM MEAN FLOW FOR DUCTED FLOWS

Author:
Kamran Zaabar

Supervisor:
Prof. A.G. Wilson
Dr. A. McAlpine

February 24, 2022

Abstract

New Non-Reflective Boundary Conditions (NRBC) are developed and validated for computational fluid dynamics (CFD) prediction of acoustic noise at discrete frequencies in ducts containing non-uniform mean flow. The new boundary conditions are implemented in a commercial time-unsteady CFD solver and validated in uniform and non-uniform mean flow over a range of wavenumbers and frequencies in two and three dimensions. A literature review is provided, discussing the ideas and state-of-the-art in non-reflecting boundary treatments. The test calculations are repeated with a subset of the treatments most commonly used in acoustic CFD calculations to allow comparison to be made with the current method. The primary target of the work is the prediction of acoustic tones in turbomachinery, and a final demonstration is performed for an acoustically representative test case modelling the generation and propagation of vortex/stator interaction tones. In addition to accommodating fully non-uniform (circumferentially as well as radially) mean flow the new boundary conditions are able to accommodate non-uniform duct geometries, including non-planar bounding surfaces at inlet and exit, although these capabilities have not yet been demonstrated. A discussion is provided of the results of the test calculations and it is concluded that the new boundary conditions offer the best overall capability of the methods tested for the target calculation type, with reflection coefficients of around -40dB and -20dB at the inlet and outlet boundaries respectively. Recommendations are given for future work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis discusses the development of boundary conditions to be used in solving the wave equation in ducted geometries with flow that is both uniform and non-uniform. The system of equations are hyperbolic Partial Differential Equations (PDE). In order to obtain a unique solution for a hyperbolic PDE, it is a requirement to specify what is known as an initial condition, or initial solution, which is typically the state of the solution at time $t = 0$ as well as impose conditions on the boundary of the domain. The resulting problem is then called an Initial-Boundary Value Problem (IBVP). For an example of a boundary and boundary condition one can picture a guitar string. The boundaries to this problem would be the two points where the string is tied down. The boundary condition here would be the displacement at those points is zero. It is important to impose the correct initial and boundary conditions for the problem at hand as initial conditions and boundary conditions can influence the solution. In numerical simulations, that is calculations done numerically, it is not practical or indeed possible to calculate a solution where the boundary is at infinity. As a result truncation of the domain is required in the numerical calculation of hyperbolic PDE's and by extension new boundary conditions have to be found at these new artificial boundaries.

Numerical methods offer engineers and physicists a means to model complicated physical systems, in a precise and accurate manner assuming correct boundary treatments. A system can be considered complicated due to the equations of motion being difficult to solve analytically such as the full Navier-Stokes equations whose solution is yet to be found. While analytical solutions for the full equation may not exist currently many problems can take assumptions that simplify the equations and can be solved via a numerical method.

This thesis is concerned with development of Eigen Based Non-Reflecting Boundary Conditions (EBNRBC) for use in Computational Fluid Dynamics (CFD). CFD is frequently chosen as a tool for high fidelity prediction of tone noise in turbomachinery. CFD calculations are based on a mesh of nodes/cells representing a truncated finite domain around the region of interest. This method works on a point by point basis in space and time on a finite mesh of points. The three most common implementations of CFD are finite difference [11], finite volume [12] and finite element [13]. Values, such as the pressure and velocities, on one node on the mesh are calculated using the values on the surrounding nodes. Steady calculations exist that are calculations to problems that do not depend on time and so there is no progression in time. This results in the need to find appropriate boundary conditions on the edge of these domains. The values at the boundary must be set by the user as at the boundary there are no nodes on “the other side” of the boundary to help calculate what the values at the boundary should be. Many general three dimensional boundary conditions interact with the solution causing spurious non-physical reflections at the boundary which contaminate the interior solution. The most common example is the constant pressure boundary condition as seen in figure 1.1. The boundaries to the domain are fundamentally reflective however the domain is large giving time to allow the acoustic waves to decay as they approaches the far field. This boundary treatment can therefore be acceptable if it is placed a considerable distance away from the interior solution. While boundary treatments of this type can give accurate results a large domain may not always be ideal as depending on the problem this may require more compute power and compute resources. In ducted flows, constant pressure boundary conditions fail totally as cut on acoustic modes travelling upstream and downstream propagate over large distances without decay. Other more advanced boundary treatments will soon be highlighted.

The jet engine itself has multiple sources of noise:

- Noise sources come from the fan and stator blades and the interaction between them. This noise source is both tonal and broadband.
- Broadband and tonal noise coming from the compressor.
- Broadband noise originating from the combustion within the engine.
- Tonal turbine noise and jet noise.
- Broadband noise coming from the exhaust due to the shear layer.

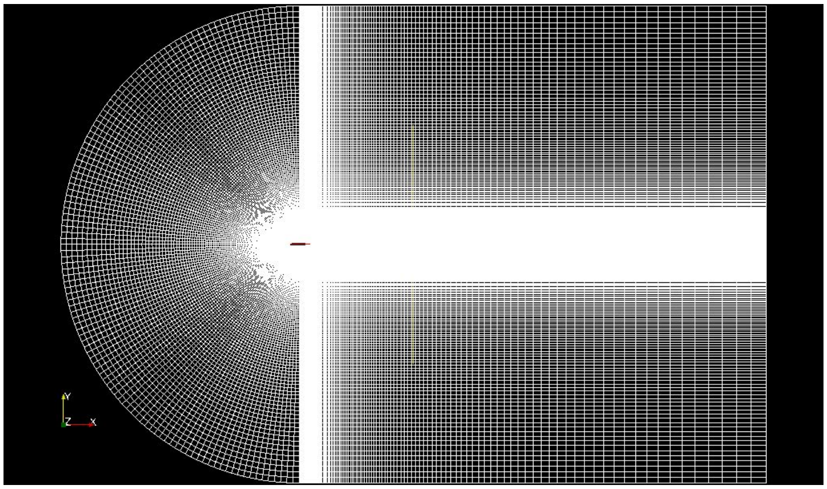


Figure 1.1: This shows a large constant pressure boundary domain. The mesh also stretches into the far field. The reason for this stretching is to help improve solution accuracy and will be discussed in more detail later on. The image above was made in ANSYS Meshing

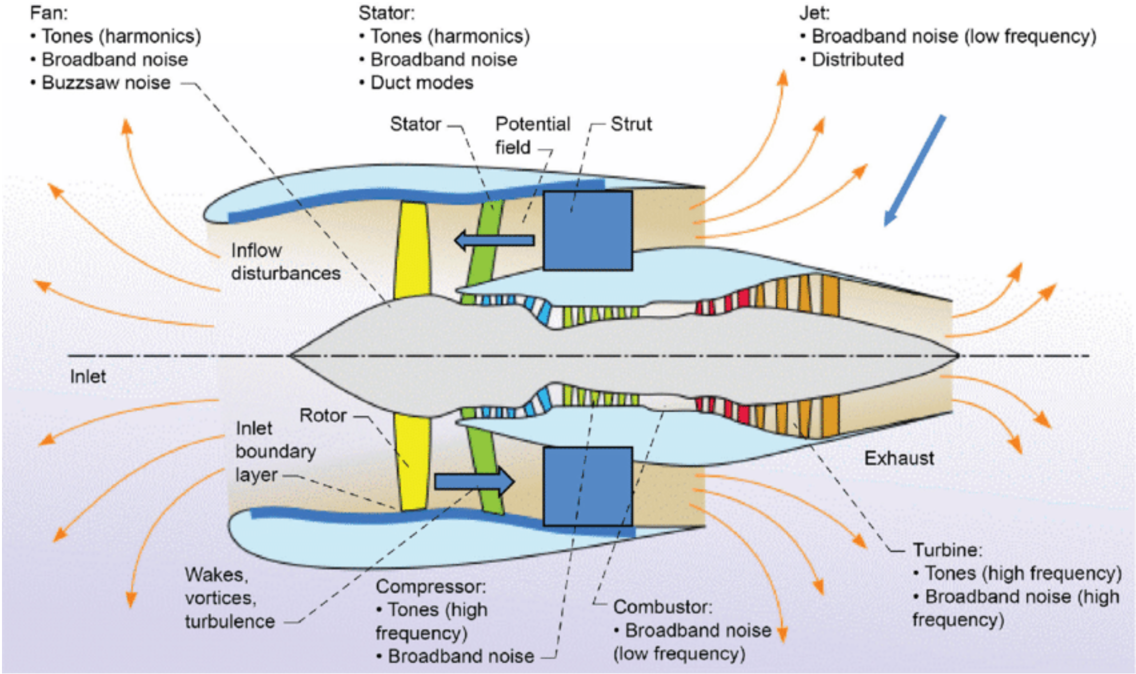


Figure 1.2: This image shows the primary noise sources of a jet engine. Image taken from [1]

The requirement for tonal boundary conditions comes from the desire to simulate tonal noise coming from the jet engine rotor/stator blades and especially the fan stage which is one of most significant noise sources. The rotation of these blades relative to each other, produces a frequency whose value is known and can be used as an input into NRBC's.

There are many types of boundary conditions and boundary treatments to help improve the accuracy of numerical simulation. In this writing there will be a distinction between a boundary condition and a boundary treatment. A boundary condition is what is set at the boundaries of the domain and what is required to solve the equations of motion analogous to what is needed when solving a differential equation analytically. A boundary treatment will be some technique used to help improve accuracy in the simulation that will then be accompanied by a boundary condition so that the equations of motion can be solved. The NRBC here would be a type of boundary condition. Types of boundary treatment include stretching meshes which help to dissipate the physics that take place far from the region of interest. This stretching mesh will then be accompanied by a boundary condition so that the numerical calculation can be completed.

A simulation that uses only a boundary condition as opposed to a boundary treatment will in theory calculate faster and more efficiently, however it is a requirement that these boundary conditions produce accurate results. A failure to do so results in an inaccurate solution; an example of this can be seen in figure 1.3.

EBNRBC's offer the user a means to calculate the solution at the boundary accurately. The method uses eigenanalysis to obtain the linear mode shapes and direction of information travel which are applied in such a way as to let outgoing waves leave the domain without any spurious reflection. An example of an effective EBNRBC can be seen in figure 1.4. These images show how much of a difference can be made by simply having the correct boundary conditions.

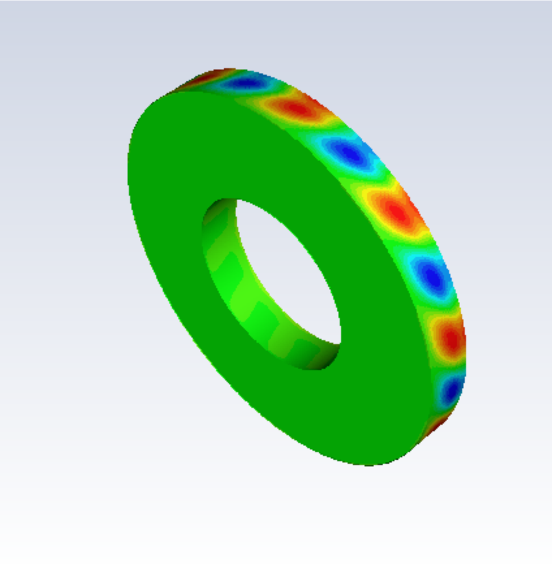


Figure 1.3: A duct with a constant pressure boundary condition at the near boundary. The acoustic wave fails to leave the domain.

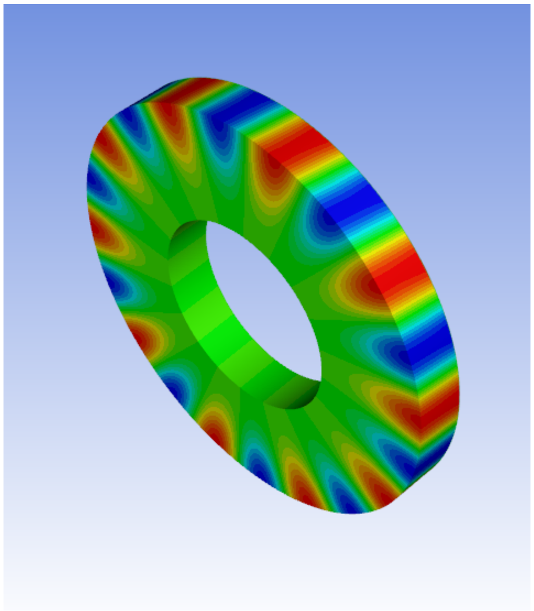


Figure 1.4: A duct with an appropriate boundary condition at the near boundary. The acoustic wave passes through the boundary without distortion.

The problem is made more difficult when considering non-uniform mean flow and upstream boundary conditions. In the images above the signal meets the downstream boundary for which only the pressure has to be set. For waves going against the flow towards the inlet, more flow parameters need to be set, in effect, to account for vortical and entropy waves entering through the upstream boundary. For non-uniform flow acoustic waves are no longer perfect harmonics described by sin waves. The non-uniformity of the flow causes the acoustic waves to distort and deform as different parts of the flow have differing velocities, that is, the acoustic wave scatters. A signal of mode $m = 10$ in a non-uniform flow will scatter into neighbouring harmonics. As the main goal in this thesis is to develop boundary conditions in non-uniform flows, the eigenanalysis needs to account for this scattering of harmonics in order to produce accurate boundary conditions.

1.1 A Brief overview of Non-Reflective Boundary Conditions

There is a wide range of literature discussing NRBC's for many different wave equations with and without flow. A popular type of boundary condition is the local boundary condition. A local boundary condition is where the flow values set on any boundary node depend only on the flow parameters at adjacent nodes within the domain. Commonly used aerodynamic boundary conditions such as fixed total pressure at the inlet or fixed static pressure at the outlet are local boundary conditions however are known to be highly reflective. The simplest non reflective local boundary conditions treat all acoustic waves as if they were plane waves travelling normal to the boundary. Methods have been proposed to account for non-plane waves at various angles at the expense of becoming increasingly non-local depending on accuracy, that is, using information that is further and further from the boundary node. Local boundary conditions will be looked at in more detail in section 2.4.

Another type of boundary condition is the Eigen Based NRBC (EBNRBC). This is a two step process where firs of all the mean flow and duct geometry are used to determine a complete set of acoustic modes (eigenvectors) travelling upstream and downstream. In the second step the unsteady flow values from every node on the boundary are decomposed into these modes and adjusted to preserve the outgoing acoustic modes while ensuring there are no incoming acoustic modes.

In principle such boundary conditions could be exact as any set of acoustic waves can be decomposed into upstream and downstream travelling modes. In practice a range of simplifications are usually made to make it easier to identify the eigenmmodes. The first EBNRBC by Giles [18] was for a two dimensional duct and uniform mean flow. Later authors extended this to quasi-three dimensional boundary conditions, fully three dimensional calculations for steady state flow simulations, and a curvilinear form in two dimensions for use in complex geometries. Extensions were also made for cases where the geometry was more complex such as elliptical ducts and ducts that varied slowly with axial position. Previous EBNRBC are described in section 2.3 . The work described in this thesis is the first example of a boundary condition which accounts for both non-uniform flow and non-uniform geometry.

1.2 Alternative Boundary Treatments

In addition to the local and eigen-based boundary conditions discussed in the previous section there is a range of alternative boundary treatments that require an extension to the calculation domain beyond the inlet and outlet boundaries.

Stretched grids can be used as buffer zones in simulations of wave propagation. Stretching the grid away from the area of interest lowers the resolution of the mesh making it incapable of accurately calculating the solution at the far field effectively dissipating the outgoing noise. The stretching is gradual as an immediate change in mesh resolution can itself cause reflections. The boundary condition applied may then be the standard constant pressure boundary condition, with the mesh itself stretching to help dissipate the solution before it reaches the boundary which would be reflective. The reflections are then small and negligible. The literature on stretched mesh is considered in section 2.6.2. The test cases in chapters 4 and 5 included the stretched mesh as a comparison method.

A sponge layer is also a method used to dissipate outgoing waves in a solution before they reaches the boundary. The concept here is to add to the linearised euler equations “sponge” terms once the solution gets to the far field. These sponge terms will cause acoustic effects to decay (as well as other wavelike phenomena), however reflections can occur off the sponge layer interface itself. As this is a three dimensional layer it takes up space in the domain and hence uses up computing power and time. Angle of incidence of the waves hitting the sponge also has an effect on the reflection coefficient and thus the effectiveness of the sponge. The sponge layer is described in more detail in section 2.6.3.

Another alternative designed specifically for wave like problems is the Perfectly Matched Layer (PML). Like the sponge layer a PML is effectively a region around the area of interest that decays wave like disturbances. Unlike the sponge layer however it does this via evaluating the solution to the linearised Euler equation at complex values of x at the far field where x is spacial distance normal to the inlet or outlet boundary. This causes the wave like solutions to decay exponentially in the zone where the PML is active. PML is reflectionless only for exact wave problems, after discretising, the surface of the PML between the PML and regular medium is no longer non reflecting. In order to reduce the reflections one must slowly introduce the PML. Again fine tuning the PML is required and a good balance between resources in compute power, time and spurious reflections should be found.

The stretching mesh and sponge layer can theoretically deal with broadband noise however some of the parameters that can be tuned in these methods depend on the frequency of the outgoing waves. Parameters should therefore be tuned to ensure that the frequency region of interest is dissipated effectively. PML's on the other hand can only deal with multiple discrete frequencies. Again tuning parameters depend on frequency so parameters should be set allowing frequencies of interest to leave the domain.

Confirming the earlier distinction; Stretched grids, Sponge layers and PML's are boundary treatments. They offer a means of damping the solution before the solution hits the boundary and so a boundary condition is still required to solve the equations. Stretched grids, sponge layers and PML's are sometimes called absorbing boundary layers, whilst NRBC's are sometimes called Absorbing Boundary Conditions (ABC's).

1.3 Thesis Structure

Further on in this introduction there will be a brief section on the aims and scope of the project.

A literature review is provided in chapter 2. The review will discuss the state of the art for artificial boundary conditions for the wave equation without flow, with flow, and boundary conditions for acoustics travelling through ducts with and without flow. The review will also cover eigenanalysis and its use in developing accurate NRBC's.

A Chapter on methodology and application will then discuss how the boundary conditions are implemented and used numerically.

Chapters 4 and 5 will describe the validation of the method in two and three dimensions. Chapter 6 will provide a demonstration of the newly developed boundary condition for prediction of wake/stator interaction tone noise. Conclusions and future work will be discussed in chapter 7.

1.4 Project Scope and Aims

The focus of this project is to develop non-reflecting boundary conditions suitable for use in computational fluid dynamics (CFD) calculations, such as turbomachinery calculations, which include the prediction of tonal noise sources. The boundary conditions should permit outgoing acoustic waves to pass through with very little spurious reflection. They should work on planar and non-planar boundary surfaces with radially and circumferentially non-uniform mean flow.

The boundary conditions should maintain the benefits of previous eigen-based boundary conditions developed by Giles [18] and Kousen [32] for ducted flows (see sections 2.2.2 through to 2.2.6). Specifically, they should:

- Be applicable at the inlet and/or the outlet surface.
- Be applied on the boundary itself, that is no extension to the domain as with the sponge layer or PML which necessarily includes knowledge of the duct geometry that might not be available.
- Allow for an acoustic or vortical signal to be input to the domain at the same plane as the non-reflecting treatment.

As with these previous boundary conditions it can be assumed that a description of the mean flow is available prior to the unsteady calculation. Previous experience with these treatments suggests that treatments based on the linearised Euler equations are usually sufficient for non-reflection in RANS calculations of turbomachinery ducts without large temperature variations, and the scope of this work is limited to these cases.

The new method should, in addition, extend the state-of-the-art for eigen-based boundary conditions in that it should be applicable

- On general (including non-planar) upstream and downstream boundary surfaces.
- In the presence of circumferentially (as well as radially) varying mean flow.
- Within ducts of non-uniform but smoothly varying geometry.

The boundary conditions should be computationally efficient, in the sense that memory and compute requirements should be small relative to the requirements for the underlying CFD calculation.

The boundary conditions should be validated over a range of operating conditions and compared with existing non-reflecting boundary treatments.

The primary focus of the work is for use in time-dependent CFD calculations, which may have tone noise sources at more than one frequency. It is noted, however, that such a boundary condition could easily be adapted for use in linear (fixed frequency) CFD or Compuational Aeroacoustics (CAA) calculations.

The work in this thesis builds on the eigenanalysis of Wilson [27] as described in section 2.2.7. The development of this eigenanalysis into non-reflecting boundary conditions, including the methodology, code development and validation, was performed entirely within the current project.

Chapter 2

Literature Review

This literature review will be split into two parts. Initially a discussion on literature directly relevant to the theory used in this thesis will take place before giving a more general look at the state of the art regarding numerical boundary conditions for the purposes of acoustic and wave like disturbances. Here in the first part a short review of duct acoustics will be given before moving to the theory needed to develop the EBNRBC's, namely eigenanalysis and its use in developing the boundary conditions as well as papers that formulate and implement EBNRBC's. A wide range of papers will be covered discussing eigenanalysis in uniform and non-uniform flows in one, two and three dimensions as well as their uses in formulating EBNRBC's. Eigenanalysis and development of EBNRBC's will also be discussed in the context of non-uniform geometries. Most of these examples will be looked at in the context of turbomachinery in cylindrical or pseudocylindrical coordinates. These papers will then be used in the development of the work here in this thesis.

In the second half of the literature review numerical boundary conditions in a more general setting will be discussed. The general goals of numerical boundary conditions are discussed. The boundary conditions here will be for numerous forms of the wave equation whose disturbances flow to infinity, with and without flow. A group of approximate eigenanalysis methods originating from Reinstra [3] is introduced for flows in which the geometry varies slowly with axial position. Alternative boundary treatments will then be discussed such as the stretching grid and the perfectly matched layer.

2.1 Fundamentals of Duct-Acoustics

Rienstra [25, 26] discusses the fundamentals of duct acoustics. The ground work is laid out in a cylindrical duct aligned with the x axis with zero flow. The reports introduce the concept of the acoustic mode as an exponential function f, representing propagation in the axial direction, multiplied by some shape function ψ which is an eigenfunction of a Laplace type operator with eigenvalues relating to axial wave number k valid on a duct cross section:

$$p\left(\mathbf{x},t\right) = \psi\left(y,z\right) f\left(x-vt\right) = \psi\left(y,z\right) e^{i\omega\left(t-\frac{x}{v}\right)} = \psi\left(y,z\right) e^{i\omega t - ikx} \tag{2.1}$$

where v is the phase velocity such that the function $f(x-vt)$ represents propagation along the duct axis. The acoustic field is then defined as:

$$p\left(\mathbf{x},t\right) := p\left(\mathbf{x}\right) e^{i\omega t}, \quad \mathbf{u}\left(\mathbf{x},t\right) := \mathbf{u}\left(\mathbf{x}\right) e^{i\omega t} \tag{2.2}$$

Here p is the static pressure, $\mathbf{u}$ is the velocity vector, ω is the angular frequency, t is the time and $\mathbf{x}$ is the spatial vector in cylindrical polar coordinates, r, θ, z. Equation (2.2) satisfies the Helmholtz equation:

$$\nabla^2 p + \frac{\omega^2}{c_0^2} p = 0 \tag{2.3}$$

The acoustic velocity $\mathbf{u}$ is then related to the acoustic pressure via the linearised momentum equation:

$$i\omega\rho_0\mathbf{u} + \nabla p = 0. \tag{2.4}$$

Here c_0 is the mean speed of sound while ρ_0 is the mean density. It is seen that ω^2/c_0^2 is an eigenvalue for the Laplace operator where p is the corresponding eigenfunction. In the current example hard-wall boundary conditions are assumed:

$$\mathbf{u}\cdot\mathbf{n} = 0 \tag{2.5}$$

where $\mathbf{n}$ is an externally directed unit vector normal to the boundary of the duct. Self similar solutions exist such that the spatial part of the pressure field can be separated into $p(x,y,z) = \phi(x)\psi(y,z)$ for $\phi(x) = e^{-ikx}$ with eigenvalues k and corresponding eigenfunctions ψ. As a result the pressure part of (2.3) becomes:

$$\nabla_{yz}^2\psi + \left(\frac{\omega^2}{c_0^2} - k^2\right)\psi = 0 \tag{2.6}$$

The Laplacian operator here is denoted with a subscript yz as ψ is a function only of y and z. As a result the most general form of the spacial part of the pressure field is found:

$$p(x, y, z) = \sum_{n=0}^{\infty} C_n \psi_n(y, z) e^{-ik_n x} \tag{2.7}$$

where ψ_n are the eigenfunctions of the Laplace operator and $\alpha_n^2 = \left(\frac{\omega^2}{c_0^2} - k_n^2\right)$ the corresponding eigenvalues. In the case of a hard-walled boundary α_n^2 give a real positive set of values with $\alpha_1^2 = 0$ and $\psi_1 = 1$. The axial wave number k_n is then given implicitly by:

$$k_n^{\pm} = \pm\sqrt{\frac{\omega^2}{c_0^2} - \alpha_n^2} \tag{2.8}$$

In the series expansion present in equation (2.7) each term represents what is called a duct mode. The wave number determines the behaviour of the mode. If k_n is real and not small i.e. ω/c_0 is more substantial than α_n, then the behaviour of the axial mode is sinusoidal with a non large axial wavelength. As ω/c_0 approaches α_n, k_n becomes small and the axial wavelength becomes large. If $\omega/c_0 = \alpha_n$, i.e. $k_n = 0$, the mode is at resonance and the wavelength is infinite. Finally if $\omega/c_0 < \alpha_n$ then k_n is imaginary and the mode exponentially decays that is becomes cut-off.

It is also noted by Rienstra [25, 26] that assuming the boundary condition is uniform along the circumference of the duct whose cross section is circular or rectangular the symmetry of the geometry gives a symmetric modal shape function, and that the solutions of the eigenvalue problem can be found using the separation of variables method. In a circular duct described by polar coordinates r, θ the eigenfunction ψ can be written as:

$$\psi\left(y, z\right) = f(\theta)g(r) \tag{2.9}$$

where the eigensolutions consist of Fourier harmonics in θ and Bessel functions in r.

2.1.1 Uniform Mean Flow

Moving to a case with uniform flow the isentropic Euler equations are introduced:

$$\frac{\mathrm{d}\,\rho}{\mathrm{d}\,t} + \rho\nabla\cdot\mathbf{u} = 0, \quad \rho\frac{\mathrm{d}\,\mathbf{u}}{\mathrm{d}\,t} + \nabla p = 0, \quad \frac{\mathrm{d}\,s}{\mathrm{d}\,t} = 0 \tag{2.10}$$

Where s is the entropy. The acoustic variables are split into mean and perturbation components: $\mathbf{u} = \mathbf{u}_0 + \mathbf{u}'$, $p = p_0 + p'$, $\rho = \rho_0 + \rho'$, where the subscripted variables are the mean components and the primed variables are the perturbation components. Linearising the Euler equation about the perturbation variables (2.10) and after some algebra which can be found in Rienstra [25] a general wave equation for the pressure is found. This equation describes the pressure of acoustic, that is, isentropic, perturbations of a parallel inviscid mean flow with a uniform mean pressure p_0:

$$\mathrm{D}_0^3 p' - \mathrm{D}_0 \nabla \cdot \left(c_0^2 \nabla p'\right) + 2c_0^2 \frac{\partial}{\partial x}\left(\nabla p' \cdot \nabla u_{x0}\right) = 0 \tag{2.11}$$

where D_0 is the differential operator $\frac{\partial}{\partial t} + u_0 \frac{\partial}{\partial x}$ where we have assumed the flow is a mean parallel flow in the x direction. In order to simplify (2.11) to a form relevant for this thesis, the acoustic perturbation is assumed to be harmonic in time and that the boundary conditions are independent of x. The medium is taken to be circularly symmetric with $u_0 = u_0(r)$ and $c_0 = c_0(r)$ in cylindrical polar coordinates (r, θ, x) such that the solutions take the form

$$p'\left(\mathbf{x}, t\right) = \mathrm{Re}\left(\hat{p}\left(r\right) e^{i\omega t - ikx - im\theta}\right), \quad \mathbf{v}'\left(\mathbf{x}, t\right) = \mathrm{Re}\left(\hat{\mathbf{v}}\left(r\right) e^{i\omega t - ikx - im\theta}\right) \tag{2.12}$$

If it is also assumed that the mean Mach number c_0 and the mean flow u_0 is constant then it is possible to write (2.11) as the Bessel equation:

$$\frac{\mathrm{d}^2 p}{\mathrm{d}r^2} + \frac{1}{r}\frac{\mathrm{d}p}{\mathrm{d}r} + \left(\frac{\Omega^2}{c_0^2} - k^2 - \frac{m^2}{r^2}\right) p = 0 \tag{2.13}$$

where $\Omega = \omega - ku_0$ and the hat has been dropped for convenience. In the case of a cylindrical annular duct with constant radius, the pressure distribution is that of a linear combination of Bessel Functions of both the first and second kind in the radial direction:

$$p\left(r\right) = aJ_m\left(\mu r\right) + bY_m\left(\mu r\right), \tag{2.14}$$

where J_m and Y_m are Bessel functions of the first kind and second kind respectively. This gives an acoustic mode in a uniform mean flow through an annular duct of uniform geometry the general form:

$$\left(aJ_m\left(\mu r\right) + bY_m\left(\mu r\right)\right) e^{i\omega t - ikx - im\theta} \tag{2.15}$$

2.2 Eigenanalysis

In developing Boundary conditions it is important to find the eigensolutions of the problem as this tells us the direction in which the information travels as well as the scattering of harmonics that occurs due to non-uniform flows and non-uniform geometries. The difficulty of this task depends on the dimension of the problem, the uniformity of the duct, and the type of mean flow. Therefore this section will give a review of eigenanalysis for use in analysis of perturbation effects in ducted flows. The problem increases in difficulty when mean flow is introduced. The simplest non-trivial flows that still allow for modal expansion are uniform mean flows. The modal solutions, in this case, are qualitatively similar to those without flow. Flows that vary spatially introduce non-constant coefficients into the acoustic equations. The problem also becomes more difficult in ducts of varying geometry as non-uniform geometries causes a scattering of harmonics into neighbouring harmonics as they travel through the duct. The difficulty in this case is to obtain a method that captures the effects of non-uniform geometries on harmonics throughout the entire duct. Eigenanalysis requires, as an input, the unsteady linearised Euler equations, the frequency, mode number, and the underlying mean flow upon which the perturbations will travel. This method has been used to analyse acoustical vibrations in a finite domain [33, 34] and can also be used to study the P-stability of finite difference approximations to scalar equations [35, 36]. P-Stability being the stability of an initial boundary value problem solved computationally on a finite mesh.

In the case of uniform flow, the problem involves finding values for the wave number k. In this description physically stable modes will be assumed. The wave numbers belong to four categories of modes: upstream and downstream propagating acoustic modes and downstream propagating vorticity and entropy modes. In order to distinguish upstream acoustical modes from the downstream acoustical modes we look first at the sign of the imaginary part of the wave number k. The exponential part of the assumed solution can be rewritten as,

$$e^{i\omega t+im\theta+ik_r x}e^{-k_i x} \tag{2.16}$$

where k_i is the imaginary part and k_r is the real part. If the imaginary part satisfies $k_i > 0$ then this corresponds to an evanescent mode propagating downstream, in the positive x direction. This is confirmed by the previous assumption of physically stable modes which allows one to simply analyse the “direction of decay”. Likewise if $k_i < 0$ then what is present is an evanescent mode propagating upstream.

For real k that is, $k_i = 0$, we introduce the group velocity [37]:

$$\mathbf{u}_g = \begin{pmatrix} \frac{\partial \omega}{\partial k} \\ \frac{\partial \omega}{\partial m} \end{pmatrix} \tag{2.17}$$

Incoming waves, that are waves moving downstream or the positive x axis, are then defined as those that have $\partial\omega\backslash\partial k > 0$. The acoustic modes are those for which the flow is irrotational and isentropic. Vortical waves are characterised by a divergence free unsteady velocity profile along with zero pressure and density perturbation while entropy modes are usually characterised by the areas of lower or higher density transported with the flow. Note that this is true only for uniform flows and flows close to uniform.

2.2.1 1 Dimensional Eigenanalysis

Giles [33] uses eigenanalysis as a means of studying the low frequency waves that travel up and down the domain during quasi-one-dimensional or two dimensional finite difference calculations that use time marching methods. These frequencies are the remaining residuals before the solutions converges to a steady state. Starting with a simple one dimensional example, Giles [33], derives the eigenvalues and eigenvectors for the linearised Euler equations in one dimension. The problem here boasts constant coefficients. It is therefore possible to derive the exact eigenmodes and eigenfrequencies of the problem. The one dimensional non-dimensional linearised Euler equation takes the form:

$$\begin{pmatrix} \rho \\ u \\ p \end{pmatrix}_t + \begin{pmatrix} M & 1 & 0 \\ 0 & M & 1 \\ 0 & 1 & M \end{pmatrix} \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}_x = 0 \tag{2.18}$$

The variables here have been normalised as follows:

$$\rho = \tilde{\rho}/\bar{\rho} \tag{2.19}$$

$$u = \tilde{u}/\bar{c} \tag{2.20}$$

$$p = \tilde{p}/\bar{\rho}\bar{c}^2 \tag{2.21}$$

In the non-dimensional domain the subsonic inflow is at $x = 0$ and the outflow at $x = 1$. As seen previously, wave like solutions of the form:

$$\begin{pmatrix} \rho \\ u \\ p \end{pmatrix} = e^{ikx - i\omega t} U \tag{2.22}$$

are assumed given that:

$$(kA - \omega I)U = 0 \tag{2.23}$$

where U are the eigenvectors and $\lambda = \omega/k$ the corresponding eigenvalues.

$$\lambda_1 = \quad M, \quad U_1 \quad = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{2.24}$$

$$\lambda_2 = \quad M + 1, \quad U_2 \quad = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{2.25}$$

$$\lambda_3 = \quad M - 1, \quad U_3 \quad = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} \tag{2.26}$$

From this it can be seen there are two downstream travelling waves and one upstream travelling wave. With $\lambda_n = \omega/k_n$ it is possible to rearrange the above equations to find equations for ω in terms of k_n. Differentiating it is found that, for subsonic flows, $\partial\omega\backslash\partial k_1 > 0$ and $\partial\omega\backslash\partial k_2 > 0$. Writing the sum of the three eigenmodes gives a general eigenmode solution:

$$U = e^{i\omega t} \sum_{n=3}^{n=1} \alpha_n e^{ik_n x} U_n \tag{2.27}$$

From here we can formulate boundary conditions. At the inflow boundary, $x = 0$, there are two boundary conditions:

$$C_{in} U = 0 \tag{2.28}$$

where C_{in} is a 2×3 matrix. Substituting (2.27) into the boundary condition gives:

$$\begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = 0 \tag{2.29}$$

where

$$\begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix} = C_{in} \begin{pmatrix} U_1 & U_2 & U_3 \end{pmatrix} \tag{2.30}$$

At the outflow, $x = 1$, there is a single boundary condition:

$$\begin{pmatrix} b_{31} & b_{32} & b_{33} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = 0 \tag{2.31}$$

where

$$\begin{pmatrix} b_{31} & b_{32} & b_{33} \end{pmatrix} = C_{out} \begin{pmatrix} e^{i(\omega/\lambda_1)}U_1 & e^{i(\omega/\lambda_2)}U_2 & e^{i(\omega/\lambda_3)}U_3 \end{pmatrix} \tag{2.32}$$

A requirement of NRBC's is to find the direction of information travel of the physical waves allowing us to identify spurious modes and set them to zero. In this case downstream travelling waves will be left to leave the domain whilst the reflected modes are what will be set to zero whilst in the case of upstream travelling waves any waves travelling downstream will be set to zero. At the inlet boundary condition it is required to set both the pressure and the velocities whilst at the outlet boundary only the pressure is required as can be seen by (2.29) and (2.31)

2.2.2 2 Dimensional Eigenanalysis

Once again using Giles [18], a description of the two dimensional problem is laid out. The non-dimensional Euler equations in two dimensions are:

$$\boldsymbol{U}_t + A\boldsymbol{U}_x + B\boldsymbol{U}_y = 0, \tag{2.33}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ and $\boldsymbol{U}$ are written as,

$$\boldsymbol{U} = \begin{pmatrix} \rho \\ u \\ v \\ p \end{pmatrix}, \qquad \boldsymbol{A} = \begin{pmatrix} M_u & 1 & 0 & 0 \\ 0 & M_u & 0 & 1 \\ 0 & 0 & M_u & 0 \\ 0 & 1 & 0 & M_u \end{pmatrix} \qquad \boldsymbol{B} = \begin{pmatrix} M_v & 0 & 1 & 0 \\ 0 & M_v & 0 & 0 \\ 0 & 0 & M_v & 1 \\ 0 & 0 & 1 & M_v \end{pmatrix}. \tag{2.34}$$

Here ρ is the perturbed density, u and v the perturbed velocity components and p the perturbed pressure, all normalised. It is also important to mention that $\boldsymbol{A}$ and $\boldsymbol{B}$ are constant $N \times N$ matrices, where M_u and M_v are the Mach numbers in their respective directions[1].

The resulting work of the eigenalysis will give the eigenvalues of k which will give information on the direction of travel of the input signal. Assuming the solution is periodic in y a Fourier transform is performed in the y direction:

$$\boldsymbol{U}\,(x, y, t) = \sum_{n=-\infty}^{\infty} \boldsymbol{u}_n e^{i(k_n x + l_n y - \omega_n t)} \tag{2.35}$$

Choosing a single mode (without loss of generality) and substituting it into (2.33),

$$\left(-\omega \boldsymbol{I} + k\boldsymbol{A} + l\boldsymbol{B}\right) \boldsymbol{u} = 0 \tag{2.36}$$

is obtained, which naturally leads to the dispersion relation,

$$\det\left(-\omega \boldsymbol{I} + k\boldsymbol{A} + l\boldsymbol{B}\right) = 0 \tag{2.37}$$

Of course non-trivial solutions of $\boldsymbol{u}_n$ are of interest. The dispersion relation is an Nth degree polynomial in ω, k, and l. ω and l are defined to be real while k may be complex.

Incoming and outgoing modes need to be distinguished. In order for that to be possible eigenvectors whose dependency is on k are of interest and so one must rewrite the matrix in (2.36) as,

$$-\omega \boldsymbol{A}^{-1} + k\boldsymbol{I} + l\boldsymbol{A}^{-1}\boldsymbol{B}. \tag{2.38}$$

[1] For information on how to arrive at the non dimensionalised linear Euler equations please refer to pages 21-22 of [18]

This is the matrix that shall be used when finding the relevant eigenvectors. The dispersion relation then becomes:

$$\det\left(-\omega \boldsymbol{A}^{-1} + k\boldsymbol{I} + l\boldsymbol{A}^{-1}\boldsymbol{B}\right) = 0 \tag{2.39}$$

Using (2.39) to find values of k, the most general form of a single mode, while holding l and ω constant, is

$$\boldsymbol{U}\left(x, y, t\right) = \left(\sum_{j=1}^{N} a_j \boldsymbol{u}_j^R e^{i(k_j x)}\right) e^{i(ly-\omega t)} \tag{2.40}$$

where the R denotes the right eigenvector of $\left(-\omega \boldsymbol{A}^{-1} + l\boldsymbol{A}^{-1}\boldsymbol{B}\right)$. In 2 dimensions the roots to the dispersion relation, k_j, are

$$k_{1,2} = \frac{\omega - vl}{u}, \qquad k_{3,4} = \frac{(\omega - vl)\,(-u \pm S)}{1-u^2} \tag{2.41}$$

where

$$S = \sqrt{\frac{1-(1-u^2)l^2}{(w-vl)^2}} \tag{2.42}$$

In subsonic conditions $k_{1,2,3}$ are all downstream travelling waves while k_4 is an upstream travelling wave. With this information all that is required is to set all a_j to 0 at the boundary for all j corresponding to incoming waves. An equivalent boundary condition would be,

$$\boldsymbol{v}_i^L \boldsymbol{U} = 0 \tag{2.43}$$

where $\boldsymbol{v}_i^L$ is the left eigenvector of $\left(-\omega \boldsymbol{A}^{-1} + l\boldsymbol{A}^{-1}\boldsymbol{B}\right)$. Recall that left and right eigenvectors of the same matrix are orthogonal to each other except for eigenvectors representing the same eigenvalue, hence the above boundary condition. $\boldsymbol{v}_i^L$ can be calculated via the relationship,

$$\boldsymbol{v}_i^L = \left.\frac{k_i}{\omega}\right|_{l=0} \boldsymbol{u}_i^L \boldsymbol{A} \tag{2.44}$$

where $\boldsymbol{u}_i^L$ is the left eigenvector of $k\boldsymbol{A} + l\boldsymbol{B}$. The relationship comes from considering $\boldsymbol{v}_i^L \boldsymbol{A}^{-1}$ also as an eigenvector of $k\boldsymbol{A} + l\boldsymbol{B}$. Doing so and comparing, it can be seen that

$\boldsymbol{v}_i^L$ is equal to $\boldsymbol{u}_i^L\boldsymbol{A}$ up to some multiple. The right eigenvectors can be calculated as,

$$\begin{aligned}
\boldsymbol{u}_1^R &= \begin{pmatrix}-1\\0\\0\\0\end{pmatrix}, \qquad \boldsymbol{u}_2^R = \begin{pmatrix}0\\-ul/\omega\\uk_2/\omega\\0\end{pmatrix}\\
\boldsymbol{u}_3^R &= \tfrac{1+u}{2\omega}\begin{pmatrix}\omega-uk_3-vl\\k_3\\l\\\omega-uk_3-vl\end{pmatrix}, \qquad \boldsymbol{u}_4^R = \tfrac{1-u}{2\omega}\begin{pmatrix}\omega-uk_4-vl\\k_4\\l\\\omega-uk_4-vl\end{pmatrix}
\end{aligned} \tag{2.45}$$

With the corresponding left eigenvectors being

$$\begin{aligned}
\boldsymbol{v}_1^L &= \Big(-1,\quad 0,\quad 0,\quad 1\Big),\\
\boldsymbol{v}_2^L &= \Big(0,\quad -ul/\omega,\quad 1-vl/\omega,\quad -l/\omega\Big),\\
\boldsymbol{v}_3^L &= \frac{1}{\omega}\Big(0,\quad \omega-vl,\quad ul,\quad u\left(\omega-vl\right)+\left(1-u^2\right)k_3\Big),\\
\boldsymbol{v}_4^L &= -\frac{1}{\omega}\Big(0,\quad \omega-vl,\quad ul,\quad u\left(\omega-vl\right)+\left(1-u^2\right)k_4\Big).
\end{aligned} \tag{2.46}$$

2.2.3 3 dimensional Eigenanalysis

Following the previous methods in three dimensions. In three dimensions the Euler equation takes the form:

$$\boldsymbol{U}_t + A\boldsymbol{U}_x + B\boldsymbol{U}_y + C\boldsymbol{U}_z = 0 \tag{2.47}$$

where the solution in this case is given to be,

$$\boldsymbol{U}\left(x,y,z,t\right) = \sum_{n=-\infty}^{\infty} \hat{\boldsymbol{U}}_n e^{i\left(-\omega_n t + k_n x + l_n y + m_n z\right)} \tag{2.48}$$

where a Fourier transform has been taken. In 3 dimensions the S present in the 2 dimensional case becomes,

$$S = \sqrt{1 - \frac{\left(1-u^2\right)\left(l^2+m^2\right)}{-\omega + lv + mw}} \tag{2.49}$$

Where m is the wave number associated to the z direction and w is the normalised flow velocity in the z direction. Labelling the denominator above as $\Lambda = -\omega + lv + mw$ the roots k_i are calculated as[2],

$$k_{1,2,3} = -\frac{\Lambda}{u}, \qquad k_4 = -\frac{\Lambda\,(S-u)}{1-u^2}, \qquad k_5 = \frac{\Lambda\,(S+u)}{1-u^2} \tag{2.50}$$

Disregarding the third dimension the above equations return to the form present in the 2 dimensional case. The boundary conditions has an equivalent form to that present in (2.43),

$$\boldsymbol{v}_i^L \hat{\boldsymbol{U}}_\ell = 0 \tag{2.51}$$

with $\boldsymbol{U}$ being a five vector rather than a four vector. The left eigenvectors in the 3 dimensional case needed to satisfy the above equation are calculated as,

$$\begin{aligned}
\boldsymbol{v}_1^L &= \Big(-1, \;\; 0, \;\; 0, \;\; 0, \;\; 1\Big), \\
\boldsymbol{v}_2^L &= \frac{1}{\omega}\Big(0, \;\; -ul, \;\; -\Lambda, \;\; 0, \;\; -l\Big), \\
\boldsymbol{v}_3^L &= \frac{1}{\omega}\Big(0, \;\; -um, \;\; 0, \;\; -\Lambda, \;\; -m\Big), \\
\boldsymbol{v}_4^L &= \frac{1}{\omega}\Big(0, \;\; -\Lambda, \;\; ul, \;\; um, \;\; \Lambda S\Big), \\
\boldsymbol{v}_5^L &= \frac{1}{\omega}\Big(0, \;\; -\Lambda, \;\; ul, \;\; um, \;\; -\Lambda S\Big).
\end{aligned} \tag{2.52}$$

2.2.4 Eigenanalysis in Cylindrical Polar Coordinates

Staying in three dimensions the eigenanalysis method is widely used for turbo-machinery flows and is explained by Giles et al [20]. The flow example here is considered to be axially and circumferentially uniform, as a result the numerical method approximates the 3 dimensional eigenmodes as eigenvectors of a general eigenvalue problem. Whilst turbo-machinery flows are not uniform in either the axial or circumferential direction, the assumption works well in many cases, particularly at the inflow and outflow. As mentioned the direction of propagation is obtained from the imaginary part of the axial wave number which are the eigenvalues of the general eigenvalue problem. In the subsonic case, the modes propagating consist of acoustic modes going up and downstream, vortical modes going downstream and entropy modes going downstream. Relevant to the scope of work present here, eigenanalysis in 3 dimensional cylindrical polar coordinates is considered before analysing a hard

[2] The normalisation has been kept consistent with Giles [18]

walled annular duct example. The Euler equations in this setting take the form:

$$\frac{\partial \mathbf{Q}_c}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}\left(r\mathbf{F}_r\right) + \frac{1}{r}\frac{\partial \mathbf{F}_\theta}{\partial \theta} + \frac{\partial \mathbf{F}_x}{\partial x} = \mathbf{G} \tag{2.53}$$

Here, $\mathbf{Q}_c$ represents the conservative variables, whilst $\mathbf{F}_r$, $\mathbf{F}_\theta$, $\mathbf{F}_x$ represent the fluxes in their respective direction denoted by their subscript. $\mathbf{G}$ represents a source term. Linearising these equations gives,

$$M\frac{\partial \mathbf{Q}_p}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}\left(rA_r\mathbf{Q}_p\right) + \frac{1}{r}A_\theta\frac{\partial \mathbf{Q}_p}{\partial \theta} + A_x\frac{\partial \mathbf{Q}_p}{\partial x} = S\mathbf{Q}_p \tag{2.54}$$

where $M = \frac{\partial \mathbf{Q}_c}{\partial \mathbf{Q}_p}$ and $\mathbf{Q}_p$ is the vector of pertubation variables akin to $\boldsymbol{U}$ in the previous section. The matrices above, M, A_r, A_θ, A_x, S are all functions of radius only. This means the eigenmodes are given in the form of a separable solution in t, θ and x, namely,

$$\mathbf{Q}_p\left(t, x, \theta, r\right) = e^{i\omega t + im\theta + ikx}\mathbf{Q}_p\left(r\right) \tag{2.55}$$

Substituting (2.55) in to (2.54) gives,

$$i\omega M\mathbf{Q}_p + \frac{1}{r}\frac{\partial}{\partial r}\left(rA_r\mathbf{Q}_p\right) + \frac{1}{r}imA_\theta\mathbf{Q}_p + ikA_x\mathbf{Q}_p = S\mathbf{Q}_p \tag{2.56}$$

In dealing with numerical calculation descritisation is necessary. Rearranging the above and then discretising, leads to the algebraic equation,

$$\left(i\omega\widehat{M} + \widehat{A}_r + im\widehat{A}_\theta + ik\widehat{A}_x - \widehat{S}\right)\mathbf{Q} = 0 \tag{2.57}$$

Here the hatted variables are square matrices, now of dimension $5N$ and $\mathbf{Q}$ is a vector of length $5N$ where N is the number of radial points.

2.2.5 Hard Walled Annular Duct Example

Here a uniform inviscid example will be considered and the equations of motion are non-dimensionalised using the speed of sound c and the outer annulus radius. In this example (2.11) is simplified and the linear pressure perturbations satisfy the equation,

$$\left(\frac{\partial}{\partial t} + M\frac{\partial}{\partial x}\right)^2 p = \boldsymbol{\nabla}^2 p, \quad \lambda < r < 1, \tag{2.58}$$

where M is the Mach number. Before going further it is necessary to note that the boundary conditions to be applied to the inner and outer walls of the annular duct are the hard wall boundary conditions, i.e. radial velocity is zero at the walls. This induces a boundary condition on the pressure at the walls,

$$\frac{\partial p}{\partial r} = 0\bigg|_{r=\lambda,1} \tag{2.59}$$

In the case of uniform flow the eigenmodes take the form

$$p\left(t,x,\theta,r\right) = e^{i\omega t+im\theta+ikx}p\left(r\right) \tag{2.60}$$

Taking the usual steps and substituting this into (2.58) leads to the Bessel equation

$$\frac{1}{r}\frac{\mathrm{d}}{\mathrm{d}r}\left(r\frac{\mathrm{d}p}{\mathrm{d}r}\right) + \left(\mu^2 - \frac{m^2}{r^2}\right)p = 0, \quad \lambda < r < 1, \tag{2.61}$$

where

$$\mu^2 = \left(\tilde{\omega} + Mk\right)^2 - k^2 \tag{2.62}$$

Where $\tilde{\omega} = \omega/\bar{\bar{c}}$. The general solution to (2.61) is a combination of Bessel functions of the first and second kind namely,

$$p\left(r\right) = aJ_m\left(\mu r\right) + bY_m\left(\mu r\right). \tag{2.63}$$

J_m is the Bessel function of the first kind while Y_m is the Bessel function of the second kind. The earlier boundary conditions to be implemented gives,

$$\begin{pmatrix} J'_m\left(\mu\lambda\right) & Y'_m\left(\mu\lambda\right) \\ & \\ J'_m\left(\mu\right) & Y'_m\left(\mu\right) \end{pmatrix} \begin{pmatrix} a \\ \\ b \end{pmatrix} = 0 \tag{2.64}$$

For the above to be true and non-trivial, the determinant of the above matrix has to be zero. This allows for values of μ to be found therefore also allowing values of k to be found using (2.62), allowing for the determination of the direction of propagation of the eigenmodes.

It is important to note the value of μ in the case of a two dimensional duct with constant radius R. This can be derived directly from the Bessel equation. As there is no radial contribution as radius is now constant, the Bessel equation reduces to:

$$\left((\tilde{\omega} + Mk)^2 - k^2 - \frac{m^2}{R^2}\right) p(R) = 0 \tag{2.65}$$

leading to:

$$(\tilde{\omega} + Mk)^2 - k^2 = \frac{m^2}{R^2} \quad \rightarrow \quad \mu^2 = \frac{m^2}{R^2} \tag{2.66}$$

The need for this will be discussed further in section 3.2.1 where implementation of the method in two dimensions is discussed.

2.2.6 Eigenanalysis For Non-Uniform Flow

This is important as a non-uniform flow as well as a non-uniform geometry can scatter the modal content of acoustic phenomena. As a result the eigenanalysis techniques need to be able to capture these effects in order to give accurate boundary conditions.

As an example, consider a cylindrical duct with a Gaussian mean flow deficit of the form

$$\bar{u}_x = 0.5 - 0.3e^{-\frac{\theta^2}{2\sigma^2}} \tag{2.67}$$

This is one of the test cases used in chapter 4 and the mean flow profile is shown in figure 4.1. The effect of the Gaussian flow deficit on the pressure profile at the outer wall for the mode most similar to the first radial mode at $m = 5$ in uniform flow is shown in figures 2.1 and 2.2. The deformation of the harmonic is greater for upstream running waves as these waves go against the flow. This gives the Gaussian more time to affect the harmonic, spreading and deforming it further.

Figure 2.3 shows a plot of the amplitude of the modal coefficient for a range of acoustic modes. Each mode here is normalised to unity maximum pressure level. From the figure, it can be observed that in non-uniform flow, reflections are spread into multiple modes and the highest amplitude reflected mode is not necessarily the mode with highest harmonic content at $m = 5$ whereas in uniform flow the reflection is dominated by a single mode.

Looking now at the theory developed by Kousen [32]. His worked revolved around, like Giles, writing the equations of motion in the form of an eigen problem. The flow is taken to be isentropic and axisymmetric and he examined three special cases: Axially sheared

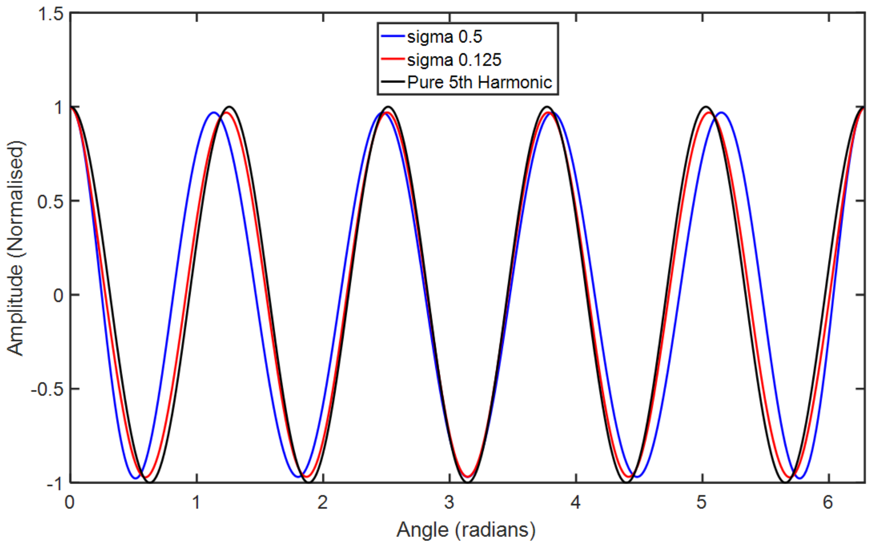


Figure 2.1: Deformation of the first radial harmonic downstream wave at $m = 5$ due to a non-uniform mean flow profile. The figure shows the circumferential pressure amplitude at the outer wall.

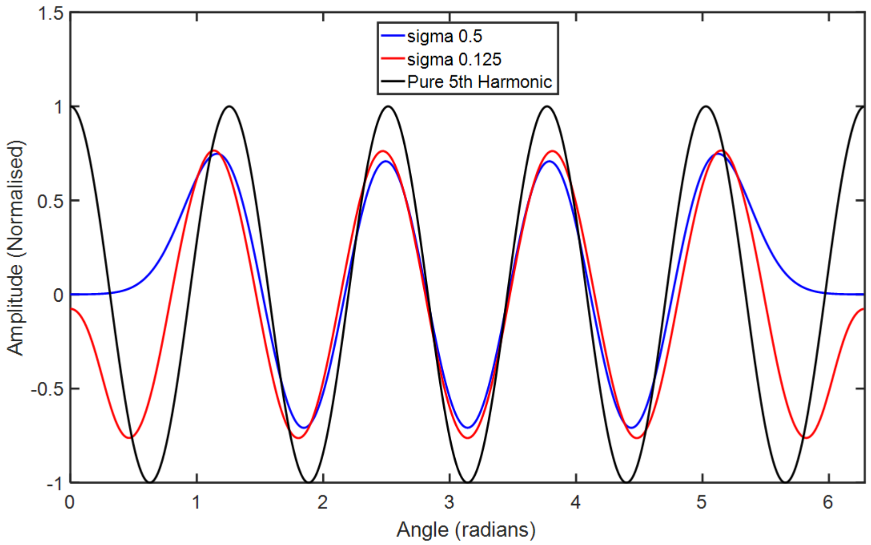


Figure 2.2: Deformation of the first radial harmonic upstream wave at $m = 5$ due to a non-uniform mean flow profile. The figure shows the circumferential pressure amplitude at the outer wall.

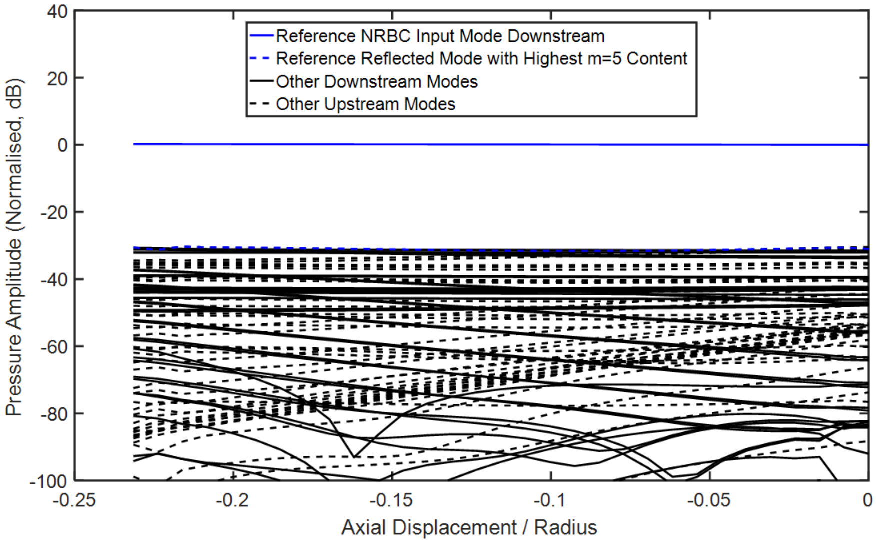


Figure 2.3: An example of reflected modes for non-uniform mean flow ($\sigma = 0.5$).

flow, solid body swirl and flows with free vortex swirl. In cases with shear flow, mean pressure, mean density and the speed of sound are constant across the duct which is no longer the case in flows with a swirl component. The normalised speed of sound in the cases of solid body swirl, free vortex swirl and the two combined is different for each case and is given in [32]. Also in [32] are the equations of motion for axial shear flow in cylindrical coordinates. Assuming that the perturbation quantities have exponential dependence,

$$f\left(t,x,\theta,r\right) = e^{i\omega t + im\theta + ikx} f\left(r\right) \tag{2.68}$$

where $f = p, v_r, v_\theta, v_x$, gives the equation of motion for pressure:

$$\frac{\mathrm{d}^2 p}{\mathrm{d}r^2} + \left(\frac{1}{r} - \frac{2M_x\bar{\kappa}}{(k - 2M_x'\bar{\kappa})}\right)\frac{\mathrm{d}p}{\mathrm{d}r} + \left((k - 2M_x'\bar{\kappa})^2 - \frac{m^2}{r^2} - \bar{\kappa}^2\right)p = 0 \tag{2.69}$$

where M_x and $\bar{\kappa}$ are the axial Mach number and the dimensionless axial wave number respectively. This second order differential equation found describing axial shear flows is not adequate, when solved, at determining the radial behaviour of the unsteady quantities as it misses entire families of perturbations. Using Goldstein [38] it is possible to derive a single equation for pressure by combining the linearised momentum and continuity equations by taking the divergence of the momentum equation and subtracting away from that the material derivative of the continuity equation. The material derivative is then applied to the resulting equation giving:

$$\mathrm{D}_0^3\, p - c_0^2\, \mathrm{D}_0\, \nabla^2 p + 2c_0^2 \frac{\partial u_0}{\partial r} \frac{\partial^2 p}{\partial x \partial r} = 0 \tag{2.70}$$

This equation can also be derived directly from (2.11) by assuming c_0 is a constant and that $u_0 = u_x(r)$ which is true for axial shear flow. Assuming solutions of the form (2.68) it is possible to obtain

$$\bar{\lambda} L\left[p\right] = 0 \tag{2.71}$$

where $\bar{\lambda} = k - \bar{\gamma} M_x$ and $L\left[p\right]$ represents the second order ODE for pressure. There are therefore 3 sets of solutions. The first being $L\left[p(r)\right] = 0$ which gives rise to a discrete set of eigenvalues. The second is $\bar{\lambda}(r) = 0$ everywhere across the duct giving rise to a continuum of disturbances and finally $L\left[p(r)\right] = \delta(\bar{\lambda} = 0)$ which gives rise to a continuum of eigenvalues. The second set of solutions represent wavenumbers that are analogous to discrete acoustic disturbances present in uniform flow.

For flows with axial shear and full body swirl it is not possible, using the same methods as with axial shear flows, to achieve a single equation for pressure, independent of the velocity perturbations. Instead the exponential assumption in (2.68) is plugged directly into the linearised perturbation equations for swirling flows. The equations can be found in [32]. Three expressions are found relating the pressure to each direction of velocity and the ODE for pressure is found. It can then be written in matrix form as:

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{B}\mathbf{x} \tag{2.72}$$

where

$$\mathbf{A} = \begin{pmatrix} -i\left(\frac{k}{c_0} - \frac{m}{r} M_\theta\right) & -\frac{2}{r} M_\theta & 0 & \frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\gamma - 1)}{r} M_\theta^2 \\ \frac{1}{r} M_\theta + \frac{\mathrm{d}M_\theta}{\mathrm{d}r} + \frac{(\gamma-1)}{2r} M_\theta^3 & -i\left(\frac{k}{c_0} - \frac{m}{r} M_\theta\right) & 0 & i\frac{m}{r} \\ \frac{\mathrm{d}M_x}{\mathrm{d}r} + \frac{(\gamma-1)}{2r} M_x M_\theta^2 & 0 & -i\left(\frac{k}{c_0} - \frac{m}{r} M_\theta\right) & 0 \\ \frac{\mathrm{d}}{\mathrm{d}r} + \frac{1}{r} + \frac{(\gamma-1)}{2r} M_\theta^2 & i\frac{m}{r} & 0 & -i\left(\frac{k}{c_0} - \frac{m}{r} M_\theta\right) \end{pmatrix} \tag{2.73}$$

$$\mathbf{B} = \begin{pmatrix} M_x & 0 & 0 & 0 \\ 0 & M_x & 0 & 0 \\ 0 & 0 & M_x & 1 \\ 0 & 1 & 1 & M_x \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} u_r \\ u_\theta \\ u_x \\ p \end{pmatrix} \tag{2.74}$$

A and **B** are only dependent on the radial coordinate and **B** has an analytical inverse assuming its determinant is non-zero. As a result **B** is singular when $M_x = 0, \pm 1$. If **B** is non-singular, then it is possible to write

$$\mathbf{B}^{-1}\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \tag{2.75}$$

however non-singularity of **B** is not required to solve the eigenvalue problem.

2.2.7 Non-Uniform Ducts and Flows

Wilson [27] developed a method that allows for successful eigenanalysis in non-uniform flows in ducts with non-uniform, varying, geometry meaning that the eigenanalysis captures the changing of the eigenmode due to the changing geometry and mean flow unlike other methods whose eigenanalysis may only capture the shape of the eigenmode at a specific axial point in the duct.

The eigenanalysis method used will be dubbed EigenAnalysis in Generalised Curvilinear Coordinates (EAGCC). As it can be applied in non-uniform geometry with non-uniform mean flow and there is no requirement that the solution to the linearised Euler equations be separable. The required inputs to the method are a duct that is described by a global body fitted coordinate system, a time-averaged flow field and an input acoustic signal or perturbation. This form of modal decomposition is based on the linearised homentropic Euler equations [27]. Writing them in tensor form in a vector space based off contravariant basis vectors:

$$\begin{aligned}
\frac{1}{\bar{\bar{c}}}\frac{\partial u_i'}{\partial t} &+ \frac{1}{2}\frac{\partial}{\partial x^i}\left(g^{jk}\left[u_j'\bar{u}_k + u_k'\bar{u}_j\right]\right) - \epsilon^{ljk}\epsilon^{mnp}g_{il}g_{km}\left[u_j'\frac{\partial \bar{u}_p}{\partial x^n} + \bar{u}_j\frac{\partial u_p'}{\partial x^n}\right] \qquad (2.76)\\
&+ \frac{1}{\bar{\rho}}\frac{\partial p'}{\partial x^i} - \frac{p'}{\gamma\bar{\rho}^2\bar{c}^2}\frac{\partial \bar{p}}{\partial x^i} = 0\\
\frac{1}{\bar{\bar{c}}}\frac{1}{\bar{c}^2}\frac{\partial p'}{\partial t} &+ \frac{p'}{\sqrt{g}\bar{c}^2}\frac{\partial}{\partial x^i}\left(\sqrt{g}g^{ij}\bar{u}_j\right) + g^{ij}\bar{u}_j\frac{\partial}{\partial x^i}\left(\frac{p'}{\bar{c}^2}\right) + \frac{\bar{\rho}}{\sqrt{g}}\frac{\partial}{\partial x^i}\left(\sqrt{g}g^{ij}u_j'\right) \qquad (2.77)\\
&+ g^{ij}u_j'\frac{\partial \bar{\rho}}{\partial x^i} = 0
\end{aligned}$$

where $\bar{c}$ is the local correction to the speed of sound $\bar{\bar{c}}$. Here the variables are non-dimensionalised, with the velocities u' and $\bar{u}$ being normalised by $\bar{\bar{c}}$, the pressure p' normalised by $\gamma\bar{\bar{p}}$ and the density $\bar{\rho}$ by $\bar{\bar{\rho}}$. These are formulated using the appendix present in [27]. g_{ij} is the metric tensor with inverse g^{ij} and determinant g and ϵ^{ijk} is the permutation tensor.

Wilson [27] reforms equations (2.76) and (2.77) into a generalised eigenvalue problem:

$$A\frac{\partial \hat{f}}{\partial \zeta} = B\hat{f}. \qquad (2.78)$$

This equation is written in tensor form using contravariant basis vectors ($\hat{\boldsymbol{\xi}}$, $\hat{\boldsymbol{\eta}}$, $\hat{\boldsymbol{\zeta}}$) related to the curvilinear coordinate system compatible with the body-fitted mesh such that the ξ, η, ζ coordinates represent the pseudo-radial, pseudo-circumferential and pseudo-axial directions respectively.

In the general case the flow vector $\hat{f}$ is a long vector consisting of each velocity component and the pressure at every node on the boundary surface. In the cases calculated in this thesis a spectral method is used in the pseudo-circumferential direction which only included circumferential harmonics relevant to the problem at hand. This spectral approach greatly reduces the dimensionality of the problem and improves the efficiency of the method.

The process for determining the A and B matrices is conceptually simple. All of the linear terms in equation (2.76) and (2.77) are calculated at each mesh point. Terms containing the ζ derivative of a perturbation variable are placed in matrix A and all the other terms are placed in matrix B. Wilson [27] provides an example of this for one of the terms.

The A and B matrices are adjusted at the inner and outer walls to include a hardwall boundary condition. Because the contravariant basis vector $\hat{\boldsymbol{\xi}}$ is normal to the wall, the hardwall boundary condition can be written simply as:

$$\hat{\boldsymbol{\xi}} \cdot \boldsymbol{u} \quad \text{or} \quad g^{1j}u_j = 0 \quad \text{in tensor form} \tag{2.79}$$

The Jacobian for this coordinate system has to be calculated both to quantify the metric tensor terms in (2.76) and (2.77) and to convert physical to and from tensor form. It can be calculated before hand directly from the definition of the calculation mesh however this requires the mesh to be of high enough resolution so as to not introduce errors. In Wilson [27] the Jacobian and metric tensor were calculated using linear partial differential operators in each direction. Differentiation in the ξ direction is represented by a matrix. The ith row of the matrix then represents the coefficients needed to provide a fourth order polynomial approximation of the derivative at the ith point using a five point stencil which is one sided at the walls. As for the pseudo-circumferential direction a pseudo-spectral solution was used. The matrices are expanded to be able to work on full $n_\xi \times n_\eta$ long vectors representing the unknowns at every point on the surface.

Recall that the mean flow here is required to be time-averaged and the flow perturbation should be at fixed frequency. Eigensolutions are generated along surfaces of constant ζ. In the present application, equation (2.78) is only required at the inlet and outlet flow boundaries, but the full body-fitted mesh is still needed to generate the A and B matrices as they encapsulate geometric information about the duct and mean flow, including the local hade of the inner and outer walls.

The method offers freedom from the slowly varying limit required in the method of multiple scales which will be discussed in the upcoming sections. This means with the exception of extremely sharp, essentially discontinuous, changes in geometry, the duct geometry can vary strongly with position.

In his later papers Wilson extended his work for use in external flows in the near field of the inlet [39] and for use in predicting non-linear acoustic perturbations such as rotor alone noise or buzz saw noise. In this latter case Wilson added a second order correction to account for non-linear effects during propagation [40].

2.3 Non-Reflecting Boundary Conditions

Much of the literature takes inspiration from Giles [18]. In this paper Giles assumes uniform axial flow, in which case, the solutions take on a wave-like nature of the form similar to:

$$\boldsymbol{u}\left(x,t\right)=\sum_{k,\omega}\boldsymbol{U}_{\mathbf{k},\omega}e^{i(\mathbf{k}\cdot\mathbf{x}-\omega t)} \tag{2.80}$$

which are all assumed to be physically stable modes. The eigenanalysis described in sections 2.2.1 to 2.2.5 is used here to develop exact two and three dimensional EBNRBC's with uniform mean flows and geometries and approximate EBNRBC's for two dimensional non uniform mean flows and geometries with preliminary analytical work done in three dimensions. Using this theory Giles discussed the development of many types of boundary conditions:

- NRBC's for a single Fourier mode.
- One dimensional unsteady NRBC's.
- Exact 2 dimensional, fixed frequency NRBC's.
- Exact, 2 dimensional, steady NRBC's.
- Approximate, 2 dimensional, unsteady NRBC's.

Once the direction of propagation had been determined any waves that did not contribute to the correct direction were set to zero.

2.3.1 One Dimensional Unsteady Boundary Conditions

The first example presented is the one dimensional unsteady case. The computational domain is defined as $0 < x < 1$ with the boundary at 0 and 1 being the inflow and outflow boundaries respectively. In this case $\lambda = 0$ and the right and left eigenvectors, denoted by $\boldsymbol{w}_i$ are

$$\boldsymbol{w}_1^R=\begin{pmatrix}-1\\0\\0\\0\end{pmatrix},\qquad \boldsymbol{w}_2^R=\begin{pmatrix}0\\0\\1\\0\end{pmatrix},\qquad \boldsymbol{w}_3^R=\begin{pmatrix}\frac{1}{2}\\\frac{1}{2}\\0\\\frac{1}{2}\end{pmatrix},\qquad \boldsymbol{w}_4^R=\begin{pmatrix}\frac{1}{2}\\-\frac{1}{2}\\0\\\frac{1}{2}\end{pmatrix} \tag{2.81}$$

and

$$\begin{aligned}\boldsymbol{w}_1^L &= \begin{pmatrix}-1, & 0, & 0, & 1\end{pmatrix}\\ \boldsymbol{w}_2^L &= \begin{pmatrix}0, & 0, & 1, & 0\end{pmatrix}\\ \boldsymbol{w}_3^L &= \begin{pmatrix}0, & 1, & 0, & 1\end{pmatrix}\\ \boldsymbol{w}_4^L &= \begin{pmatrix}0, & -1, & 0, & 1\end{pmatrix}\end{aligned} \tag{2.82}$$

Assuming correct normalisation between left and right eigenvectors, such that $\boldsymbol{w}_m^L \boldsymbol{w}_n^R = \delta_{mn}$, one can define a transformation to and from the 1 dimensional characteristic variables,

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{\rho} \\ \tilde{u} \\ \tilde{v} \\ \tilde{p} \end{pmatrix} \tag{2.83}$$

$$\begin{pmatrix} \tilde{\rho} \\ \tilde{u} \\ \tilde{v} \\ \tilde{p} \end{pmatrix} = \begin{pmatrix} -1 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} \tag{2.84}$$

In terms of characteristic variables the boundary conditions are then,

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = 0 \tag{2.85}$$

at $x = 0$ and

$$c_4 = 0 \tag{2.86}$$

at $x = 1$

2.3.2 2 Dimensional Boundary Conditions

Moving on to the 2 dimensional subsonic examples and examining the steady 2 dimensional case, it is seen that exact boundary conditions can be formulated for simple geometries and when $\boldsymbol{U}$ is periodic in y with a period of 2π at the boundary. In that case $\boldsymbol{U}$ at $x = 0$

is written as,

$$\boldsymbol{U}(0,y,t) = \sum_{-\infty}^{\infty} \hat{\boldsymbol{U}}_l(t)\, e^{ily} \tag{2.87}$$

where $\hat{\boldsymbol{U}}_l(t)$ is the Fourier transform of $\boldsymbol{U}$ in the y direction. In the steady case the left eigenvectors need to be found for $\omega \to 0$ and they are found to be, denoting them as $\boldsymbol{s}_i$,

$$\begin{aligned}
\boldsymbol{s}_1^L &= \lim_{\lambda\to\infty} \boldsymbol{v}_1^L = \begin{pmatrix}-1, & 0, & 0, & 1\end{pmatrix} \\
\boldsymbol{s}_2^L &= \lim_{\lambda\to\infty} \frac{1}{\lambda}\boldsymbol{v}_2^L = \begin{pmatrix}0, & -u, & -v, & -1\end{pmatrix} \\
\boldsymbol{s}_3^L &= \lim_{\lambda\to\infty} \frac{1}{\lambda}\boldsymbol{v}_3^L = \begin{pmatrix}0, & -v, & u, & \beta\end{pmatrix} \\
\boldsymbol{s}_4^L &= \lim_{\lambda\to\infty} \frac{1}{\lambda}\boldsymbol{v}_4^L = \begin{pmatrix}0, & v, & -u, & \beta\end{pmatrix}
\end{aligned} \tag{2.88}$$

where

$$\beta = \begin{cases} i\mathrm{sign}(l)\sqrt{1-u^2-v^2}, & u^2+v^2<1, \\ -\mathrm{sign}(l)\sqrt{u^2+v^2-1}, & u^2+v^2>1. \end{cases} \tag{2.89}$$

Fourier transforming $\boldsymbol{U}$ in y as before, the boundary conditions are formulated. At the inflow the boundary conditions are,

$$\begin{pmatrix} -1 & 0 & 0 & 1 \\ 0 & -u & -v & -1 \\ 0 & -v & u & \beta \end{pmatrix} \hat{\boldsymbol{U}}_l(t) = 0 \tag{2.90}$$

and at the outflow they are,

$$\begin{pmatrix}0, & v, & -u, & \beta\end{pmatrix} \hat{\boldsymbol{U}}_l(t) = 0 \tag{2.91}$$

In terms of characteristic variables this reads as,

$$\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -v & -\frac{1}{2}(1+u) & -\frac{1}{2}(1-u) \\ 0 & u & \frac{1}{2}(\beta-v) & \frac{1}{2}(\beta+v) \end{pmatrix} \begin{pmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \hat{c}_3 \\ \hat{c}_4 \end{pmatrix} = 0 \tag{2.92}$$

at the inflow and

$$\begin{pmatrix} 0 & -u & \frac{1}{2}(\beta+v) & \frac{1}{2}(\beta-v) \end{pmatrix} \begin{pmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \hat{c}_3 \\ \hat{c}_4 \end{pmatrix} = 0 \tag{2.93}$$

at the outflow.

Finally there is the most general example, the approximate unsteady case in 2 dimensions. In this case, in order to obtain a second order approximation, $\boldsymbol{v}_i^L$ is expanded about $\lambda = 0$ in a Taylor series. In the limit where $\lambda = 0$, the left eigenvalues, denoted as $\bar{\boldsymbol{v}}_i^L$ are calculated as,

$$\begin{aligned} \bar{\boldsymbol{v}}_1^L &= \begin{pmatrix} -1, & 0, & 0, & 1 \end{pmatrix} \\ \bar{\boldsymbol{v}}_2^L &= \begin{pmatrix} 0, & -u\lambda, & 1-v\lambda, & -\lambda \end{pmatrix} \\ \bar{\boldsymbol{v}}_3^L &= \begin{pmatrix} 0, & 1-v\lambda, & u\lambda, & 1-v\lambda \end{pmatrix} \\ \bar{\boldsymbol{v}}_4^L &= \begin{pmatrix} 0, & -(1-v\lambda), & -u\lambda, & 1-v\lambda \end{pmatrix} \end{aligned} \tag{2.94}$$

Note that the first two eigenvectors are exact. Applying these boundary conditions one gets the inflow boundary condition as,

$$\begin{pmatrix} -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \boldsymbol{U}_t + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & u & v & 1 \\ 0 & v & -u & v \end{pmatrix} \boldsymbol{U}_y = 0 \tag{2.95}$$

and

$$\begin{pmatrix} 0 & -1 & 0 & 1 \end{pmatrix} \boldsymbol{U}_t + \begin{pmatrix} 0 & -v & u & v \end{pmatrix} \boldsymbol{U}_y = 0 \tag{2.96}$$

at the outflow. In terms of characteristics the boundary conditions are

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}_t + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & v & \frac{1}{2}(1+u) & \frac{1}{2}(1-u) \\ 0 & -u & v & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix}_y = 0 \tag{2.97}$$

at the inflow and

$$(c_4)_t + \begin{pmatrix} 0 & u & 0 & v \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix}_y = 0 \tag{2.98}$$

at the outflow.

2.3.3 3 Dimensional unsteady Boundary Conditions

Using the three dimensional eigenvectors:

$$\begin{aligned}
\boldsymbol{v}_1^L &= \begin{pmatrix} -1, & 0, & 0, & 0, & 1 \end{pmatrix}, \\
\boldsymbol{v}_2^L &= \frac{1}{\omega}\begin{pmatrix} 0, & -ul, & -\Lambda, & 0, & -l \end{pmatrix}, \\
\boldsymbol{v}_3^L &= \frac{1}{\omega}\begin{pmatrix} 0, & -um, & 0, & -\Lambda, & -m \end{pmatrix}, \\
\boldsymbol{v}_4^L &= \frac{1}{\omega}\begin{pmatrix} 0, & -\Lambda, & ul, & um, & \Lambda S \end{pmatrix}, \\
\boldsymbol{v}_5^L &= \frac{1}{\omega}\begin{pmatrix} 0, & -\Lambda, & ul, & um, & -\Lambda S \end{pmatrix}.
\end{aligned} \tag{2.99}$$

The three dimensional EBNRBC's are:

$$\begin{pmatrix}
-1 & 0 & 0 & 0 & 1 \\
0, & -\frac{1}{\omega}ul, & -\frac{1}{\omega}\Lambda, & 0, & -\frac{1}{\omega}l \\
0, & -\frac{1}{\omega}um, & 0, & -\frac{1}{\omega}\Lambda, & -\frac{1}{\omega}m \\
0, & -\frac{1}{\omega}\Lambda, & \frac{1}{\omega}ul, & \frac{1}{\omega}um, & \frac{1}{\omega}\Lambda S
\end{pmatrix} \hat{\boldsymbol{U}}_\ell = 0 \tag{2.100}$$

at the inlet and

$$\frac{1}{\omega}\begin{pmatrix} 0, & -\Lambda, & ul, & um, & -\Lambda S \end{pmatrix} \hat{\boldsymbol{U}}_\ell = 0 \tag{2.101}$$

2.3.4 3 Dimensional Steady NRBC's

Petrie-Repar [41] formulates a 3 dimensional form of these boundary conditions, inspired by Giles and tests them for 2 dimensional problems and 3 dimensional problems which will be discussed further below. The extension to three dimensions, as can be seen, involves the addition of an extra dimension in space giving rise to the third direction of velocity. The algebra is more or less the same as the third dimension and does not impinge on the ability to separate the variables and find a dispersion relation assuming the flow is uniform.

Anker et all [9] expanded on the theory developed by Giles by developing NRBC's in 3 dimensions for steady-state flow simulations. The theory here mimics Giles with the addition of one extra dimension. Applying the same principles as that of Giles, 3 dimensional steady NRBC's were developed and tested against Saxer's [42] quasi-three-dimensional method, a hybrid method and a full 3 dimensional NRBC. The quasi-three-dimensional method involves treating each radial point independently from one another. For this to work the assumption is taken that the circumferential variation in the flow is much greater than the radial variation. This may not be true for turbomachinery flows. The hybrid method involves a quasi-three-dimensional approach however the spanwise distribution of boundary values is prescribed. The results will be discussed in section 2.7.

2.3.5 NRBC's In Generalised Curvilinear Coordinates

Medida [10] extended the Giles boundary conditions to generalised coordinates in 2 dimensions. A review of Giles original method is present. Medida also aimed to solve the issue of corner points in the domain. The corner is of concern as it is part of both the wall and the inlet. Using the chain rule, the Euler equations are written in a curvilinear coordinate system, linearised and written in the form required for Fourier analysis and eigenanalysis. The Generalised eigenvalues are found and used to formulate:

- One dimensional, unsteady boundary conditions.
- Approximate, 2 dimensional, unsteady boundary conditions in curvilinear co-ordinates.

In the curvilinear setting, Medida also reformulated his wall corner compatibility conditions. Medida tested his formulations using Ringleb flow which will be discussed in a coming section.

Sescu [19] extended the work done by Medida [10] by attempting to generalise to 3 dimensions. The same approach present in Medida and Giles was taken here. The linearised Euler equations for uniform flow is a constant coefficient differential equation. When shifting to generalised coordinates however these coefficients can no longer be assumed to be constant, as a result of this, in order to follow the method set out by Giles, the coefficients are assumed to be locally constant and so the method can be repeated for each grid point. Eigenvectors and eigenvalues were found in the usual fashion along with the

NRBC's. Sescu developed approximate, quasi-three-dimensional, unsteady NRBC's. His paper however was purely theoretical and testing of the NRBC's is required to assess its efficiency and effectiveness. Though not an NRBC for ducted flows it is worth mentioning a paper by Ryaben'kii [43] in which boundary conditions for a domain of varying shape is developed. It is also worth looking at [44] written by Caretto for familiarity on mesh transformations for CFD calculations.

2.4 Boundary Conditions For The Wave Equation Without Flow

In this part of the literature review a more general look is taken at the current methods and state of the art for general numerical boundary conditions for acoustic and wavelike disturbances.

A paper by Givoli [15] discusses the use of local boundary conditions in both space and time based on the Sommerfeld radiation condition [45]. In his paper he mentions goals common to this thesis, that is:

- The problem and its domain D and its boundary condition is well posed, a description of which can be found in [46].
- The domain and boundary give a good approximation of the original problem when solved in an infinite domain.
- The boundary condition B is highly compatible with the numerical scheme used in the domain D.
- That the numerical method and boundary condition together must be numerically stable.
- The spurious reflections caused by the boundary are small.
- The computational effort to solve the problem in domain D with boundary condition B is not large.

In his paper Givoli focuses on problems described by the scalar wave equation:

$$u_{tt} = c^2\nabla^2 u \tag{2.102}$$

where t is time and c is the wave speed. A subscript in t here means a differential in time. Wave type solutions of the form:

$$u(\mathbf{x},t) = \tilde{u}(\mathbf{x})e^{i\omega t} \tag{2.103}$$

were assumed where ω is the wave frequency. For the one dimensional problem, in which only one spatial dimension is considered, the boundary condition is exact. The domain here is an infinitely long vibrating rod that can be truncated at any point while still remaining exact assuming the boundary condition:

$$u_t + cu_x = 0 \tag{2.104}$$

derived directly from the sommerfeld radiation condition. Using the same method in a two or three dimensional problems causes large spurious reflections and thus requires the use of what is known as pseudodifferential operators. When moving to two or three dimensions the dispersion relation relating the axial wave number to the circumferential, found by either substituting the assumed solution into the differential equation or via adopting the use of a Fourier transform, becomes irrational:

$$k_x = \pm k\sqrt{1-s^2} \quad \text{on } B, \quad s = \frac{k_y}{k} \tag{2.105}$$

where k_x is the axial wave number and $k = k_x^2 + k_y^2$. The plus and minus represent the outgoing and incoming plane waves respectively. As a result the operators present in the scalar wave equation become pseudodifferential operators, more information of which can be found in [47]. A pseudodifferential operator is non-local in both space and time and so is not practical for computational uses such as CFD. An introduction to pseudodifferential operators can be found in Ruzhansky [48]. Enquist and Majda [14] used a method in which they approximated the non-local pseudodifferential operator by a local differential operator. This was done via rational approximation of increasing accuracy of the aforementioned dispersion relation using Padé. The first order approximation is perfectly non-reflecting for plane waves at normal incidence and performance increasingly drops as wave angle increases. Many other approximation methods were used [46, 49] but the Padé approximation performs best for normal incident waves. Halpern [49] looked at alternative approximations to the pseudodifferential operators. It is also found that the application of local boundary conditions works well for high frequency acoustics, where there is a small number of grid points per wavelength. A paper by Giles [18], while focused on two dimensional non-local NRBC's, briefly discusses a local version of his boundary conditions via taking a local approximation of the left eigenvectors of the dispersion relation of a general unsteady, two dimensional, hyperbolic partial differential equation.

A paper by Grote [50] focused on time dependant scattering problems, that is a domain with an acoustic source and an obstacle diverting the waves. The problem in one dimension once again gives exact results with the equation of motion now being:

$$u_{tt} - u_{xx} = f \quad x > 0, \quad t > 0 \tag{2.106}$$

with boundary and initial values being:

$$u(0,t) = 0, \quad t > 0, \quad u(x,0) = U_0(x), \quad \frac{\partial}{\partial t}u(x,0) = V_0(x), \quad x > 0 \tag{2.107}$$

and being bounded in a domain $D = [0, L]$. The same boundary condition is found in one dimension as that in Givoli, however using characteristic theory instead of the Sommerfeld radiation condition. Going to higher dimensions results in needing, once again, local boundary conditions. Grote resorted to the same method here in approximating the pseudodifferential operators. It is also seen, in this paper, that going into three dimensional polar coordinates allows one to derive exact NRBC's where now the waves were mn-th spherical harmonics, $Y_{nm}(\theta, \phi)$, multiplied by some function $u_{nm}(r, t)$. Grote derived boundary conditions for $n = 0$ and $n \geq 0$, using the work laid out by Sofronov [51,52]. Sofronov has a series of papers detailing boundary conditions in spherical coordinates for the wave equation [51–55].

Tsynkov [16] used the same methods above to write boundary conditions for a half line problem:

$$\nabla^2 u = \mu^2 u = f, \quad x \geq 0 \tag{2.108}$$

$$u(0) = 0 \tag{2.109}$$

$$u(x) \to 0, \text{ as } x \to \infty \tag{2.110}$$

The boundary condition in one dimension is exact while the boundary condition in higher dimensions needs approximate expansions of pseudodifferential operators. It should be noted that in developing boundary conditions with problems involving a source term f the assumption is made that its value is 0 outside the domain. Gustafsson [56] developed boundary conditions where this assumption was lifted leading to an inhomogeneous boundary condition. The development required the separation of variables via Laplace transforms in time and Fourier transforms in space.

Work by Ting and Miksis [17] considered the problem of a scatterer in a bounded domain. The wave is split into two components: $\psi = \psi_i + \psi_s$. The incident and scattered contributions respectively. $H(\psi) = 0$ was the condition inside the scatterer for $x \in D$, where H is a nonlinear operator. The scattered part of the wave was represented using the Kirchhoff formula, derivation for which can be found in [57]. This treats the scattered wave field as infinitely many spherical waves that are generated by a continuous distribution of sources on the boundary of the domain. This means the values on the boundary are dependent on past values of the solution on some surface S inside the domain. This can help one achieve an exact solution to the problem however this requires large computational power and memory requirements. If N is the number of grid points in one dimension then the number of iterations required per timestep would be N^3 making it computationally expensive to progress the solution, along with this is the requirement of the solution represented as a Kirchhoff integral to store past values of the solution on S for the length of time it takes one wave to cross the domain, which requires large amounts of memory. These therefore make it difficult to implement the method in a numerical scheme. Colonius [58] gives a detailed overview on the developments of many types of ABC's including absorbing layers such as the PML and NRBC's.

2.5 Boundary Conditions for The Euler Equations

Tsynkov [59,60] derived boundary conditions for the linearised Euler equations with source terms. The boundary conditions developed here rely on what is known as lacunae. If one pictures the area in space-time that the waves have travelled through, the lacuna is the part of the space-time domain where the solution has become zero again, that is, the wave has passed. The requirement to these boundary conditions is that the flow be less than the speed of sound and the equation be linear. It is also required to know the analytical solution in the physical domain in order to apply the method computationally. The boundary conditions are based on the premise that no acoustic wave can reach a distance $x = ct$ where c is the wave speed and t is the time passed, as such the pressure disturbance at a distance just after this point is always zero. Though not entirely relevant to this thesis Tsynkov [61] also formulates external boundary conditions for viscous flows for the compressible and incompressible fluids for the linearised Euler equations with additional viscous terms. In this paper the boundary conditions also have corrections for non-linear effects present at higher Mach numbers. Tsynkov also has a review on the computational problems on unbounded domains [16].

Poinsot and Lele [2] developed characteristic boundary conditions for the Navier-Stokes equations in three dimensions. These boundary conditions at the outflow boundary have been implemented in the commercially available ANSYS FLUENT solver. The EBNRBC's developed in this thesis will be compared with Poinsot and Leles method in chapters 4,5, and 6. The boundary condition is based off of characteristic analyses using Linear Relaxation Method to find the characteristic variables at the boundary. The method involves characteristic analysis of the Navier-Stokes equation in order to modify terms corresponding to waves travelling along one spatial direction. Using this analysis, wave amplitudes can be determined and the eigenvalues of the system can be used to determine the outgoing and incoming waves and once again cancelling any waves that should not be entering the domain.

Assuming a compressible viscous flow, the Navier-Stokes equations are:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(m_i) = 0 \tag{2.111}$$

$$\frac{\partial \rho E}{\partial t} + \frac{\partial}{\partial x_i}\left[(\rho E + p)u_i\right] = \frac{\partial}{\partial x_i}(u_j \tau_{ij}) - \frac{\partial q_i}{\partial x_i} \tag{2.112}$$

$$\frac{\partial m_i}{\partial t} + \frac{\partial}{\partial x_j}(m_i u_j) + \frac{\partial p}{\partial x_i} = \frac{\partial \tau_{ij}}{\partial x_j} \tag{2.113}$$

where:

$$\rho E = \frac{1}{2}\rho u_k u_k + \frac{p}{\gamma - 1} \tag{2.114}$$

$$m_i = \rho u_i \tag{2.115}$$

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right) \tag{2.116}$$

In these sets of equations p is the thermodynamic pressure, m_i is the momentum density in the x_i direction, ρE is the total energy density and q_i is the heat flux in the x_i direction which is given by:

$$q_i = -\lambda \frac{\partial T}{\partial x_i} \tag{2.117}$$

Here λ is the thermal conductivity and is obtained from the viscosity coefficient μ:

$$\lambda \mu C_p / P_r \tag{2.118}$$

where finally P_r is the Prandtl number.

Characteristic analysis is used to modify the hyperbolic terms in equations 2.111 - 2.113 that correspond to waves travelling in the x_1 direction towards the $x_1 = L$ boundary shown in figure 2.4. The system of equations are now written as:

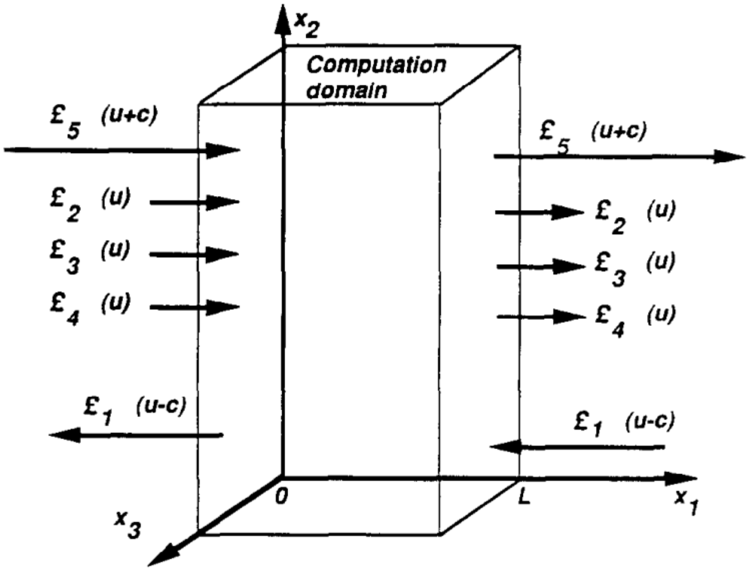


Figure 2.4: Taken from [2] this shows waves leaving and entering a computational domain through an inlet plan at $x_1 = 0$ and an outlet plane at $x_1 = L$.

$$\frac{\partial \rho}{\partial t} + d_1 + \frac{\partial}{\partial x_2}(m_2) + \frac{\partial}{\partial x_3}(m_3) = 0 \tag{2.119}$$

$$\begin{aligned}\frac{\partial \rho E}{\partial t} + \frac{1}{2}u_k^2 d_1 + \frac{d_2}{\gamma - 1} + m_1 d_3 + m_2 d_4 \\ + m_3 d_5 + \frac{\partial}{\partial x_2}\left[(\rho E + p)u_2\right] + \frac{\partial}{\partial x_3}\left[(\rho E + p)u_3\right] \\ = \frac{\partial}{\partial x_i}(u_j \tau_{ij}) - \frac{\partial q_i}{\partial x_i}\end{aligned} \tag{2.120}$$

$$\frac{\partial m_1}{\partial t} + u_1 d_1 + \rho d_3 + \frac{\partial}{\partial x_2}(m_1 u_2) + \frac{\partial}{\partial x_3}(m_1 u_3) = \frac{\partial \tau_{1j}}{\partial x_j} \tag{2.121}$$

$$\frac{\partial m_2}{\partial t} + u_2 d_1 + \rho d_4 + \frac{\partial}{\partial x_2}(m_2 u_2) + \frac{\partial}{\partial x_3}(m_2 u_3) + \frac{\partial p}{\partial x_2} = \frac{\partial \tau_{2j}}{\partial x_j} \tag{2.122}$$

$$\frac{\partial m_3}{\partial t} + u_3 d_1 + \rho d_5 + \frac{\partial}{\partial x_2}(m_3 u_2) + \frac{\partial}{\partial x_3}(m_3 u_3) + \frac{\partial p}{\partial x_3} = \frac{\partial \tau_{3j}}{\partial x_j} \tag{2.123}$$

A vector **d** is defined as:

$$\mathbf{d} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{pmatrix} = \begin{pmatrix} \frac{\partial m_1}{\partial x_1} \\ \frac{\partial c^2 m_1}{\partial x_1} + (1-\gamma)\mu\frac{\partial p}{\partial x_1} \\ u_1\frac{\partial u_1}{\partial x_1} + \frac{1}{\rho}\frac{\partial p}{\partial x_1} \\ u_1\frac{\partial u_2}{\partial x_1} \\ u_1\frac{\partial u_3}{\partial x_1} \end{pmatrix} = \begin{pmatrix} \frac{1}{c^2}\left[\mathcal{L}_2 + \frac{1}{2}(\mathcal{L}_5 + \mathcal{L}_1)\right] \\ \frac{1}{2}(\mathcal{L}_5 + \mathcal{L}_1) \\ \frac{1}{2\rho c}(\mathcal{L}_5 + \mathcal{L}_1) \\ \mathcal{L}_3 \\ \mathcal{L}_4 \end{pmatrix} \tag{2.124}$$

where

$$\mathcal{L}_1 = \mathcal{V}_1\left(\frac{\partial p}{\partial x_1} - \rho c\frac{\partial u_1}{\partial x_1}\right) \tag{2.125}$$

$$\mathcal{L}_2 = \mathcal{V}_2\left(c^2\frac{\partial \rho}{\partial x_1} - \frac{\partial p}{\partial x_1}\right) \tag{2.126}$$

$$\mathcal{L}_3 = \mathcal{V}_3\frac{\partial u_2}{\partial x_1} \tag{2.127}$$

$$\mathcal{L}_4 = \mathcal{V}_4\frac{\partial u_3}{\partial x_1} \tag{2.128}$$

$$\mathcal{L}_5 = \mathcal{V}_5\left(\frac{\partial p}{\partial x_1} + pc\frac{\partial u_1}{\partial x_1}\right) \tag{2.129}$$

where $\mathcal{V}_i$ are the characteristic velocities:

$$\mathcal{V}_1 = u_1 - c \tag{2.130}$$

$$\mathcal{V}_2 = \mathcal{V}_3 = \mathcal{V}_4 = u_1 \tag{2.131}$$

$$\mathcal{V}_5 = u_1 + c \tag{2.132}$$

and c is the speed of sound given by:

$$c^2 = \frac{\gamma p}{\rho} \tag{2.133}$$

The vector $\mathbf{d}$ contains derivatives normal to the boundary at x_1 and is given by characteristic analysis found in [62]. Parallel values are ones of the type : $(\partial/\partial x_2)(\rho u_2)$. The values $\mathcal{L}_i$ are the amplitudes of the characteristic waves associated with each characteristic velocity $\mathcal{V}_i$. Looking at the linearised Navier-Stokes equation for one-dimensional inviscid acoustic waves it is possible to give a physical interpretation of $\mathcal{L}_i$. Consider an upstream -propagating wave associated to the characteristic velocity $\mathcal{V}_1 = u_1 - c$. Allow p' and u' to be the pressure and velocity perturbations respectively. The wave amplitude $A_1 = p' - \rho c u'$ is conserved along the characteristic line $x + \mathcal{V}_1 t$ giving:

$$\frac{\partial A_1}{\partial t} + \mathcal{V}_1 \frac{\partial A_1}{\partial x_1} = 0 \quad or \quad \frac{\partial A_1}{\partial t} + \mathcal{L}_1 = 0 \tag{2.134}$$

This shows that $-\mathcal{L}_1$ represents the time variation of the wave amplitude A_1. This is true for all $\mathcal{V}_i$. Note that the viscous terms are not included in the expressions for $\mathcal{V}_i$. Poinsot and Lele approximate the wave amplitudes in the viscous case by their inviscid expression.

An approximation for the incoming wave amplitude variations is needed. To obtain the approximation Local One-Dimensional Inviscid (LODI) relations are used. A LODI system is considered by neglecting both viscous and transverse terms. The equations then allow one to infer values for the wave amplitude variations by considering the flow locally as inviscid and one-dimensional. Writing in terms of primitive variables the LODI system in this case is :

$$\frac{\partial \rho}{\partial t} + \frac{1}{c^2}\left[\mathcal{L}_2 + \frac{1}{2}(\mathcal{L}_5 + \mathcal{L}_1)\right] = 0 \tag{2.135}$$

$$\frac{\partial \rho}{\partial t} + \frac{1}{2}(\mathcal{L}_5 + \mathcal{L}_1) = 0 \tag{2.136}$$

$$\frac{\partial u_1}{\partial t} + \frac{1}{2\rho c}(\mathcal{L}_5 + \mathcal{L}_1) = 0 \tag{2.137}$$

$$\frac{\partial u_2}{\partial t} + \mathcal{L}_3 = 0 \tag{2.138}$$

$$\frac{\partial u_3}{\partial t} + \mathcal{L}_4 = 0 \tag{2.139}$$

Poinsot and Lele mention that these relations can be combined in order to find the time derivatives of other quantities of interest such as the temperature or the flow rate. It should be noted that most physical boundary conditions will have a LODI relation.

It should be noted that the boundary variables will be time advanced using the system eq (2.119) - (2.123) where viscous terms and parallel terms are considered and so the approximation is negligible.

Poinsot and Lele outlined three steps to produce boundary conditions for the Euler equations.

1. Eliminate the corresponding conservation equations from the system of equations (2.119) - (2.123) for each inviscid physical boundary condition imposed on the boundary.
2. Express the $\mathcal{L}_i$ corresponding to incoming waves as functions of the $\mathcal{L}_i$ corresponding to outgoing waves which should be known. This should be done for each inviscid boundary condition using the LODI relations.
3. Using the values of $\mathcal{L}_i$ obtained from step two and the remaining conservation equations in system (2.119) - (2.123), compute all variables which were not given by the inviscid boundary conditions. In the case of a constant pressure outlet, the density and velocities are obtained through the corresponding conservation equations (2.119), (2.121) - (2.123) where $\mathcal{L}_1 = -\mathcal{L}_5$

To elaborate on the steps above, Poinsot and Lele mention that step 2 is the key part of the method. The use of LODI relations to find reasonable information on the amplitude of the incoming waves as well as using conservation equations formulated on the boundary removes the need to have some arbitrary numerical condition. The time advancement of step 3 includes parallel terms to obtain the solution at the next timestep. Step 1 is also necessary to discard equations in the system (2.119) - (2.123) which are then replaced by inviscid boundary conditions. This is needed as the system of equations and the LODI relations would not satisfy the physical boundary conditions imposed.

Poinsot and Lele also generalise to the Navier-Stokes equations. Complete Navier-Stokes boundary conditions are obtained by using inviscid boundary conditions while adding viscous conditions. The viscous conditions should have negligible effect when viscosity goes to zero. These conditions are added during step 3 and should only be used to modify the conservation equations used in step 3. Steps 1 and 2 remain the same. Poinsot and Lele discuss in more detail how these viscous conditions are chosen.

A paper by Hixon [63] uses the Thompson Characteristic method [62, 64] to develop improved mean flow boundary conditions for ducted flows described by the Euler equations with source terms for Computational Aero Acoustics (CAA) calculations. This method also required incoming and outgoing waves to be specified. A unit normal $\bar{n}$ is defined and points outward from the computational domain and can be used to find the propagation velocity with respect to the computational boundary. Waves with a negative propagation velocity are entering the domain whilst waves with a positive propagation velocity are exiting the domain. Using this method, four types of boundary conditions are found: subsonic inflow and outflow, supersonic inflow and outflow. The mean flow boundary conditions then uses this information to set a correct mean pressure at the boundaries while still allowing the acoustic disturbances to leave the domain.

2.5.1 Eigenanalysis in Slowly Varying Duct Geometries

Rienstra [3, 21] discusses the transmission of sound through ducts of slowly varying radius using the method of multiple scales. The requirement of having a slowly varying radius means one can utilise a small parameter related to the slow radial variation in the duct using the method of multiple scales. The mean flow is assumed to be a compressible inviscid isentropic irrotational flow. As a result potential flow is used; $\tilde{u} = \nabla\tilde{\phi}$ where the tilde represents the combined mean part and perturbation part of the variable. Rienstra splits the problem up into equations of motion describing the mean flow and the equations of motion describing acoustic field. For the mean flow a modified hard walled boundary condition is used to take into account the slowly varying duct. Working in cylindrical coordinates and considering an annular duct, the radii are given by:

$$r = R_1(X), \quad r = R_2(X), \quad X = \epsilon x, \tag{2.140}$$

where ϵ is the small parameter and R_1 and R_2 are the inner and outer radius respectively. $R_{1,2}$ is dependant only upon ϵ. The modified hard walled boundary condition is then

$$u_0 \cdot n_i = 0 \quad \text{at} \quad r = R_i(X) \quad i = 1, 2 \tag{2.141}$$

where

$$n_{1,2} = \mp \frac{e_r - \epsilon R'_{1,2} e_x}{(1 + \epsilon^2 R'^2_{1,2})^{1/2}} \tag{2.142}$$

The acoustic field has an impedance boundary condition given by Myers [65]

$$i\omega(u \cdot n_i) = [i\omega + u_0 \cdot \nabla - n_i \cdot (n_i \cdot \nabla u_0)] \left(\frac{p}{Z_i} \right) \quad \text{at} \quad r = R_i(X) \quad i = 1, 2 \tag{2.143}$$

In order to find a unique solution, near uniform flow is assumed with axial variations only in X:

$$u_0 = u_x(X, r, \epsilon) e_x + u_r(X, r, \epsilon) e_r \tag{2.144}$$

whilst the acoustic field consists of a constant duct mode perturbed by variations in X. The velocities u_x and u_r are expanded about ϵ. Rienstra [3] notes that small axial mass variations are balanced by small radial variations therefore giving:

$$u_x(X, r; \epsilon) = u_{x0}(X) + O(\epsilon^2), \quad u_r(X, r; \epsilon) = \epsilon u_{r1}(X, r) + O(\epsilon^3) \tag{2.145}$$

where

$$u_{x0}(X) = \frac{F}{\bar{\rho}_0(X)(R_2^2(X) - R_1^2(X))} \tag{2.146}$$

and

$$u_{r1}(X, r) = \frac{F}{2r\bar{\rho}_0(X)} \frac{\partial}{\partial X} \frac{r^2 - R_1^2(X)}{(R_2^2(X) - R_1^2(X))} \tag{2.147}$$

u_{r1} is found using the continuity equation and the boundary condition:

$$-\frac{\mathrm{d}R_i}{\mathrm{d}X} u_{x0} + u_{r1} = 0 \quad \text{at} \quad r = R_i(X) \quad (i = 1, 2) \tag{2.148}$$

The rest of the flow variables are given by:

$$\bar{f}(X, r; \epsilon) = \bar{f}_0(X) + O(\epsilon^2) \tag{2.149}$$

where $\bar{f} = \bar{p}, \bar{\rho}, \bar{c}$, where the bar represent the mean values of the pressure, density and speed of sound respectively.

The modes in the current case are modes whose axial and radial wave numbers depend on X, that is, are slowly varying:

$$\phi(x,r,\theta;\epsilon) = \phi(X,r;\epsilon)\exp\left(-im\theta - i\epsilon^{-1}\int^{X}\mu(\zeta)\mathrm{d}\zeta\right) \tag{2.150}$$

where ϕ is the perturbation part of $\tilde{\phi}$. The differential operator with respect to x is transformed as a result:

$$\frac{\partial}{\partial x} = -i\mu(X) + \epsilon\frac{\partial}{\partial X} \tag{2.151}$$

A more detailed derivation can be found in [3] however assuming:

$$\phi(X,r;\epsilon) = \phi_0(X,r) + \epsilon\phi_1(X,r) + ... \tag{2.152}$$

the acoustic solution here takes the form of a linear combination of Bessel functions of both the first and second kind that changes with X

$$\phi_0(X,r) = N(X)J_m(\alpha(X)r) + M(X)Y_m(\alpha(X)r) \tag{2.153}$$

whilst the boundary condition produces:

$$\frac{\alpha R_2 J'(\alpha R_2) - \delta_2 J_m(\alpha R_2)}{\alpha R_2 Y'(\alpha R_2) - \delta_2 Y_m(\alpha R_2)} = \frac{\alpha R_1 J'(\alpha R_1) + \delta_1 J_m(\alpha R_1)}{\alpha R_1 Y'(\alpha R_2) + \delta_1 Y_m(\alpha R_2)} = -\frac{M(X)}{N(X)} \tag{2.154}$$

where

$$\delta_i = \frac{\Omega^2\bar{\rho}_0 R_i}{i\omega Z_i}, \quad \Omega = \omega - k_{sv}u_{x0} \tag{2.155}$$

from which the eigenvalue α can be found. k_{sv} is the axial wave number in a slowly varying duct. Here the radial wave number and coefficient of the Bessel functions are dependent on X unlike that seen in (2.14). It is important to note that it is not required to find an expression for ϕ_1, only that there is a solvibility condition which indeed there is [66].

The next stage is to find an expression for $N(X)$ which for the hard walled case is

$$\left(\frac{1/2\pi Q_0}{N}\right)^2 = \frac{\bar{\rho}\omega\sigma}{2\bar{c}}\left(\frac{R_2^2 - m^2/\alpha^2}{\left(\alpha R_2 Y_m'(\alpha R_2)\right)^2} - \frac{R_1^2 - m^2/\alpha^2}{\left(\alpha R_1 Y_m'(\alpha R_1)\right)^2}\right) \tag{2.156}$$

The derivation of which can be found again in [3]. Here σ is the reduced axial wave number:

$$\sigma = \left(1 - (\bar{c}_0^2 - \bar{u}_0^2)\frac{\alpha^2}{\omega^2}\right)^{1/2} \tag{2.157}$$

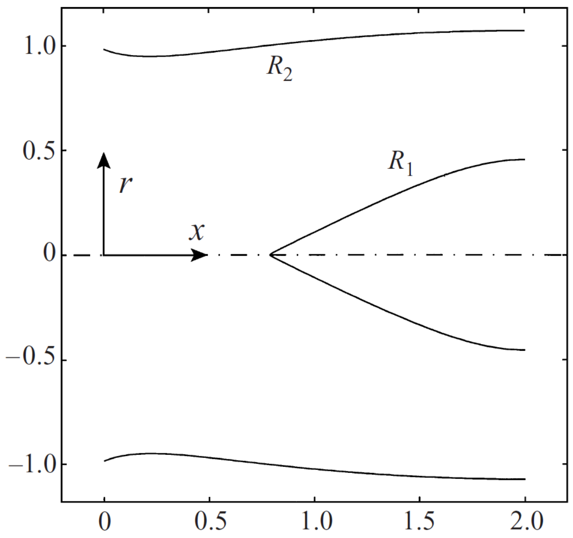


Figure 2.5: Sketch of the hollow to annular duct geometry taken from Rienstra [3].

Using the presented techniques, Rienstra was one of the first to develop a method for mode analysis for the hollow to annular duct problem so long as the annular part was introduced "slowly" as seen in figure 2.5. An example of this is given in his examples section [3] and the results displayed give good results. Rienstra [22] generalised the solutions for irrotational isentropic mean flow to flows with slowly varying modes in ducts of arbitrary cross section.

Peake and cooper [23], also applied the method of multiple scales to ducts with slowly varying elliptic cross section. In this case the eccentricity of the duct varies slowly along the axial direction of the duct where the eccentricity is a function of the axial coordinate and a small parameter similar to what was seen in the axisymmetric case: $e = e(\epsilon x)$. The standard transformation from elliptical to Cartesian then becomes:

$$y = e(\epsilon x)\cosh\tau\cos\alpha, \quad z = e(\epsilon x)\sinh\tau\sin\alpha \tag{2.158}$$

where τ and α are analogous to the radius and angle in cylindrical polar coordinates. The assumptions and the equations of motion are equivalent to those in Rienstra [3]. Peake and cooper make it a point to mention the coordinate system is not orthogonal. Using standard results in differential geometry the definition of the gradient and divergence in this space was found giving:

$$\nabla\phi = \frac{1}{\sqrt{g_{11}}}\frac{\partial\phi}{\partial\tau}e_\tau + \frac{1}{\sqrt{g_{11}}}\frac{\partial\phi}{\partial\alpha}e_\alpha + \frac{1}{\sqrt{g_{11}}}\frac{\partial\phi}{\partial x}e_x \tag{2.159}$$

The metric tensor g_{ij} for this space is

$$g_{ij} = \begin{pmatrix} h_\tau^2 & 0 & \epsilon g_{13} \\ 0 & h_\tau^2 & \epsilon g_{23} \\ \epsilon g_{13} & \epsilon g_{23} & \epsilon^2 \tilde{g}_{33} \end{pmatrix}, \quad h_\tau = e\sqrt{\sinh^2\tau + \sin^2\alpha} \tag{2.160}$$

The metric terms, g_{13}, g_{23} and $\tilde{g}_{33}$ are all O(1) variables and are subsequently not needed to form a solution.

The steady state equations of motion for this problem are equivalent to those present in [3]. The boundary conditions are conceptually same for that in [3] with mathematical differences appearing due to the difference in space i.e. a slowly varying elliptical space effecting the unit normals:

$$n_{1,2} = \mp\frac{h_\tau^{-1}e_\tau - \epsilon\tau'_{1,2}e_x}{(h_\tau^{-2} + \epsilon^2\tau'^2_{1,2})^{1/2}} \tag{2.161}$$

The solution follows a similar pattern with a result analogous to that of a linear combination of Bessel functions known as modified Mathieu functions multiplied by Mathieu functions. Two distinct set of solutions are possible, even about the major axis and odd written as:

$$\left(M_m(X)Ce_m(\tau,q) + N_m(X)Fe_m(\tau,q)\right) ce_m(\theta,q) \tag{2.162}$$

$$\left(M_m(X)Se_m(\tau,q) + N_m(X)Ge_m(\tau,q)\right) se_m(\theta,q) \tag{2.163}$$

respectively, where q is analogous to the radial mode number. $ce_m(\theta,q)$ and $se_m(\theta,q)$ are the Mathieu functions which in this case are analogous to cos θ and sin θ respectively. $Ce_m(\tau,q)$, $Se_m(\tau,q)$, $Fe_m(\tau,q)$ and $Ge_m(\tau,q)$ are the modified Mathieu functions analogous to the Bessel functions. The first order form of the boundary condition is $\partial\phi_0/\partial\tau = 0$ at $\tau = \tau_{1,2}$ which when applied gives the dispersion relation for non-trivial solution:

$$Ce'_m(\tau_1,q)Fe'_m(\tau_2,q) = Ce'_m(\tau_2,q)Fe'_m(\tau_1,q) \tag{2.164}$$

and

$$Se'_m(\tau_1,q)Ge'_m(\tau_2,q) = Se'_m(\tau_2,q)Ge'_m(\tau_1,q) \tag{2.165}$$

for the even and odd solutions respectively. Similar to the slowly varying example above the next step is to obtain an equation for $M(X)$ which, for the even modes, is:

$$M_m^2(X) = \frac{Q_0^2 C_0^2(X) {Fe'_m}^2(\tau_2, q)}{(C_0^2 k_e + \Omega u_{x0})\rho_0 I} \quad \Omega = \omega - k_e u_{x0} \tag{2.166}$$

where

$$I = \int_0^{2\pi} \int_{\tau_1}^{\tau_2} ce_m^2(\theta, q) \left(Ce_m(\tau, q) Fe'_m(\tau_2, q) - Ce'_m(\tau_2, q) Fe_m(\tau_1, q) \right)^2 h_\tau^2 \mathrm{d}\tau \mathrm{d}\theta. \tag{2.167}$$

Here k_e is the axial wave number for a duct with slowly varying elliptical cross section. $N_m(X)$ is then

$$N_m(X) = -M_m(X) \frac{Ce'_m(\tau_2, q)}{Fe'_m(\tau_2, q)} \tag{2.168}$$

A more detailed derivation is available in [23]. It is possible to split the integral into several terms which involve an integral in τ multiplied by an integral in θ. Some integrals can be solved in closed form while the rest need to be evaluated numerically. In order to derive the results for the odd modes simply replace Ce_m and ce_m by Se_m and se_m respectively.

Peake and cooper [24] extended the method to solve the problem of a slowly varying duct with swirling mean flow. Swirl causes all mean flow terms to vary in X and r where X here is unchanged in meaning and r is the radius. Again it is a requirement that the variation of the duct is slow. The flow field now has the form:

$$u_0 = u_x(X, r, \epsilon) e_x + u_r(X, r, \epsilon) e_r + u_\theta(X, r\epsilon) e_\theta, \tag{2.169}$$

The velocity field is given by:

$$\begin{aligned} u_x(X, r; \epsilon) &= u_{x0}(X, r) + O(\epsilon^2), && (2.170) \\ u_r(X, r; \epsilon) &= \epsilon u_{r1}(X, r) + O(\epsilon^3) && (2.171) \\ u_\theta(X, r; \epsilon) &= u_{\theta 0}(X, r) + O(\epsilon^2) && (2.172) \end{aligned}$$

giving a vorticity of the form:

$$\nu = \frac{1}{r} \frac{\partial (r u_{\theta 0}) 1}{\partial r} e_x - \epsilon \frac{\partial (u_{\theta 0})}{\partial X} e_r - \frac{\partial u_{x0}}{\partial r} e_\theta + O(\epsilon^2) \tag{2.173}$$

Writing the $O(\epsilon)$ continuity equation in terms of streamlines functions and using Bernoulli's theorem along with the use of Croccos relation it is possible to obtain the equations of motion of the mean flow:

$$\frac{\partial}{\partial r}\left(\frac{1}{r\rho_0}\frac{\partial\psi}{\partial r}\right) = r\rho_0 H'(\psi) - \rho_0 C'(\psi)\frac{C}{r} \tag{2.174}$$

$$\frac{1}{2}\frac{1}{r^2\rho_0^2}\left(\frac{\partial\psi}{\partial r}\right)^2 + \frac{C^2}{2r^2} + \frac{\rho_0^{\gamma-1}}{\gamma-1} = H(\psi) \tag{2.175}$$

where here ψ is the streamline function $\psi(X, r)$, H is the enthalpy and C is the circulation. The accompanying boundary conditions are:

$$\psi(X, r = R_1) \quad = \quad 0 \tag{2.176}$$

$$\psi(X, r = R_2) \quad = \quad \text{constant} \tag{2.177}$$

For a more detailed derivation visit [67]. These equations can be solved numerically for any prescribed inlet flow.

Due to the presence of vorticity the perturbation velocity is split into potential and rotational parts:

$$u' = \nabla\phi + \mathbf{u}^R \tag{2.178}$$

while the pressure is described in terms of the unsteady potential ϕ:

$$p' = -D_0\frac{\mathrm{D}\phi}{\mathrm{D}t} \tag{2.179}$$

where the operator above is the convected derivative. Then from the linearised Euler equations become two coupled equations:

$$\frac{\mathrm{D}\mathbf{u}^R}{\mathrm{D}t} + \left(\mathbf{u}^R\cdot\nabla\right)u_r \quad = \quad -\nu\times\nabla\phi \tag{2.180}$$

$$\left[\frac{\mathrm{D}}{\mathrm{D}t}\frac{1}{C_0^2}\frac{\mathrm{D}}{\mathrm{D}t} - \frac{1}{D_0}\nabla\cdot(D_0\nabla)\right]\phi \quad = \quad \frac{1}{D_0}\nabla\cdot(D_0\mathbf{u}^R) \tag{2.181}$$

The derivation for the main result can be found in [67], here a quick overview will be given. Solutions of the form:

$$(\phi, u_x^R, u_r^R, u_\theta^R)(x, r, \theta, t; \epsilon) = (A, \mathcal{X}, \mathcal{R}, \mathcal{T})(X, r; \epsilon)\exp\left(i\omega t - im\theta - \frac{i}{\epsilon}\int^X k(\eta)\mathrm{d}\eta\right) \tag{2.182}$$

are assumed. The boundary conditions in this case are:

$$\begin{aligned} i\omega\left(\frac{\partial A}{\partial r}+\mathcal{R}\right)\pm\frac{D_0\Lambda^2 A}{Z_j} &= \epsilon\omega\frac{\mathrm{d}R_j}{\mathrm{d}X}(kA+i\mathcal{X}) \\ \pm\frac{i\epsilon}{A}\left[u_{x0}\frac{\partial}{\partial X}+u_{r1}\frac{\partial}{\partial r}-\frac{\partial u_{r1}}{\partial r}+\frac{\mathrm{d}R_j}{\mathrm{d}X}\frac{\partial u_{x0}}{\partial r}\right]&\left(\frac{D_0\Lambda A^2}{Z_j}\right) \end{aligned} \tag{2.183}$$

which can be directly derived from Equation (2.143). Here $\Lambda = \omega - ku_{x0} - mu_{\theta 0}/r$. Expanding the assumed solution in powers of ϵ and after some algebra sets of coupled equations at O(1) and O(ϵ) are found:

$$(\mathcal{L}-k\mathcal{K})\psi_0 \equiv \mathcal{P}\psi_0 = 0 \tag{2.184}$$

$$(\mathcal{L}-k\mathcal{K})\psi_0 1 \equiv \mathcal{P}\psi_1 = \mathbf{f} \tag{2.185}$$

where $\psi_n = (A_n, \eta_n, \mathcal{R}_n, i\mathcal{T}_n)$ and $\eta_n = k\beta_0^2 A_n$ $n = 0, 1$. This variable is introduced in order to allow the system to be written as a linear eigenvalue problem [68]. The expressions for $\mathcal{L}$, $\mathcal{K}$ and $\mathbf{f}$ are given in appendix A of [67]. The corresponding boundary conditions are then:

$$i\omega\left(\frac{\partial A_0}{\partial r}+\mathcal{R}_0\right)\pm\frac{D_0\Lambda^2 A_0}{Z_j} = 0 \tag{2.186}$$

$$\begin{aligned} i\omega\left(\frac{\partial A_1}{\partial r}+\mathcal{R}_1\right)\pm\frac{D_0\Lambda^2 A_1}{Z_j} &= \omega\frac{\mathrm{d}R_j}{\mathrm{d}X}(kA_0+i\mathcal{X}_0) \\ \pm\frac{i\epsilon}{A_0}\left[u_{x0}\frac{\partial}{\partial X}+u_{r1}\frac{\partial}{\partial r}-\frac{\partial u_{r1}}{\partial r}+\frac{\mathrm{d}R_j}{\mathrm{d}X}\frac{\partial u_{x0}}{\partial r}\right]&\left(\frac{D_0\Lambda A_0^2}{Z_j}\right) \end{aligned} \tag{2.187}$$

Equation (2.184) together with boundary condition (2.186) are solved numerically, the details of which can be found in [67]. The results of the calculation gives the leading order solution up to an arbitrary slowly varying function:

$$\psi_0(X,r) = N(X)\hat{\psi}_0(X,r) = N(X)(\hat{A}_0, \hat{\eta}_0, \hat{\mathcal{R}}_0, i\hat{\mathcal{T}}_0) \tag{2.188}$$

where after some calculation $N(X)$ is given by:

$$N^2(X) = N_0^2\exp\left(\int^X \frac{G(\eta)}{F(\eta)}\mathrm{d}\eta\right) \tag{2.189}$$

Where N_0 is a normalisation constant and the expressions for F and G are found in appendix B of [67].

2.6 Zonal Boundary Treatments

2.6.1 Large Domain

The most basic of the zonal boundary treatments is to simply create a large enough domain such that effects caused within the area of interest become small and negligible. At the edge of the large domain is simply a Dirichlet boundary condition. Of course this leads to reflections but with the solution effects being small this does nothing to spoil the solution. The issue with this kind of boundary condition comes with the fact that the solutions of acoustic problems are typically oscillating and decay with distance r as $1/r^{(d-1)/2}$, where d is the number of dimensions. This slow decay means the domain has to be rather large compared to our area of interest. As one can expect this increases simulation time and requires higher processing power. An ideal boundary condition should allow one to simulate closer to the area of interest in hopes of reducing time taken to obtain a solution and potentially processing power. This is without mentioning that in ducted flows a larger domain fails entirely as propagating waves travel to infinity.

2.6.2 Stretched Grid

The theory of stretched grids will be discussed using Kreiss [4] as a base. Looking at a 2 dimensional hyperbolic equation:

$$\frac{\partial u}{\partial t} + A\frac{\partial u}{\partial x} + B\frac{\partial u}{\partial y} = 0 \quad x \geq 0 \tag{2.190}$$

Fourier transforming in y and t:

$$i\omega\hat{u} + A\frac{\partial \hat{u}}{\partial x} + ilB\hat{u} = 0 \quad x \geq 0 \tag{2.191}$$

where l is the wave number in the y direction and ω is the temporal frequency. This is a system of ordinary differential equations for each l and ω:

$$\frac{\partial \hat{u}}{\partial x} = -i\omega A^{-1}\left(1 - \frac{l}{\omega}B\right)\hat{u}, \quad x \geq 0 \tag{2.192}$$

The solution can be expressed in terms of eigen solutions:

$$u_k(x, y, t) = e^{i(\omega t - kx - ly)}\boldsymbol{u}_k \tag{2.193}$$

where k is the wave number and ω is the temporal frequency which are related via $\omega = ck$. Discretising this in space, using central differences yields:

$$i\omega\hat{v}_j + AD_0\hat{v}_j + ilB\hat{v}_j = 0 \tag{2.194}$$

where $D_0 = \frac{v_{j+1}-v_{j-1}}{2h}$ and h is the distance between points. Rewriting (2.194) as shown in (2.192):

$$D_0\hat{v}_j = -i\omega A^{-1}\left(1 + \frac{l}{\omega}B\right)\hat{v}_j. \tag{2.195}$$

The system is then diagonalized using the eigenvectors $\boldsymbol{u}_k$,

$$D_0\hat{z}_j^{(k)} = -ik\hat{z}_j^{(k)} \tag{2.196}$$

Solving this system of differential equation an ansatz is formed, $\hat{v}_j = \eta^j$, where η solves the characteristic equation:

$$\eta^2 + i2kh\eta - 1 = 0 \tag{2.197}$$

whose roots are:

$$\eta_\pm = -ikh \pm \sqrt{1-(kh)^2} \tag{2.198}$$

For different mesh resolutions in space the roots satisfy three conditions:

- For $0 < kh < 1$ the roots have an absolute value unity, corresponding to a propagating wave.
- For $kh = 1$ both roots equal $-i$. This causes a solution with a linearly growing component.
- For $kh > 1$, the roots $|\eta_+| < 1$ and $|\eta_-| > 1$ correspond to evanescent waves, decaying and growing respectively.

The grid stretching will take place here in one dimension by using the equation $x_{j+1} = sx_j$, where s is the stretching factor. To achieve efficient damping the grid is stretched such that out going waves are under-resolved. The approximate solutions for the decaying and growing modes are:

$$v_n^{k+}(y,t) = \left(\Pi_{j=0}^n \eta_{j+}^k\right) e^{i(\omega t + ly)}\phi_k \tag{2.199}$$

$$\eta_{j+}^{(k)} = -ikh_j + \sqrt{1-(kh_j)^2} \tag{2.200}$$

for decaying modes and

$$v_n^{k-}(y,t) = \left(\Pi_{j=0}^{n} \eta_{j-}^{k}\right) e^{i(\omega t + ly)} \phi_k \tag{2.201}$$

$$\eta_{j-}^{(k)} = -ikh_j - \sqrt{1-(kh_j)^2} \tag{2.202}$$

for growing modes, where ϕ_k is the eigenvector in the discretised setting. As distance between points, h, increases waves will enter the evanescent regime and decay exponentially. This was then implemented for use with the 2 dimensional linearised Euler equations. Artificial dissipation is also added to aid in the damping of the waves. This can aid in making the domain even smaller thus being more efficient with computing power and run time. With this implementation, it was shown that a one dimensional wave, in a fluid with base flow $M = 0.5$ can be made to decay in roughly 50 or so grid points depending on the grid stretching ratio s as can be seen from figure 2.6, that is, it becomes evanescent.

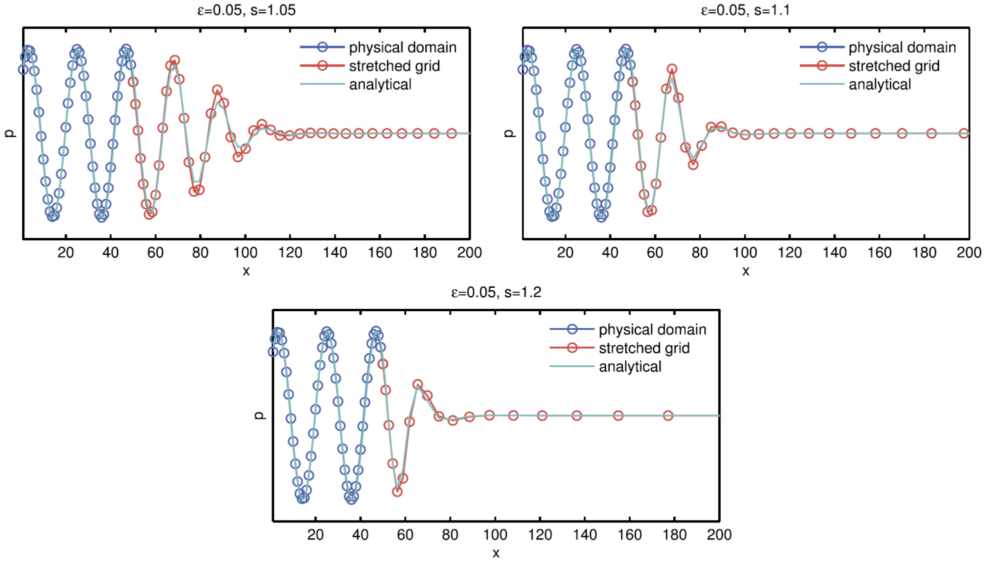


Figure 2.6: The decay rate for a one dimensional wave in domains of different grid stretching ratios. In each case the grid stretching starts at $x = 50$. The wave is travelling through a fluid of flow $M = 0.5$ where M is the Mach number. Graphs taken from [4]

This was also tested with constant s and changing artificial viscosity ϵ_x. In the 2 dimensional example flow is present in the x direction of base flow $M = 0.5$ and periodicity is enforced in the y direction with numerical viscosity turned off. Oblique waves are generated at the inflow boundary. Waves become cut-off, i.e. evanescent, when the distance between grid points becomes $h \geq 1/|\omega k|$, where ω is the temporal frequency and k is the wave number in the x direction. In the 2 dimensional setting the input wave was tested at three different wave angles. $\beta = 0°, 20°, 40°$ for the wave front relative to the x axis. This shows that waves hitting the buffer zone at angles up to 40° are well damped and become cut off at $h = 1/|\omega k|$. The paper concludes that the evanescent decay is very efficient in dampening the amplitude of physical waves, but the viscous damping is significant for high frequency spurious waves.

This boundary treatment produces good results in damping outgoing waves, however it requires additional domain space to be implemented and with additional numerical viscosity terms calculation times and compute resources are not as efficiently used as they could be with alternative methods such as the Perfectly Matched Layer (PML, see below).

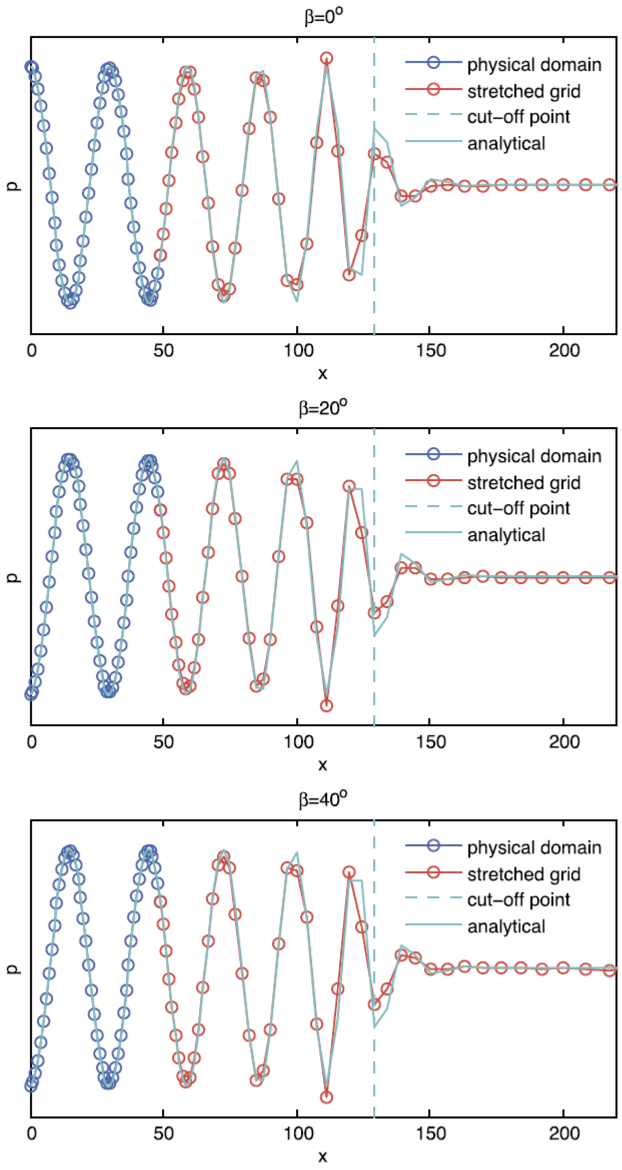


Figure 2.7: The decay rate for a two dimensional wave in a domain of stretching ratio $s = 1.05$ and different incidence angles β. In each case the grid stretching starts at $x = 50$. The wave is travelling through a fluid of flow $M = 0.5$ where M is the Mach number. Graphs taken from [4].

Vichnevetsky and Tuner [69] looked at wave propagation in irregular grids and also looked at the spurious scattering due to "sponges layers" in one dimension [70]. This type of boundary treatment is discussed in the following section.

2.6.3 Sponge Layer

Colonius et al [71], first proposed the concept of adding non-physical entrance and exit zones. This zone applied numerical filtering to damp disturbances in the exit zone thereby reducing spurius reflections. An advancement of this was done by Ta'asan and Nark [72] whereby they added convective terms to the linearised Euler equations causing them to be supersonic at the boundary of the numerical domain. Hu [73] extended a zonal boundary treatment developed by Berenger [74] for the Maxwell equations, to the Euler equations. This is what is now known as the sponge layer as in this method a damping term is added to the equations of motion in the entry and exit zones, the goal of which is to dampen the solution to a quiescent state. In summary a sponge layer is a numerical zone surrounding or trapping the area of interest that is designed to absorb outgoing waves, that is, it is an absorbing boundary layer, not a boundary condition. A boundary condition is still required when using sponge layers. These damping terms are "activated" at a reasonable distance away from the area of interest. In the general case the sponge layer is placed far enough away from the area of interest such that the effects produced by the simulation are linear. In acoustics however this is not necessary in most cases as acoustic effects typically tend to be linear, with the exception of certain phenomena such as shocks [5].

Using Mani [5], sponge layers will be discussed. The concept of the sponge layer can be summarised as follows, the Euler equations with additional "sponge" terms:

$$\begin{aligned} \frac{\partial u}{\partial t} + M\frac{\partial u}{\partial x} + \frac{\partial p}{\partial x} &= -\sigma(x)u \\ \frac{\partial v}{\partial t} + M\frac{\partial v}{\partial x} + \frac{\partial p}{\partial y} &= -\sigma(x)v \\ \frac{\partial p}{\partial t} + M\frac{\partial p}{\partial x} + \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= -\sigma(x)p \end{aligned} \tag{2.203}$$

Here the equations of motions are non-dimensionalised. The concept here is that after a certain distance, i.e. in the far field these sponge terms are added to help decay any waves that interact with them. Analysing this concept in one dimension the equations of motion reduce to:

$$
\begin{aligned}
\frac{\partial R}{\partial t} + (M+1)\frac{\partial R}{\partial x} &= -\sigma(x)R \\
\frac{\partial v}{\partial t} + M\frac{\partial v}{\partial x} &= -\sigma(x)v \\
\frac{\partial L}{\partial t} + (M-1)\frac{\partial L}{\partial x} &= -\sigma(x)L
\end{aligned}
\tag{2.204}
$$

where $R = u + p$ and $L = u - p$. Here all characteristics are uncoupled however with the sponge their amplitude decays exponentially. Using this sponge layer with a boundary condition that remains decoupled after the sponge, it is possible to obtain perfectly non-reflective boundary conditions. In the case of reflection three things may happen; an incident sound wave can reflect as a sound wave. A sound wave can reflect as a vorticity wave (for $M < 0$) and a vorticity wave can reflect as a sound wave (for $M > 0$). As a sponge Layer can be reflective, it is wise to set a damping target. The ratio of reflected sound to incident sound amplitude is given by:

$$
\eta \equiv \frac{|L_r|}{|R_i|} = \eta_{BC} \exp\left(-\frac{2}{1-M^2}\int_0^{l_{sp}} \sigma(x)\mathrm{d}x\right), \tag{2.205}
$$

where l_{sp} is the sponge length. Defining the damping target:

$$
\eta_t \equiv 20\log\left[\exp\left(-\frac{2}{1-M^2}\int_0^{l_{sp}} \sigma(x)\mathrm{d}x\right)\right] = -20\frac{2\log e}{1-M^2}\int_0^{l_{sp}} \sigma(x)\mathrm{d}x \tag{2.206}
$$

The derivation of which can be found in [5]. In the one dimensional example, the sponge performs ideally as it achieves its target damping whilst also not causing its own reflection.

Oblique waves will now be looked at. Setting $M = 0$ for simplification and Fourier transforming in the homogeneous directions (2.203) becomes:

$$
\frac{\mathrm{d}}{\mathrm{d}x}\begin{pmatrix}\hat{u}\\ \hat{p}\end{pmatrix} = (i\omega - \sigma)\begin{pmatrix}0 & 1 + \frac{l^2}{(i\omega-\sigma)^2}\\ 1 & 0\end{pmatrix}\begin{pmatrix}\hat{u}\\ \hat{p}\end{pmatrix}, \tag{2.207}
$$

where ω is the angular frequency and l is the wave number in the y direction, $l = 2\pi\sin\theta/\lambda$ where θ is the angle of the wave front normal relative to the x direction. Assuming constant

σ (2.207) can be diagonalized:

$$\frac{\mathrm{d}}{\mathrm{d}x}\begin{pmatrix}\mathcal{R}\\ \mathcal{L}\end{pmatrix} = (i\omega - \sigma)\begin{pmatrix}c & 0\\ 0 & -c\end{pmatrix}\begin{pmatrix}\mathcal{R}\\ \mathcal{L}\end{pmatrix} \tag{2.208}$$

where $c = \sqrt{1 + \frac{l^2}{(i\omega-\sigma)^2}}$ and $\mathcal{R}$ and $\mathcal{L}$ are right and left going waves respectively. Eigenvectors can be obtained from:

$$\begin{pmatrix}\hat{u}\\ \hat{p}\end{pmatrix} = \begin{pmatrix}c & -c\\ 1 & 1\end{pmatrix}\begin{pmatrix}\mathcal{R}\\ \mathcal{L}\end{pmatrix} \tag{2.209}$$

For $l \neq 0$ the eigenvectors depend on σ. As a result they respond to changes in σ. This causes reflection due to the discontinuous jump in σ as the sponge is encountered i.e. a right running wave will no longer remain a purely right running wave after interaction with the sponge layer.

The reflection ratio η in this case is given by:

$$\eta = \left[\frac{\theta^2}{4}\left(1 - \frac{\omega^2}{(\omega + i\sigma)^2}\right) + \frac{\theta^2}{4}\frac{\omega^2}{(\omega + i\sigma)^2}e^{-2l_{sp}\sigma + 2il_{sp}\omega}\right] \tag{2.210}$$

The first term measures the reflection due to the sponge interface at $x = 0$ whilst the second terms represents the reflection off the boundary $x = l_{sp}$. As the reflection due to the interface scales as θ^2 one can obtain a great improvement if they align the sponge with the outgoing wave fronts. In the case of large wavelengths the reflection coefficient becomes:

$$\lim \eta_{\omega\to 0} = \frac{\theta^2}{4} \tag{2.211}$$

which is identical to a reflection coefficient without a sponge. In the short wavelength limit the coefficient becomes:

$$\lim \eta_{\omega\to\infty} = \frac{\theta^2}{4}e^{-2l_{sp}\sigma} \tag{2.212}$$

which is equal to (2.205) when $m = 0$ and $\eta_{BC} = \theta^2/4$. In order to achieve this it is required that $\omega \gg \sigma$ which therefore leads to:

$$\frac{l_{sp}}{\lambda} \gg \eta_{target} \tag{2.213}$$

This requirement can be relaxed greatly if one optimises the sponge layer. In fact the best sponges are optimised. Looking at figure 2.8 it can be seen that an ideal performance can be achieved for a sponge larger than 10 wavelengths. The oscillations present in the right graph is due to interference in the waves between $x = 0$ and $x = l_{sp}$. Looking at the graph on the left it is seen that in the worst interference scenario, achieving an ideal performance for a sponge strength with $\eta_{target} = -60\text{dB}$ requires a sponge length of 10^4 wavelengths. This ineffective behaviour can be resolved by introducing non-uniform sponges.

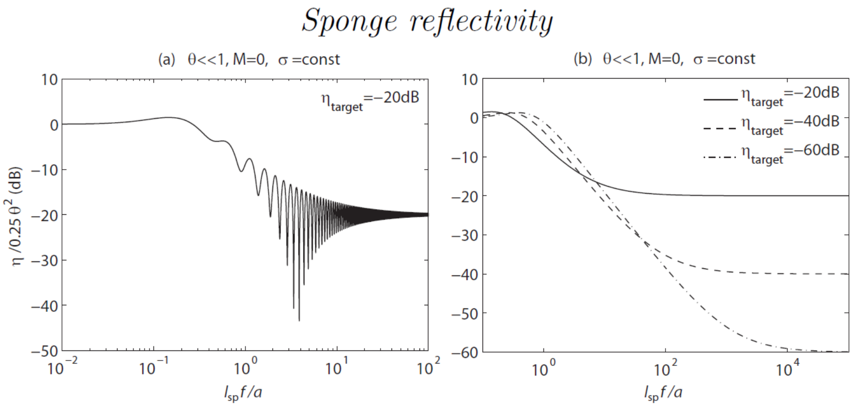


Figure 2.8: These graphs taken from [5] show the reflectivity as a function of sponge length (dimensionless) with constant σ and $\eta_{target} = -20\text{dB}$ (right) and the upper envelop of this profile is shown and compared with sponges of various strengths (left).

A non-uniform sponge is developed as such; discretise the sponge profile and assume a constant σ in each interval Δx. The same steps previously taken can then be taken for each Δx. Considering power law profiles ($\sigma(x) \sim x^n$ with $n = 0, 1, 2, 3$) one can see from figure 2.9 that the rate to convergence to ideal performance is much quicker for sponges of higher order when compared to that of a constant sponge. This is because as σ is gradually increased the interfacial reflection is distributed over the length of the sponge.

Moving over to the general analysis with non-constant σ, nonzero M and general θ, (2.203) the equations of motion are:

$$\frac{\mathrm{d}}{\mathrm{d}x}\begin{pmatrix}\hat{u}\\ \hat{v}\\ \hat{p}\end{pmatrix} = \begin{pmatrix}M & 0 & 1\\ 0 & M & 0\\ 1 & 0 & M\end{pmatrix}^{-1}\begin{pmatrix}i\omega-\sigma(x) & 0 & 1\\ 0 & i\omega-\sigma(x) & -il\\ 0 & -il & i\omega-\sigma(x)\end{pmatrix}\begin{pmatrix}\hat{u}\\ \hat{v}\\ \hat{p}\end{pmatrix}. \tag{2.214}$$

Discretising and applying a non-uniform sponge layer, (2.214) can be diagonalized and

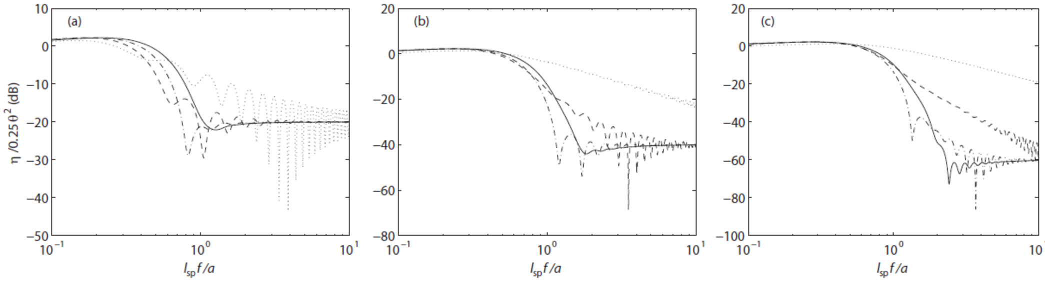


Figure 2.9: These graphs taken from [5] show the effectiveness of non-uniform sponge profiles of the form $\sigma(x) \sim x^n$ with $n = 0$ (dotted line), $n = 1$ (dashed line) $n = 2$ (dashed-dotted line) and $n = 3$ (solid line).

solved to obtain:

$$\begin{pmatrix} \hat{u}\left(x+\Delta x\right) \\ \hat{v}\left(x+\Delta x\right) \\ \hat{p}\left(x+\Delta x\right) \end{pmatrix} = S_x e^{\Lambda \Delta x} S_x^{-1} \begin{pmatrix} \hat{u}(x) \\ \hat{v}(x) \\ \hat{p}(x) \end{pmatrix} \tag{2.215}$$

where Λ is the matrix of local eigenvalues and S is the matrix of local eigenvectors. A common choice used for the boundary at $x = l_{sp}$ is:

$$\hat{u}(l_{sp}) - \hat{p}(l_{sp}) = 0. \tag{2.216}$$

In the general case, non dimensional sponge length (normalised by frequencey, f and sound speed c) is no longer equal to the ratio of sponge length to the wavelength. The relationship between the frequency and wave number k is:

$$\omega = k\left(1 + M\cos\theta\right) \tag{2.217}$$

$$\omega = kM\cos\phi \tag{2.218}$$

for sound waves and vorticity waves respectively. The angles θ and ϕ are angles at which the wave hits the sponge relative to the x direction. When l_{sp} is small, the sponge is essentially useless but when $l_{sp}f/a$ is of order 0.5 to 2 for the case where acoustic waves reflect as acoustic waves results improve. This was tested in [5] at an angle of incidence ranging from 0 to $\cos^{-1}(-M)$. In his testing a wide range of frequencies are damped by ratios that exceed the set target.

In some cases where $M < 0$ acoustic waves can reflect back into the physical domain as vortical waves, however as frequency increases so does the effectiveness of the sponge so this is not much of a concern and this is also the case for when vortical waves reflect as sound.

Sponges offer a good solution but as mentioned take up more computing power than the NRBC's mentioned above. Sponge layers are also required to be put out into the far field and the sponge itself will take up space in the domain contributing to the requirement of more compute power and calculation time. Bodony [75] also found that applying a sponge to the linearised Euler equations can cause the solution to lose stability when being damped. This was avoided by using the same sponge strength for each equation. He found that the rate of convergence to the target was the inverse of the sponge strength.

2.6.4 Perfectly Matched Layer

A perfectly Matched Layer (PML), first introduced by Berenger [74], may be considered an alternative form of a sponge layer. Developed for computational electrodynamics [76, 77] the method has been adopted to situations with flow making it usable in acoustic problems with a mean flow. The concept of the Perfectly Matched Layer (PML), turns the boundary condition from a 2 dimensional “shell” enveloping the area of interest into a shell with 3 dimensions, that is, a layer. Again this is an absorbing boundary layer, with an additional boundary condition. A basic review on PML's can be found in [6, 78]. The issue with this layer type boundary treatments is that when transitioning from one medium to another reflections still occur. It is possible to construct an absorbing medium, however, that is not reflective at the interface. This is what makes a PML. Most modern PML's use a formulation called uniaxial PML which is a formulation that involves expressing the PML as the wave equation with a combination of artificial, anisotropic absorbing materials. PML's also work for inhomogeneous media, as long as the medium is homogeneous in the direction perpendicular to the boundary.

It is worth mentioning that PML's can help to accelerate the decay of evanescent waves. Doing this however will in turn cause propagating waves to oscillate even faster meaning tuning is required on the part of the engineer.

One can think of a PML as a complex coordinate stretching. The first step is to think of some solution to some wave like equation:

$$\boldsymbol{u}\left(x,t\right)=\sum_{k,\omega}\boldsymbol{U}_{\mathbf{k},\omega}e^{i(\mathbf{k}\cdot\mathbf{x}-\omega t)} \tag{2.219}$$

As usual k and ω are the wave number and angular frequency respectively. The key here is to evaluate (2.219) on complex x where the contour on which the evaluation takes place enters the complex space some place outside the area of interest. Splitting the exponential part $e^{ik_r\,\mathrm{Re}\,x}e^{-k_i\,\mathrm{Im}\,x}$ it is seen in figure 2.10 that the solution decays as $\mathrm{Im}\,x$ increases as if it is in some sort of absorbing material. As the solution itself has remained unchanged the PML acts like a reflectionless absorbing material.

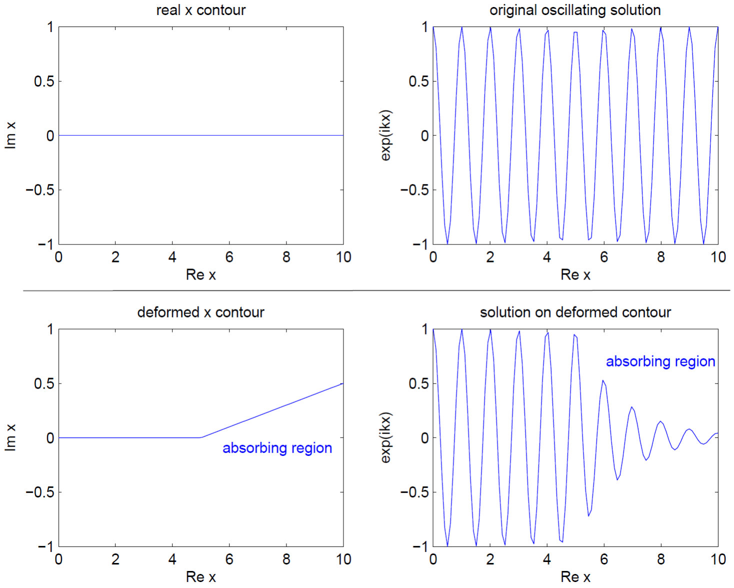


Figure 2.10: Taken from [6] the above graphs show a solution oscillating through real x while the bottom graphs show the solution decay as imaginary x is slowly introduced.

To spare one from the difficulty of solving differential equations along a complex contour it is best to coordinate transform back to real x. This is done via writing the differential equation in terms of the real part of x. Writing x as $x(x_r)=x_r+if(x_r)$, where $f(x_r)$ is the function that deforms the contour, it is seen that:

$$\begin{aligned} \partial x &= \left(1 + i\frac{\mathrm{d}f}{\mathrm{d}x_r}\right)\partial x_r \\ \rightarrow \quad \frac{\partial}{\partial x} &= \frac{1}{\left(1 + i\frac{\mathrm{d}f}{\mathrm{d}x_r}\right)}\frac{\partial}{\partial x_r} \end{aligned} \tag{2.220}$$

That is, PML can be seen simply as a transformation of the differential equation. Taking $\frac{\mathrm{d}f}{\mathrm{d}x_r} = \frac{\sigma_{x_r}(x_r)}{\omega}$ for some function $\sigma_{x_r}(x_r)$, where the factor $1/\omega$ is given to ensure all wavelengths decay at equal rates. In the above example a PML in the positive x direction has been established. In the negative x direction the exact same steps are taken and this is the case for the y and z directions also. A more detailed formulation of the PML can be found in [79–83].

This formulation of the PML is such that the attenuation rate in the PML is independent to frequency ω i.e. all wavelengths decay at the same rate. This makes PML suitable for broadband noise as well as tonal noise.

Once the PML has been established truncation needs to occur with the boundary conditions usually being a constant pressure boundary condition. Reflections from the boundary condition in this case should not be an issue as any reflected waves decay on the way back. The next question is how thick or thin must a PML be? Analytically it is possible to make it as thin as one desires simply by choosing a large σ_{x_r} however numerically large σ_{x_r} will cause reflections. In fact once discretising occurs reflections cannot be helped. The reflections however should be small if there is a good approximation to the wave equation. In order to better control the reflections one can slowly turn on the PML quadratically or cubically making said reflections negligible. This will cause the PML to be of a thickness roughly half of that of a single wavelength.

PML also requires a large enough domain such that angle of incidence of the wave onto the PML becomes less of a factor. In developing the PML in 1 dimension, direction was not a factor however in 2 and 3 dimensions angle of incidence does indeed impact the effectiveness of the PML. The wave number k was discussed however it would be more correct to say that the decay rate is proportional to $|k|\cos\theta$. As the angle of incident increases the decay rate decreases. A large enough domain however can help to ensure the angles of incidence stay such that an acceptable decay rate is obtained.

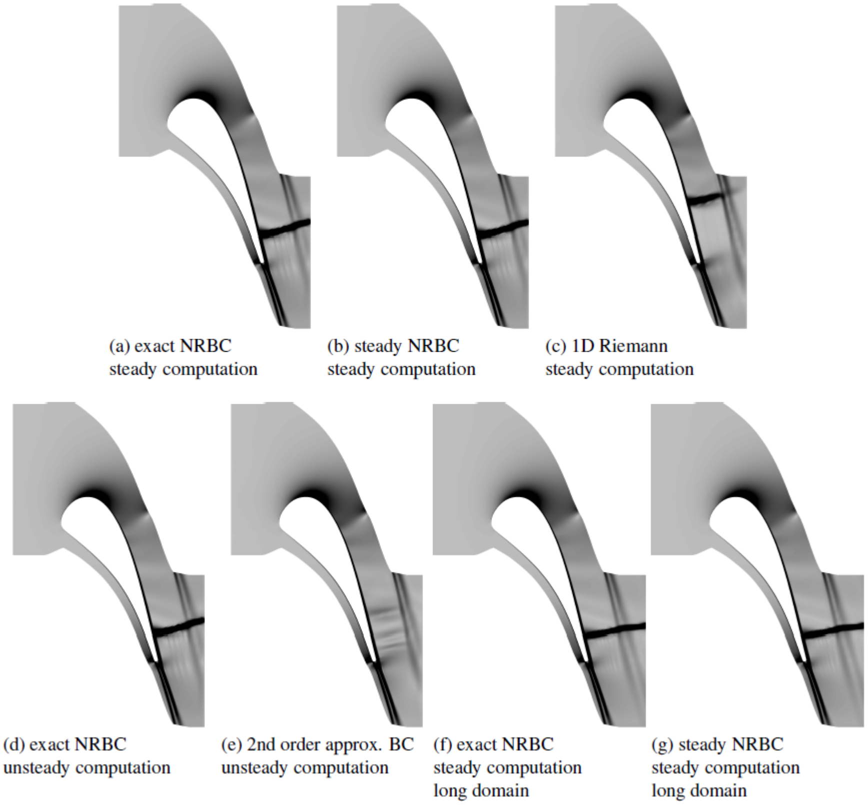
Whilst PML's offer a good solution for absorbing boundary conditions for acoustic problems, they still require a dedicated area in the mesh surrounding the area of interest, in order to be effective. This once again takes up computing power and time. One must also carefully apply PML's to ensure reflections are negligible and this can cause the area dedicated for PML's to grow cubically in 3 dimensions, causing, again the same problem.

2.7 NRBC Implementation and Tests Cases

Frey et al [7] tested a range of NRBC's for steady and unsteady flow simulations, in the context of turbomachinery, in duct, showing promising results in a Reynolds-averaged Navier-Stokes equations (RANS) Computational Fluid Dynamics (CFD) code. He uses the same formulation as that of Giles deriving the NRBC's using the same principles. Uniform entropy and enthalpy was enforced in the steady case while setting ω to 0. In the unsteady case local boundary conditions were formulated by adopting the approach developed by Engquist and Majda [14]. Multiple types of NRBC's were applied to a 2 dimensional, linear turbine cascade and compared:

- Exact NRBC (Steady)
- Steady NRBC (Steady)
- Exact NRBC (Unsteady)
- Exact NRBC long domain (Steady)
- Steady NRBC long domain (Steady)
- 1D Riemann (Steady)
- 2nd order approximate boundary conditions (Unsteady)

The calculations were carried out at a supercritical operating point with stagnation temperature being 415k and stagnation pressure at the inlet at 147500 Pa and 78000 Pa static pressure at the outlet with a base frequency set to 6.1 kHz which is of the order of magnitude of typical of turbomachinery flows. 64 times steps are taken to resolve one period, with 25 sub iterations between each timestep. Figures 2.3 a, b, d, agree with the reference results figures 2.3, f and g. It is shown the steady NRBC's and the exact NRBC's in both steady and unsteady computations agree qualitatively with the long domain results taken to be reference results. Looking at the graph in figure 2.12 it is seen that the 2nd order approximate boundary condition and the Riemann Boundary condition both smear out the shock leading to a differing pressure distribution on the suction side. The pressure distributions of all other methods match, however it should be noted that all short domain computations deviate slightly from the reference results produced by the long domain calculations.



(a) exact NRBC steady computation

(b) steady NRBC steady computation

(c) 1D Riemann steady computation

(d) exact NRBC unsteady computation

(e) 2nd order approx. BC unsteady computation

(f) exact NRBC steady computation long domain

(g) steady NRBC steady computation long domain

Figure 2.11: Taken from [7]. This shows plotting density magnitude.

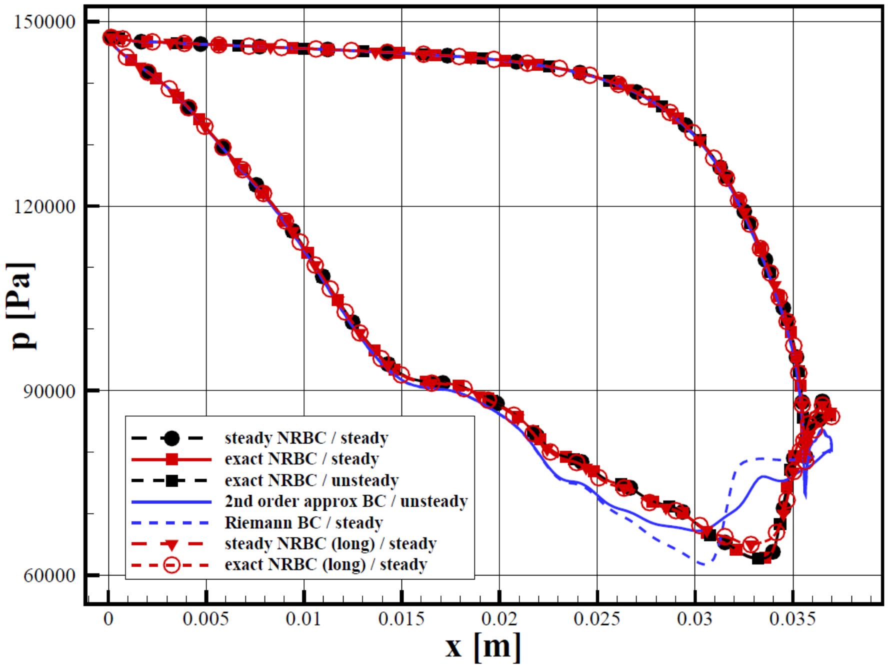


Figure 2.12: Taken from [7]. This shows blade pressure distribution.

Bertsch [8], evaluated and tested the NRBC's present in ANSYS FLUENT. The NRBC's tested here in FLUENT are implemented using characteristic waves which are calculated from the primitive variables; pressure, temperature, and the velocity components in all 3 dimensions. They were tested with laminar pipe flow with an NRBC set up on the outflow boundary condition with standard slip walls. It was tested via an input broadband signal ranging from 0kHz to 5 kHz. The effectiveness of the NRBC was studied via analysing the reflection coefficient. For frequencies between 0.5 - 5 kHz the reflection coefficient, $\left|\hat{R}\right|$ drops to values below 0.05. The NRBC for lower frequencies however fails as the reflection coefficient approaches 1 as the frequency tends to 0. NRBC's were also tested for turbulent pipe flow with the boundary condition once again at the outlet this time with no slip walls. This introduces an element of non-uniform flow, which is important for the scope of this project. The results here are once again promising for higher frequencies. For frequencies going above 0.6 kHz the reflection coefficient reaches values below 0.05 but as the frequency tends to 0 the coefficient goes to 1 showing once again more promise at higher frequencies. The authors therefore state that the method is effective over the 0 - 5000Hz range that they were considering, but note that this was for a lab scale duct with a diameter of 30mm. The majority of this range of frequencies can therefore accurately studied.

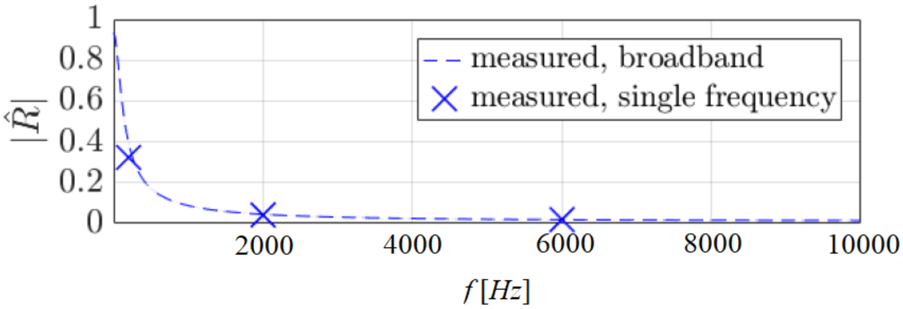


Figure 2.13: Taken from [8]. This graph shows the reflection coefficient determined at the centre of the outflow boundary for laminar pipe flow. Here the x axis shows frequency

Anker [9] tested his NRBC's in a few examples the first of which was subsonic flow through a linear turbine cascade. The performance was assessed by comparing solutions calculated on domains of different boundary treatments, mainly NRBC and large domain as shown in 2.15. The reference solution was the solution calculated using the 3D-NRBC on the largest domain. The flow is viscous and isentropic and tested at three different outflow Mach numbers: $M = 0.62, 0.81, 1.21$. The results obtained show that all three variants of the

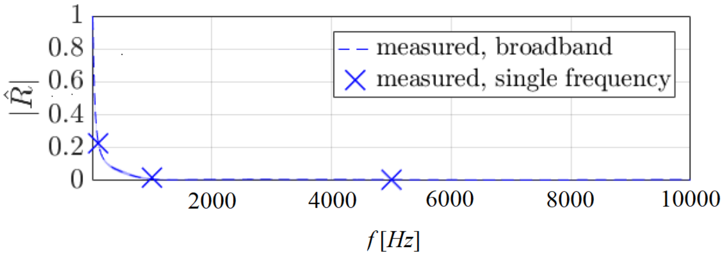


Figure 2.14: Taken from [8]. This graph shows the reflection coefficient determined at the centre of the outflow boundary for turbulent pipe flow. Again the x axis shows frequency

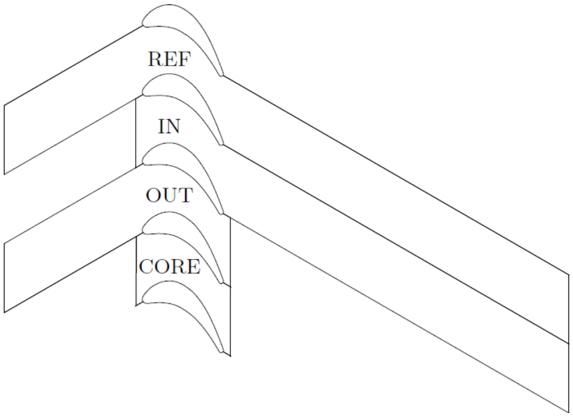


Figure 2.15: Taken from Anker [9]. This image demonstrates the different boundary treatments. REF shows the reference large domain treatment. IN uses an inlet boundary treatment and OUT uses an outlet boundary treatment while core uses a treatment at both inlet and outlet.

3D-NRBC match. The solutions obtain from the “IN” mesh are in agreement for all three Mach numbers, whilst the solutions obtained from the “OUT” mesh are slightly deviated from the reference solution for the subsonic Mach numbers however the supersonic Mach number deviates more considerably. This is due to a shock crossing the outlet boundary, and hence can be attributed to linearisation errors as a shock is no longer a linear effect.

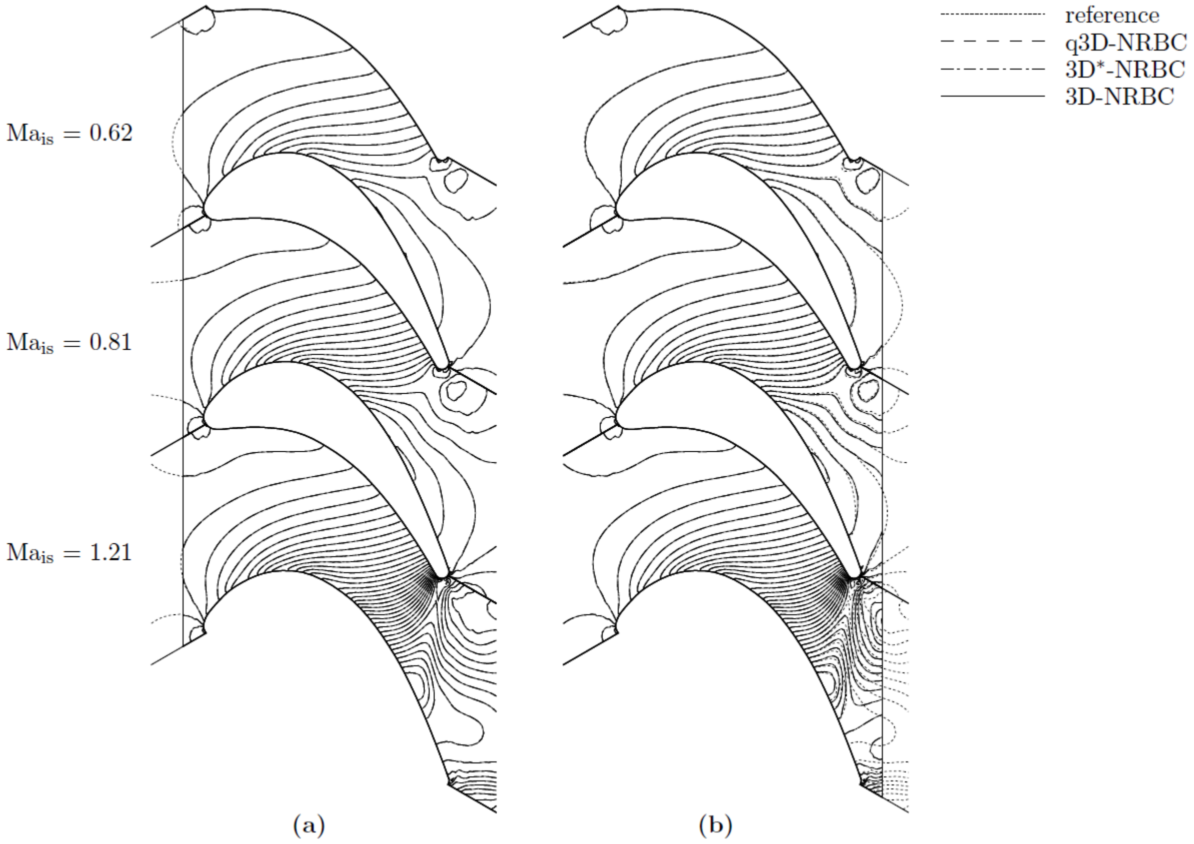


Figure 2.16: Taken from Anker [9]. This figure shows the solution resulting aforementioned boundary treatments.

Medida [10] tested his 2 dimensional generalised curvilinear boundary conditions against a Ringleb flow example. He tested his NRBC's with different base flows of Mach numbers ranging from $M = 0.1 - 0.35$ at the inflow and $M = 0.52 - 1.20$ at the centre of the inner wall, on meshes of different resolutions. The Cartesian NRBC's were tested without the wall condition that Medida had also developed. This leads to flow through the wall. Applying the wall corner condition the flow through the wall vanished however the correct flow solution at the corners could still not be achieved causing the solution to diverge.

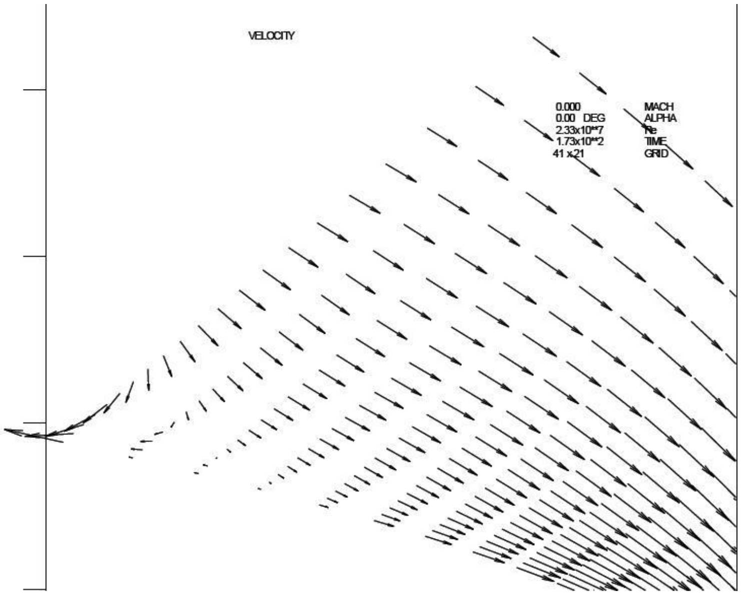


Figure 2.17: Taken from Medida [10]. Here the solution presented is based off of boundary conditions that do not include the aforementioned additional terms and also do not implement the wall condition.

Applying now the Curvilinear NRBC's without the wall condition, Medida found it best to derive the curvilinear boundary conditions from the curvilinear form of the Euler equations as opposed to deriving the Cartesian boundary conditions and transforming them. In doing so additional terms were found to be part of the equation, which play a crucial role. Applying the curvilinear NRBC's without the additional terms leads to solutions similar to that demonstrated in figures 2.17 and 2.18. When applying the boundary conditions with the additional terms and the wall condition turned on the correct flow solution is observed.

The solution displayed long term instability which was cured through grid stretching towards the inflow and outflow boundaries. Medida found that in this case grid stretching breaks any diverging feedback loops and damps unresolved frequencies.

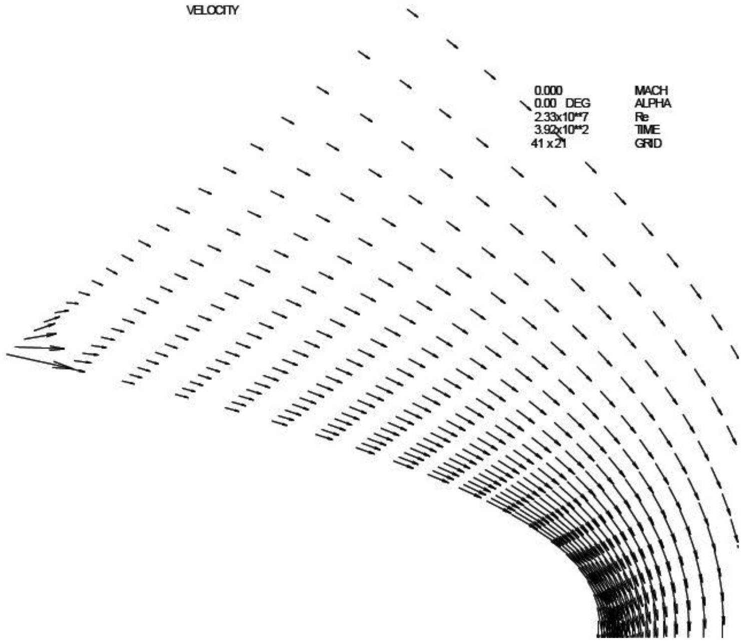


Figure 2.18: Taken from Medida [10]. Here the additional terms are still lacking however the wall condition has been implemented.

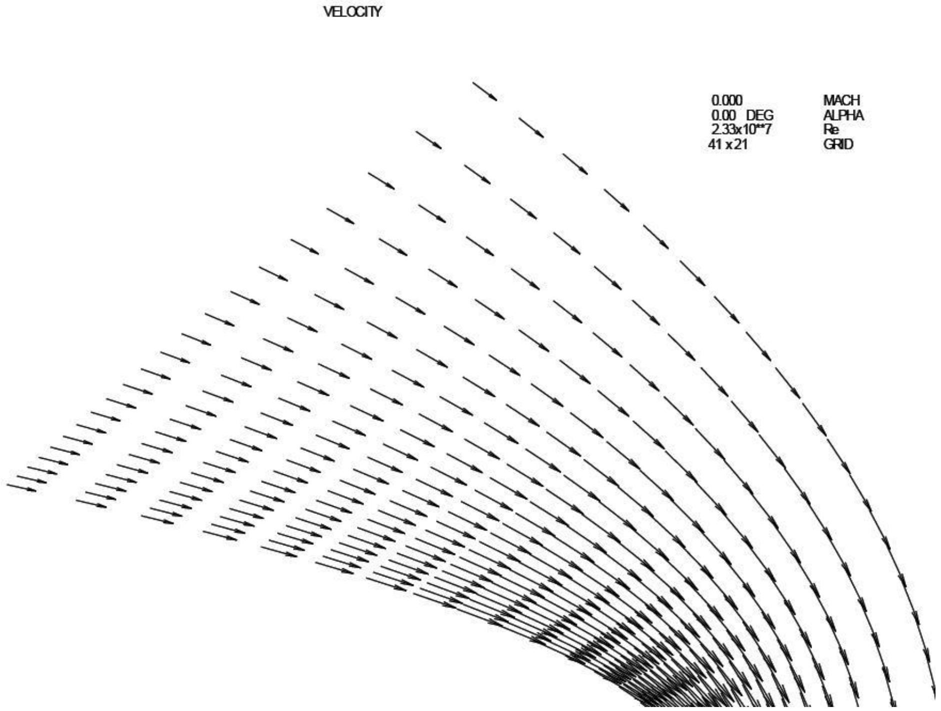


Figure 2.19: Taken from Medida [10]. This solution has both the additional terms along with the wall condition.

Chapter 3

Methodology and Application

This chapter will be divided into two parts. The first will describe the methodology and application of the EBNRBC and its implementation into the ANSYS FLUENT solver. The second part will discuss the implementation of the test cases described in chapters 4, 5 and 6.

3.1 EBNRBC Methodology and Implementation

3.1.1 EBNRBC Methodology

The eigenvectors and eigenvalues of the equation below will be found and used to formulate the EBNRBC's:

$$A\frac{\partial \hat{f}}{\partial \zeta} = B\hat{f}. \tag{3.1}$$

The first step is finding the eigensolutions of (3.1) along surfaces of constant ζ. The use of curvilinear coordinates during the analysis ensures the effects of the curvature, gradients in the mean flow amplitude and direction are captured. In weakly to moderately non-uniform mean flow, the eigenvalues and eigenvectors derived from this equation relate to mainly vortical and mainly acoustic modes in a similar manner to the uniform flow case. The eigenmodes, however, are no longer simple circumferential harmonics. To illustrate this, figures 2.1 and 2.2 show the effect of non-uniform flow on the shape of the pressure profile in the case of a Gaussian mean flow profile around the circumference described in section 2.2.6 for the acoustic mode most similar to the fifth circumferential first radial Bessel-Fourier mode in uniform flow. The effect of the non-uniform mean flow is to spread the peaks non-uniformly around the circumference, which in Fourier space is represented by the the content spreading into neighbouring harmonics.

The eigenmodes are classified in terms of direction of information travel. Wilsons duct propagation method [27] requires a more-or-less complete classification of modes, but in the present method it is only necessary to identify the subset of outgoing modes to which the NRBC is to be applied. In weakly to moderately non-uniform duct geometry and mean flow, including all realistic duct geometries considered to date, this can be done relatively easily by comparing the eigenvalues and eigenvectors with those of standard acoustic and vortical uniform duct modes in uniform flow.

Formulating the required boundary conditions requires the left and right eigenvectors at the boundary. Each mean flow profile and frequency requires a different set of eigenvectors. The left eigenvectors can be written together to form a matrix which can be multiplied by the flow variables to produce a vector of complex coefficients for each eigenmode. In non-uniform mean flow, as in uniform flow, the complex coefficients of an individual mode multiplied by the right eigenvector gives the full velocity and pressure profile for that specific component of the unsteady flow.

Before starting some geometric identities are required. First is the Jacobian matrix J which relates the rate of change of each coordinate of the (ξ, η, ζ) body fitted mesh to the rate of change of coordinates in the Cartesian coordinate system (x, y, z),

$$J = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{pmatrix} \tag{3.2}$$

The metric tensor g_{ij} required to cast the flow equations into tensor form as described in section 2.2.7 is defined as:

$$g_{ij} = J^T J \tag{3.3}$$

Implementation of the EBNRBC requires first a preprocessing stage illustrated by the left hand path in the flow chart in figure 3.1 before applying them to the unsteady calculation illustrated by the right hand path.

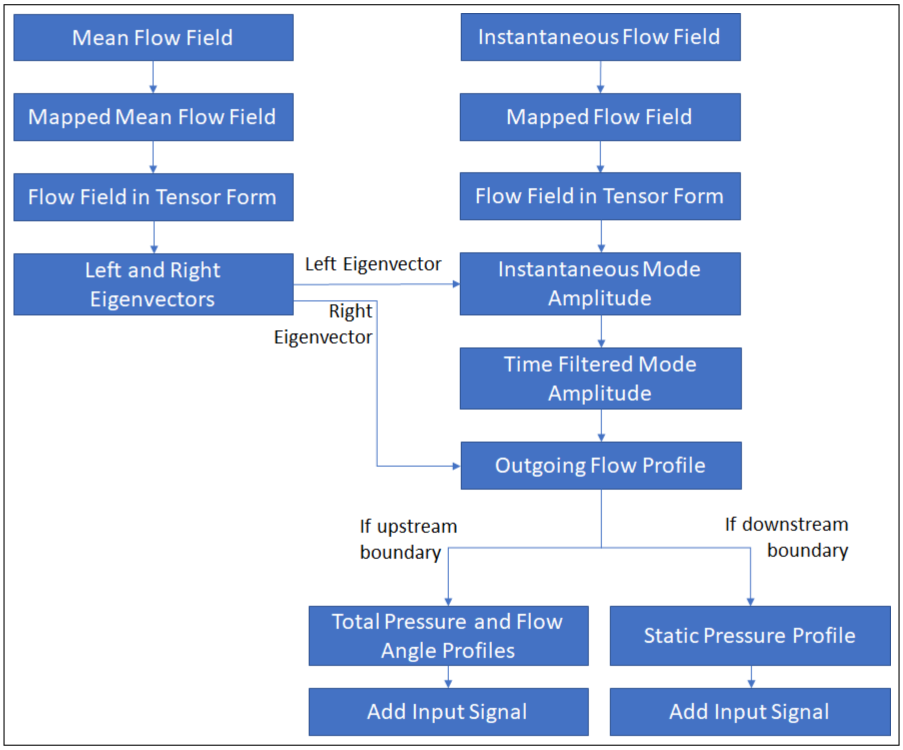


Figure 3.1: Flow chart showing the implementation of the method for a single mode at a single frequency.

Looking first at the preprocessing stage:

- A steady flow field is required for the calculation of the right and left eigenvectors. In turbomachinery CFD simulations this is often calculated as a preliminary solution prior to starting the unsteady calculation.
- In the general case the mean flow data has to be mapped from the calculation mesh to the body fitted mesh. In the test cases presented here the CFD mesh was chosen to be coincident with the body fitted mesh and this step was not required.
- The physical velocities are transformed to tensor form by premultiplying with the transpose of the Jacobian matrix. It's also transformed into a spectral representation into the pseudo-circumferential direction using a truncated FFT.
  The test cases presented in chapters 4, 5 and 6 use only circular duct geometries, where the Jacobian is very simple and effects only the θ component of the velocity. In these cases the transformation into tensor form was done directly.
- Left and right eigenvectors are calculated as described by Wilson [27] and discussed in detail in sections 2.2.6 and 2.2.7.

Only the modes of interest are extracted and the left and right eigenvectors saved for use in the unsteady boundary condition calculation. In uniform mean flow the main modes are Tyler-Sofrin modes [85]. In the general case, all cut-on modes with significant content should be considered. In the specific case of the aeroengine fan stage all of the cut-on Tyler-Sofrin modes should be included at all harmonics of rotor blade passing frequency.

In principle with non-uniform mean flow it is necessary to include all modes at harmonics of rotor blade passing frequency for a general calculation however in practice it is more efficient to identify a subset of those modes where it is likely that there will be significant modal content. It is often possible to do this in analogy with the uniform flow case by identifying modes similar to Tyler-Sofrin modes recognising that there will be some scattering into neighbouring circumferential harmonics by the non-uniform mean flow and geometry.

Looking now at the unsteady calculation (right path) we have:

- The instantaneous flow field which refers to the current state of the unsteady CFD calculation.
- This to has to be mapped to the EAGCC body fitted mesh and converted to tensor form in the same manner as the steady flow described previously.
- Because the method of Wilson uses a spectral method in the pseudo-circumferential direction a Fourier transform is applied in this direction before applying the left eigenvector to give the instantaneous mode amplitude.
- The actual mode amplitudes at the required frequency are obtained via the use a numerical bandpass filter. A phase correction is applied to account for the response time of the numerical filter and also the fact that the boundary condition uses flow variables from the previous timestep. In the current work a fourth order Butterworth filter was used further discussed in section 3.1.2.
- The outgoing flow profile is calculated by multiplying the complex mode amplitude by the right eigenvector and converting back to physical quantities. This is done using the inverse of the Jacobian matrix and an inverse Fourier transform in the pseudo-circumferential direction. The flow profile is then mapped back to the CFD mesh.
- In many solvers (including ANSYS FLUENT as used here) boundary conditions can only be implemented in certain ways. In the test cases presented here, the unsteady flow profile related to the outgoing wave was mapped to a static pressure boundary condition at the downstream end and a total pressure inflow boundary condition at

the upstream end.
In the case of a downstream travelling wave only the static pressure profile is required and in the case of an upstream travelling wave the instantaneous total pressure and flow angles are retained and input into the standard boundary conditions. The instantaneous total pressure was calculated from the static pressure and flow Mach number (including both the steady state and unsteady contributions). The calculation was then carried out as if the flow was steady state as seen in section 3.2. This is because ANSYS FLUENT treats the input total pressure as if the flow were in steady state.

- In either case an input signal can be added to the out going wave information as was done for the test cases presented later. This input signal typically includes the steady boundary data along with an unsteady acoustic or vortical input wave.

This approach allows low circumferential orders to be treated without affecting higher circumferential modes for which the CFD mesh is insufficiently fine. These high circumferential order modes are typically cut off at the frequencies of interest and so do not play a significant role in the prediction. Acoustic waves at other frequencies are filtered out and left untreated.

3.1.2 Filtering

The time filtering was done using a fourth order Butterworth bandpass filter. The Butterworth filter was chosen due to its flat amplitude response and smooth phase response through the passband. In cases with closely spaced frequencies an alternative filter may be preferred such as the Chebyshev filter as this has a steeper roll off from the passband.

The filter script used requires as input the angular frequency ω and the timestep used. In practice the timestep has to be chosen to minimise discretisation errors in the solver, for example in the two dimensional test cases described in chapter 4 the number of timesteps was chosen to be 200 per period and in the 3 dimensional case discussed in chapter 5 this was increased to 400 per period. From this the required inputs to the filter were obtained. The Nyquist frequency is defined as half the sampling rate of the signal:

$$f_N = \frac{1}{2}sf \tag{3.4}$$

where s is the number of timesteps. The Butterworth filter was generated in MATLAB.

The Butterworth function used in the script requires as input a normalised frequency band and a defined order of accuracy, the higher the order the higher the accuracy. The order of the filter is $2n$ where n is the chosen input value in this case is $n = 2$. The butterworth filter function in MATLAB outputs what is known as the transfer function coefficients a and b. The number of elements in a and b is $2n + 1$. The filtered time signal at each timestep is given by:

$$u = \begin{pmatrix} b_1 \\ ... \\ b_5 \end{pmatrix} \begin{pmatrix} \mathbf{u}_{ts}, & ... & , \mathbf{u}_{ts-4} \end{pmatrix} - \begin{pmatrix} a_2 \\ ... \\ b_5 \end{pmatrix} \begin{pmatrix} \mathbf{u}_{ts-1}, & ... & , \mathbf{u}_{ts-4} \end{pmatrix} \tag{3.5}$$

Here $\mathbf{u}_{ts}$ is the FLUENT Fourier transformed variables at the outlet at a certain timestep to be filtered while ts represents the signal at a certain timestep while $ts-1$ represents the timestep prior and so on. It is with these filtered variables that the boundary conditions are developed. It should be mentioned that the use of a filter introduces errors in the phase which need to be corrected for.

The width of the bandpass had a normalised frequency range of 0.2. The narrower the bandpass the longer the response time of the filter and so wider bandpasses were tested and this is discussed below in chapter 5. The calculation also seemed to have convergence times that depended on how close the input signal was to being cut off however this behaviour was only present in the two dimensional calculations and is again discussed below.

3.1.3 Implementation into a Commercial CFD Code

Implementation of the new boundary condition requires run time read/write access to flow variables at the domain boundaries. In the commercial code chosen (FLUENT Ver. 19.4) this was achieved through the use of inbuilt macros within the framework of a User Defined Function (UDF). The UDF is coded in the programming language C.

The code makes use of 3 different types of FLUENT macro. FLUENT's DEFINE ON DEMAND was used to set up the boundary conditions in advance of the unsteady calculation. The DEFINE AT END macro was used to determine the amplitude and phase of the outgoing modes as described in section 3.1.1. The DEFINE PROFILE macro was used to set the values of the flow parameters at the boundary.

The first task of the UDF was to read in the left and right eigenvectors obtained from the EAGCC method and use them to formulate the correct NRBC's at the inlet and outlet. The eigenvectors were read in using standard C read functions that were placed under FLUENT's DEFINE ON DEMAND macro. This allows the user to run the code on demand, in this case at the start, before the calculation has started.

FLUENT allows direct read and write access to the flow variables in the pressure based solver, however in the density based solver, the chosen solver, pressure is inaccessible at the boundary. Instead the pressure set by the NRBC in the previous iteration was stored is User Defined Memory and used alongside the velocity in the current timestep. The pressure and velocity perturbation from the mean flow was Fourier transformed and converted to tensor form as described in section 3.1.1.

The Fourier transformed variables were then run through a numeric bandpass butterworth filter as described in section 3.1.2 to ensure only the correct frequency was input into the domain. The filter also took the form of a void function.

Once the perturbed flow variables had been filtered for the correct frequency they were multiplied by the left eigenvector via a dot product to produce the complex coefficient for each outgoing mode. A phase correction was included to account for the phase lag introduced by the filter. The corrected coefficient was then multiplied by the right eigenvector creating a new outgoing vector ready to be applied at the relevant boundary.

All of the above functions were carried out within a DEFINE AT END macro such that they were carried out once at the end of each timestep.

It should be noted that generally the left and right eigenvectors are complex values. As a result a C header file was created. This header file defined a new data type “complex” and defined the functions necessary for basic complex mathematics such as multiplication, addition, subtraction and division.

The pressure and velocity parts of the outgoing modes were transformed back into physical rather than tensor form and returned to normal circumferential coordinates from spectral form as described in section 3.1.1, before being applied at the outlet/inlet forming the correct boundary condition. The pressure and velocity profiles were coded and placed under FLUENT's DEFINE PROFILE macro. The inlet boundary conditions require the total pressure and flow angles to be set while the outlet boundary condition requires only the static pressure to be set.

3.2 Methodology Related to the Test Cases in Chapters 4, 5 and 6

3.2.1 Implementation of the Two Dimensional Test Cases

This section covers the background theory for the test cases present in chapter 4. This includes the calculation of the cut-on ratio for a two dimensional wave, from the frequency and the circumferential mode number and also includes the calculation of the steady state boundary values for temperature and pressure.

In two dimensions the cut off ratios were found using equation (2.62) which for recollection is:

$$\mu^2 = (\tilde{\omega} + Mk)^2 - k^2 \tag{3.6}$$

Rearranging this to find the axial wave number k:

$$k = \frac{\tilde{\omega}M \pm \sqrt{\tilde{\omega}^2 + (\mu^2 (M^2 - 1))}}{(M^2 - 1)} \tag{3.7}$$

k should be real for propagating waves and imaginary for cut-off waves. As a result attention is focused on the argument of the square root. In two dimensions μ^2 is simply m^2/R^2 where R^2 is the radius of the thin annulus duct on which the calculation was performed, a basic derivation of which can be found in section 2.2.5. The cut off frequency is then defined as the frequency where the term under the square root passes through zero as shown by:

$$\tilde{\omega}_c^2 = \mu^2 \left(M^2 - 1\right) \tag{3.8}$$

A number of ratios were then chosen and the corresponding frequencies calculated:

$$\tilde{\omega} = \tilde{\omega}_c \upsilon \tag{3.9}$$

where υ is the required ratio. Acoustic modes of these frequencies were then used as input modes to validate the use of the EBNRBC in two dimensions. Choosing a constant number of timesteps per period meant the filter coefficients did not have to be recalculated.

Looking now at the steady state boundary conditions. The correct pressure for uniform mean flow at the inlet for the chosen Mach number was found using standard isentropic flow equations:

$$\frac{p_s}{p_t} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{-\gamma}{\gamma - 1}} \tag{3.10}$$

where p_s and p_t are the static and total pressures respectively.

For non-uniform mean flow the mean flow profile was defined by a Gaussian deficit that shifted the Mach number from 0.5 to 0.2 and back again circumferentially. This was done via finding the required pressure at Mach number 0.5 and 0.3 using equation 3.10, multiplying the pressure at $M = 0.3$ by a Gaussian distribution and subtracting it from the pressure at $M = 0.5$:

$$p_{nt} = p_{mt} - p_{dt} e^{-\theta^2 / 2\sigma^2} \tag{3.11}$$

where p_{nt} is the non-uniform total pressure, p_{mt} is the max total pressure and p_{dt} is the total pressure difference between the maximum total pressure and the minimum total pressure, θ is the circumferential angle and σ is the variance of the Gaussian. As the total pressure now varied significantly it was a requirement to also correct for the temperature. The static temperature was found using standard isentropic flow equations:

$$\frac{T_s}{T_t} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{-1} \tag{3.12}$$

at $T_t = 300k$. The total temperature shift around the circumference was then found using the relation:

$$\frac{T_s}{T_t} = \left(\frac{p_s}{p_{nt}}\right)^{\frac{\gamma - 1}{\gamma}} \tag{3.13}$$

The test cases can be considered as an unsteady perturbation atop a precalculated steady state. The boundary conditions in ANSYS FLUENT however use total pressure and temperature as if the flow were steady state as mentioned in section 3.1.1. The same equations were used to calculate these total quantities but using static pressure and Mach number values which include the unsteady contribution. The flow swirl angle was also required for the boundary condition at the upstream boundary. They were input with both the steady and unsteady (outgoing wave) contributions.

3.2.2 Three Dimensional Implementation

In three dimensions the implementation is relatively similar with the pressure at the inlet once again being found using equation (3.10) for the uniform mean flow cases. In this case radial as well as swirl flow angles are required for the unsteady calculation. The correct frequencies for the given cut-on ratios were found using the same method as demonstrated in section 3.2.1 where now μ is the scaling factor for the Bessel-Fourier mode.

The non-uniform mean flow cases once again had a Gaussian total pressure deficit profile in the circumferential direction which approximates to a Gaussian Mach number deficit. The peak and trough pressures were chosen to ensure that the mean Mach number remained at 0.5 and the maximum Mach number deficit was 10%, 15%, 20% of the freestream (peak) Mach number. The temperature was also calculated using the same method presented above.

As before similar calculations were used to set the boundary conditions in the EBNRBC by including the unsteady outgoing wave contributions to the static pressure at exit and the static pressure and velocities at inlet. The calculation was now done on a fully three dimensional annular duct where the acoustic disturbances now had a radial distribution. Accordingly a radial as well as a swirl flow angle was set at the upstream boundary condition.

The test signal in the non-uniform test cases was a non-uniform flow acoustic mode rather than a pure circumferential harmonic. For these modes there is no formal definition for a cut-on ratio so instead an approximate cut-on ratio was defined relative to the cut-on frequency to the nearest pure Bessel-Fourier harmonic in uniform flow with the same axial Mach number (0.5).

It is worth noting that the boundary conditions described in the literature review in section 2.3 focus on setting any spurious reflection to zero, for example if an acoustic wave is downstream travelling, any upstream acoustic waves are set to zero at the boundary. The implementation performed in this thesis however is subtly different. The left and right eigenvectors are found enabling one to know the direction of travel, shape and modal content of the harmonic, as well as any scattering due to non-uniform flows and geometries. This allows for the boundary to set the correct pressure and velocity flow angles (if upstream boundary) at the boundary, including contribution from outgoing waves, as opposed to setting any spurious reflections to zero. In the absence of discretisation errors either in the EAGCC method or the CFD calculation the net effect is identical.

3.2.3 Input Acoustic Signal Implementation

An acoustic signal was input for the two and three dimensional demonstrations. Using the right eigenvector directly and multiplying it by the desired amplitude divided by $\gamma \bar{p}$ an input signal was created. The division is required as FLUENTs variables are not normalised. The eigenvectors are in spectral form so an inverse Fourier transform was required to calculate the full circumferential profile. The unsteady contribution from the input signal was included in the total pressure and total temperature calculations as described in the previous sections.

One of the difficulties faced in defining boundary conditions for CFD codes running in parallel is that the boundary faces can be split across multiple partitions. The FLUENT UDF structure facilitates calculations in this situation and the code has been programmed to work with parallel processing and so can work across multiple cores and nodes. It is noted however that in larger calculations communication between partitions might cause high computational expense and further work may be required to make the procedure computationally efficient.

3.2.4 Implementation of Reference Non-Reflecting Boundary Treatments

In two dimensions the new NRBC was compared with two reference boundary treatments:

- Standard constant pressure boundary condition

- FLUENT reference NRBC

The meshes used for these alternative methods were the same as that used in the testing of the EBNRBC.

In three dimensions the reference boundary treatments were:

- Stretched mesh
- Giles EBNRBC
- FLUENT reference NRBC

The mesh used for the stretched mesh was equivalent to the mesh used in the EBNRBC case with a stretching mesh directly attached to the outlet or inlet. The stretching took place axially at a ratio of 1.1, that is, each proceeding axial point was 1.1 times further away than the last. Axial stretching is easy to implement and is all that is needed to turn cut-on waves into decaying cut-off waves. This stretching ratio was chosen as previous work [4] has shown it to work well over a wide range of acoustic modes.

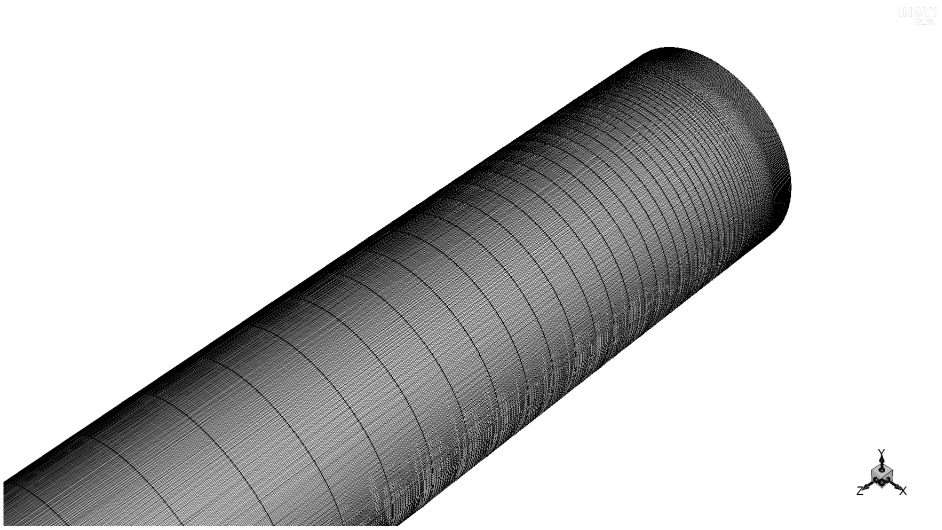


Figure 3.2: The three dimensional mesh used for the non-uniform flow test calculations that applied the stretched mesh. Here the stretching is attached to the outlet. The uniform case used a one tenth circumferential sector of this mesh. The mesh used for signals input at the outlet had the same stretching placed at the inlet. Mesh created in ANSYS Meshing.

The mesh used for the other comparisons was identical to the mesh used in the EBNRBC cases.

In the case of non-uniform flow the mode shape is deformed due to the flow. The Giles method was tested by finding the eigenvectors for uniform flow and the mode chosen to be let out of the domain was the mode closest to the mode deformed due to the flow.

3.2.5 Results Analysis

The results were analysed using the following wave splitting procedure to calculate the complex amplitude of each of the upstream and downstream travelling modes.

Noting that in each of the test cases, the mean flow was axially uniform, the wave splitting can be performed at any axial position using the left eigenvectors calculated at the boundary when setting up the non-reflecting boundary conditions.

This was done in MATLAB and involved reading in ten data files exported from the commercial CFD code containing information at each axial plane. The output variables were all three coordinates in cylindrical polar and the pressure and velocities. The ten data files made up one period of the wave equispaced in time, for example with 400 timesteps a data file was output from the calculation every 40 timesteps with the final data file (when the solution is converged) being a repeat of the first. The data was Fourier transformed in time and only the complex amplitude of the first harmonic was retained.

Before applying the left eigenvectors it was necessary to convert the flow information into tensor form. This was done by multiplying the velocity vector by the Jacobian matrix (section 2.2.7) and taking a Fourier transform in the circumferential direction. The velocity components and the pressure at each radial position were then collected into a single long flow vector $\hat{\boldsymbol{f}}$ (section 2.2.7) and premultiplied by the left eigenvector matrix.

This process provided the complex amplitude in each of the upstream and downstream travelling modes at each axial position. By plotting the absolute values of the complex amplitudes against axial position it is possible to trace the amplitude of the input mode as it traverses down the duct and compare it to the amplitude of any spurious reflected waves.

All acoustic eigenvectors are normalised such that the maximum pressure is unity, allowing direct comparison of modal amplitudes. In each case the modal coefficients are further normalised such that the outgoing mode is of uniform amplitude. In this way minor propagation errors related to discretisation in the CFD calculation do not effect the calculated reflection coefficient.

Chapter 4

Validation in 2 Dimensions

This section covers the validation of the method for use in two dimensional calculations. The results will be analysed using wave splitting. This way it is possible to see each harmonic as they traverse down the duct and the amount by which they are reflected.

The boundary conditions developed in this thesis are very general, allowing for non-uniform mean flows at a wide range of frequencies, both cut-on and cut-off, and therefore can be difficult to find the whole range of applicable conditions.

The test matrix described below uses low to moderate wave numbers and cut-on ratios as these represent an important range of frequencies for tonal analysis in the fan stage of the aeroengine, however further work is required to establish the capability at higher cut on ratios.

A fine CFD mesh was used in the test cases to ensure that discretisation errors did not contaminate the solutions to the test cases. As a result further work is required to look at cases where the outgoing mode is closer to the limit of what can be captured in the CFD mesh.

The two dimensional test cases used a thin annulus duct of length 0.1m inner radius 0.4315m and outer radius of 0.4325m. The radial distribution can thus be neglected. The mesh had 720 points equispaced circumferentially and 30 points equispaced axially. The same mesh was used in the EAGCC and the CFD calculation. A short duct was chosen to allow the calculation to converge quickly allowing a larger test matrix to be calculated. The two dimensional EBNRBC was tested at a freestream Mach number of 0.5, and 6 different frequencies, while the flow was taken to be inviscid and isentropic. A time unsteady RANS CFD calculation was performed using the inviscid model with 200 timesteps per period for each frequency. There was also 144 points per wavelength in the circumferential direction and around 100 points per wavelength in the axial direction depending on frequency. This is sufficient to capture circumferential modes ranging from $m = -10$ to 10 to be predicted to high accuracy at cut off ratios of $\omega/\omega_c = 0.9$, 1.05, 1.1, 1.2, 1.5 and 2.0 for the uniform case and 1.1 and 2.0 for the non-uniform case, where ω is the angular frequency and ω_c is the angular cut off frequency. The mode chosen in the input signal was $m = 5$ for the uniform flow case and the non-uniform mean flow mode with a pressure profile closest to the pure $m = 5$ mode for the non-uniform case. This captures the effectiveness of the boundary conditions for cut off acoustic signals and signals close to the cut off point.

In the case of downstream propagation a fixed total pressure profile was applied at inlet to achieve the desired mean flow profile. The uniform case has an axial flow of Mach number 0.5 while the non-uniform case has present a Gaussian flow deficit around the circumference that varies the axial Mach number locally from 0.5 to 0.2. Three different widths of Gaussian (variance $\sigma = 0.5$, 0.25 and 0.125 radians) were used as shown in figure 4.1.

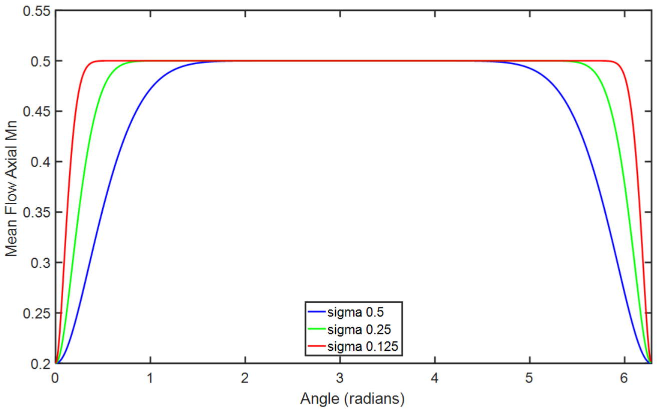


Figure 4.1: The Gaussian mean flow profiles used as test cases with variances $\sigma = 0.5, 0.25, 0.125$

The two dimensional boundary conditions were tested for upstream and downstream travelling acoustic waves.

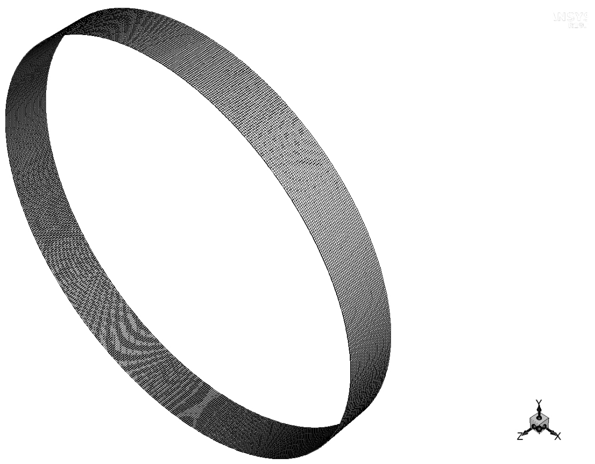


Figure 4.2: The two dimensional mesh used for both the uniform and non-uniform calculations.

These results are then compared to the standard constant pressure boundary conditions and ANSYS FLUENTs in built NRBC's dubbed here, the reference non-reflecting boundary conditions which are based on characteristic wave relations derived from the Euler equations as described in section 2.5, and can only be applied on the outlet flow boundary. An acoustic signal was introduced by applying an unsteady rotating pressure disturbance in addition to the fixed profile. In each mean flow case the temperature was corrected for ensuring a correct temperature profile for each Gaussian.

4.1 Results: Uniform Flow

4.1.1 Cut-Off Example ($\omega/\omega_c = 0.9$)

Recalling that ω_c is the cut-on frequency, with $\omega/\omega_c = 0.9$ this harmonic is a cut off mode. Figure 4.3 shows the wave amplitude normalised as described in section 3.2.5. It can be seen that the NRBC here reduces reflections by 25dB as shown by figure 4.3 predictably performing considerably better than the constant pressure boundary condition which here has negligible effect on reducing reflection and performing better than the reference inbuilt NRBC, which also does very little to reduce reflections at this cut-on ratio.

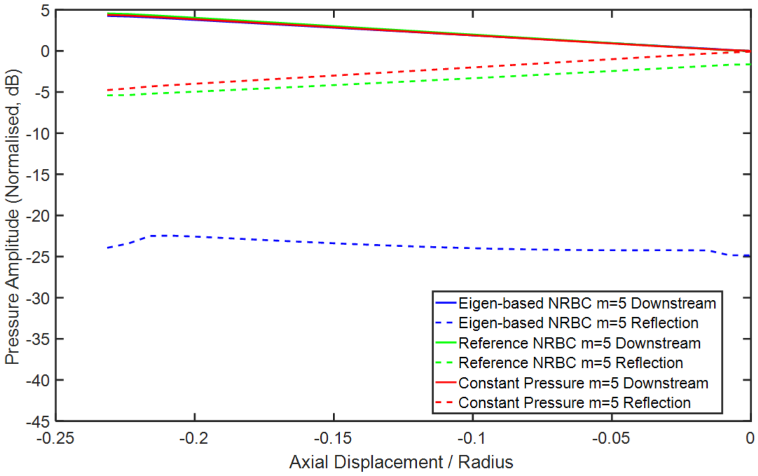


Figure 4.3: Results showing the effectiveness of the different boundary conditions at $\omega/\omega_c = 0.9$. The downstream wave is the input wave the amplitude of which decreases with axial distance as it is a cut-off evanescent wave.

4.1.2 Cut-On Example ($\omega/\omega_c = 1.5$)

In this case the reference NRBC performs much better reducing reflection by 23dB. The constant pressure boundary condition fails here whilst the eigen-based NRBC reduce reflections by 35-40dB.

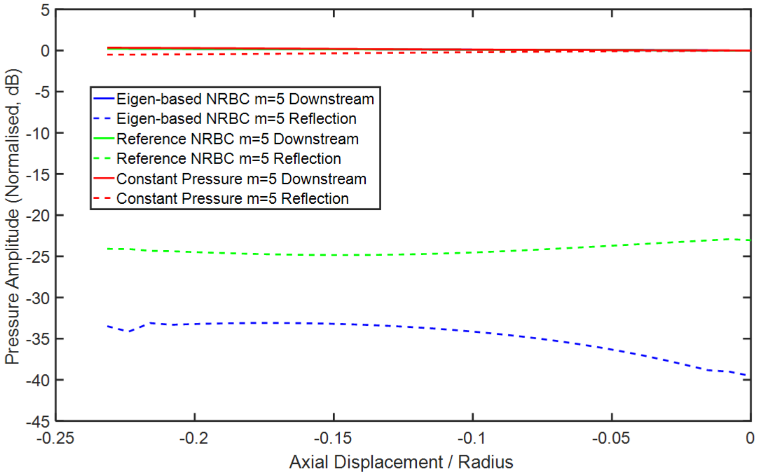


Figure 4.4: Results showing the effectiveness of the different boundary conditions at $\omega/\omega_c = 1.5$

As expected the input wave being cut-on propagates at nearly constant amplitude.

There is a slight decay due to residual discretisation errors in the CFD calculation. The reflected wave for the constant pressure boundary condition (red dashed line) shows similar behaviour. The reflection for the other two boundary conditions however show some variation in amplitude with axial distance. These signals are very small compared to the input signal. The small variation represent a noise floor at around -30dB in the wave splitting procedure in the presence of a large signal.

4.1.3 Reflection Coefficient Graph

Looking at the performance at each cut-on ratio ω/ω_c in figure 4.5, it is seen that the performance for both of the non-reflecting boundary conditions improves the further the frequency is from the cut off frequency. With that said the eigen-based NRBC's perform considerably better and are accurate enough to use at lower ratios than the reference boundary condition. Comparing the above results with $\omega/\omega_c = 1.05$ it can be seen that for the reference NRBC the reflection is reduced by 19dB whilst the eigen-based NRBC's reduce reflection by 33dB. Overall the NRBC's developed here improve over the reference NRBC by roughly 15dB regardless of how close the cut-on ratio is to 1 with the exception of the cut off mode itself where there is an improvement of 23dB. The reference boundary conditions do not perform well close to the cut off boundary whilst the eigen-based boundary conditions perform well for each cut-on ratio. As expected the constant pressure boundary condition is highly reflective and fails to produce adequate results in any of the test cases. These results show promise in the method considering the difficulty in producing boundary conditions for frequencies close to cut off.

Figure 4.5 shows the reflection coefficient based on the instant and reflected amplitudes at $m = 5$. In this uniform flow case there is no mechanism to scatter energy from $m = 5$ to other modes and the content in other circumferential harmonics is extremely small as shown in figure 4.6.

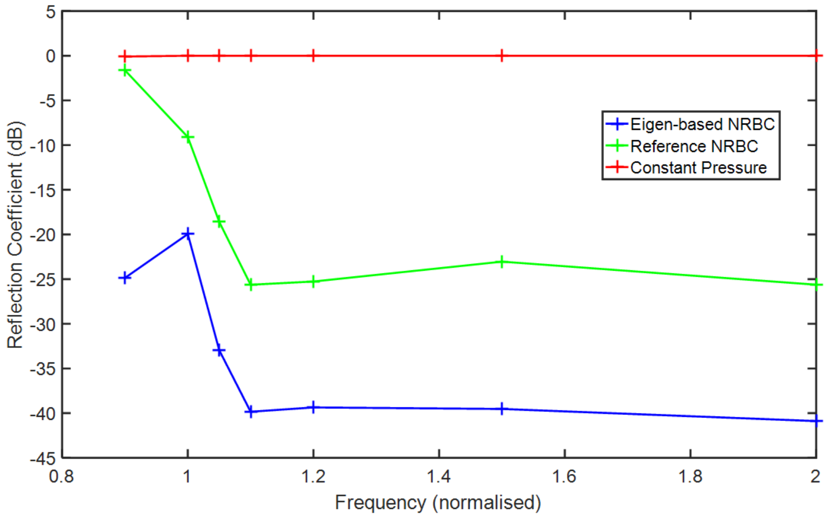


Figure 4.5: A graph showing the reflection coefficients of the different cut off ratios 0.9, 1.05, 1.1, 1.2, 1.5, 2.0

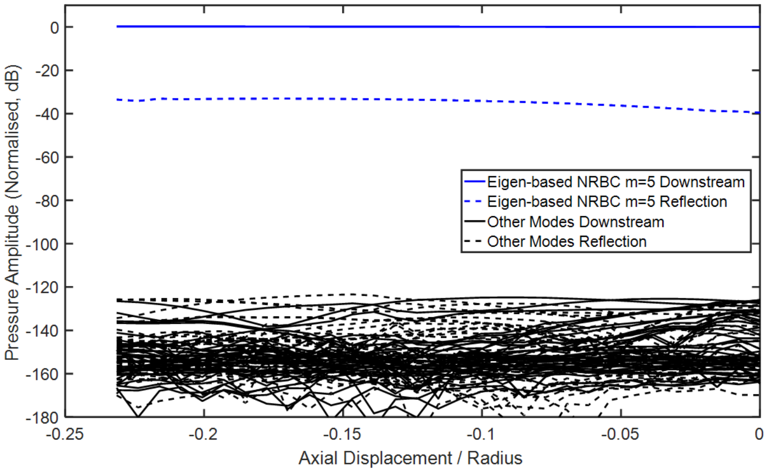


Figure 4.6: Amplitude of each circumferential mode against axial position.

Overall the new eigen-based NRBC performs best out of the boundary conditions tested. There is however a caveat. Due to the filtering process as the signal comes in, the eigen-based boundary conditions takes a considerable number of timesteps to get to the desired level of accuracy. The filter has a response time and this causes the EBNRBC to take roughly 4 times as long as the reference NRBC (This is discussed in greater detail for the three dimensional test cases in section 5.5). Hence depending on the level of accuracy required and the type of simulation the reference NRBC might be a better choice. However if a boundary condition is required upstream the EBNRBC is the better choice as the reference NRBC can only be applied downstream.

4.2 Results: Non-Uniform Flow

In the non-uniform mean flow case the test signal represents a non-uniform mean flow mode (as show previously in figures 2.1 and 2.2) not a pure circumferential harmonic. The noise present in the uniform flow cases is still present at higher amplitude however not enough to significantly effect the results. A potential reason as to why these modes increase in amplitude could be due to the EAGCC method and the CFD calculation not agreeing exactly on how much the mode spreads to neighbouring modes. This can come about due to the different discretisation methods between the eigenanlaysis and the CFD code: the eigenanalysis uses a fourth order finite difference method radially and a spectral method circumferentially, whereas the CFD code uses a less accurate third order MUSCL Finite Volume scheme. The two methods agree well for modes at low to middle wave numbers which are well-discretised in both schemes, as demonstrated by the wavesplit results for the input and reflected modes, which show the correct linear behaviour.

The method was developed to allow any number of modes to be treated, but in these test cases only a single mode was selected for treatment (see chapter 6 for a test case with multiple modes present). This results in modes not being dealt with and thus they reflect at both the inlet and outlet. Figures 4.7 and 4.8 shows forward and reflected mode amplitudes with different Gaussian flow deficit widths. Comparing the two figures it can be seen that the larger the value of the variance σ the more prominent these spurious modes become, but the amplitude remains small.

The results for the constant pressure boundary conditions will not be displayed here as they are highly reflective just as in the uniform mean flow case.

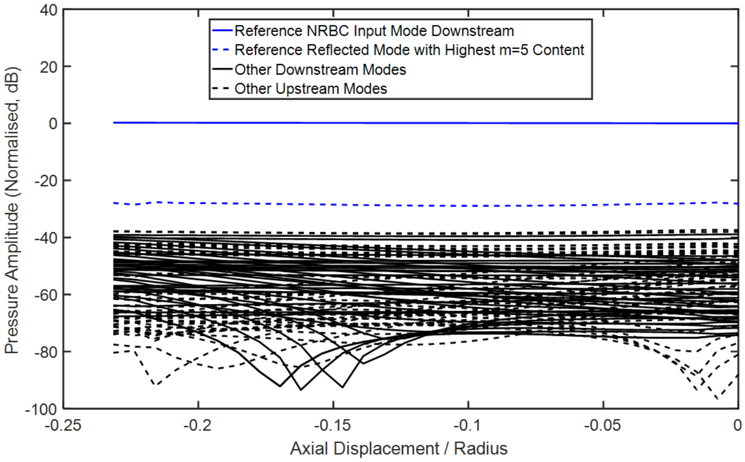


Figure 4.7: The effect of the Gaussian on the remaining mode shapes. Here $\sigma = 0.125$. It can be seen here that the highest reflected mode is the one whose highest harmonic content is 5 meaning in this case the effects of the spurious modes is negligible

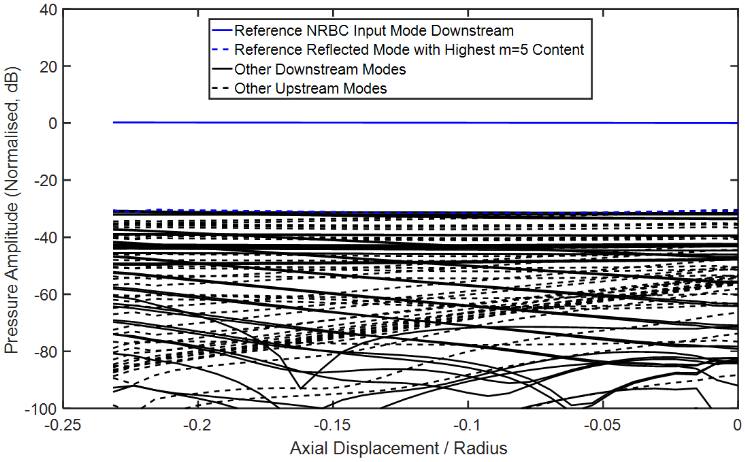


Figure 4.8: The effect of the Gaussian on the remaining mode shapes. Here $\sigma = 0.5$. Here the highest reflected mode is not necessarily the mode with highest harmonic content $m = 5$. The overall results are not spoiled.

4.2.1 Reflection Coefficient Graph at $\omega/\omega_c = 2.0$

Figure 4.9 shows the reflection coefficient of the reference NRBC and the EBNRBC. The graph shows 2 reflection coefficients. One for the strongest reflected mode and one for reflected mode with highest $m = 5$ content. Here it is shown that the eigen-based NRBC performs roughly 15dB better than the reference NRBC. When the variance is $\sigma = 0.125$ the mode with highest $m = 5$ content is also the strongest reflected mode. When $\sigma = 0.25$ this is no longer the case however the two modes are very close in amplitude. Finally when $\sigma = 0.5$ the reflected mode with highest $m = 5$ content is 10dB lower than the highest reflected mode. In any case none of these reflected modes play any considerable role in spoiling the results.

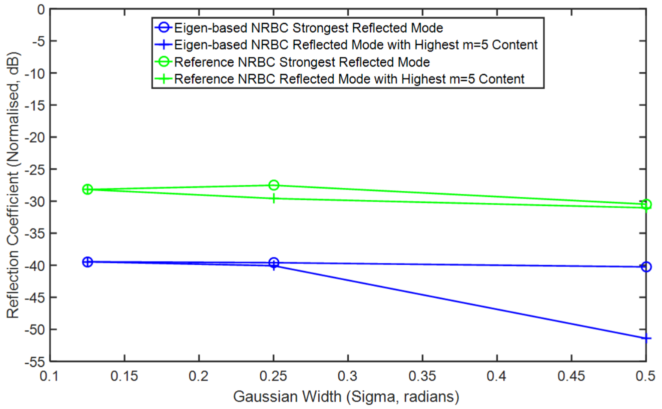


Figure 4.9: Comparison of the reference NRBC to the eigen-based NRBC. Included here are the strongest reflected modes and the reflected mode with the highest mode 5 content. This is at a cut-on ratio of $\omega/\omega_c = 2.0$

4.2.2 Reflection Coefficient Graph at $\omega/\omega_c = 1.1$

The results here are close to those shown above. It can be seen that the eigen-based NRBC performs roughly 15dB better than the reference NRBC and when $\sigma = 0.125$ the mode with highest $m = 5$ content is the strongest reflected mode and when $\sigma = 0.25$ this is no longer the case. The two modes, however, are very close in amplitude. Finally when $\sigma = 0.5$ the mode with highest $m = 5$ content is much lower than the highest reflected mode by 10dB. None of these reflected modes play any considerable role in spoiling the results.

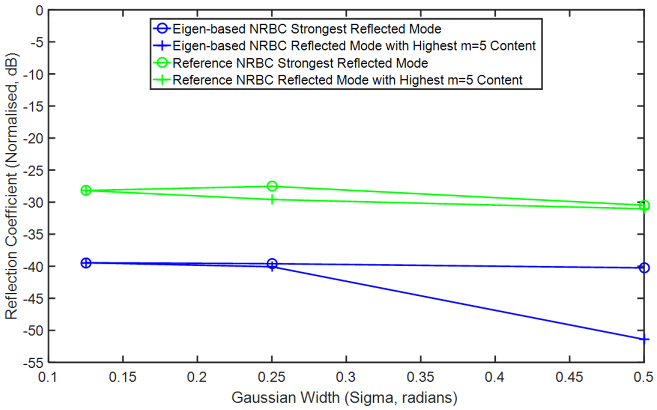


Figure 4.10: A graph to compare the reference NRBC to the eigen-based NRBC. Included here are the strongest reflected modes and the reflected mode with the highest mode 5 content. This is at a cut-on ratio of $\omega/\omega_c = 1.1$

The newly developed EBNRBC's perform well in the non-uniform flow setting reducing reflections downstream by 40dB. The reference NRBC does surprisingly well reducing the reflection by roughly 26 - 30dB. However, as mentioned, the eigen-based NRBC's outperforms the reference NRBC by roughly 15dB for each run. The constant pressure boundary is entirely reflective.

The timestep problem here returns trading in a quick calculation time for increased accuracy. It is possible however to increase the band gap in the bandpass filter allowing for quicker response times of the filter. This will be discussed further in section 5.4.2.

The issue present with the reference NRBC's is that they can only be applied to the outlet while the eigen-based NRBC's developed here have been demonstrated to work effectively at inlet and outlet.

Chapter 5

Validation in 3 Dimensions

The three dimensional cases were calculated in a uniform cylindrical annular duct. The mesh had 720 points equispaced circumferentially, 30 points equispaced axially and 60 points equispaced radially. The length of the duct was 0.1m while the inner and outer diameters were 0.2m and 0.432m respectively. Again a short duct was chosen to allow the calculation to converge quicker giving results faster. The EBNRBC's were tested at 3 different Mach numbers, $M = 0.3$, 0.5, 0.7 and three different frequencies normalised by the cut-on frequency, while the flow was taken to be inviscid and isentropic. It should be noted that the uniform case was a sector of the full cylindrical mesh. The circumferential input mode chosen was constant at $m = 10$ for both the uniform and non uniform cases and so the mesh was chosen to be a one tenth sector as the acoustic wave in the uniform case is periodic.

The non-uniform mean flow runs took place on the full 360 degree annulus mesh as circumferential periodicity can no longer be assumed when acoustics travel through non-uniform mean flow. The flow was made non-uniform by having a circumferentially varying Mach number with a mean Mach number of 0.5 as described in section 3.2.2.

The three dimensional boundary conditions were tested for upstream and downstream travelling acoustic waves. The mean flow was kept constant in each case, in order to assess the effect of circumferential variation in flow separately to overall mean flow effects.

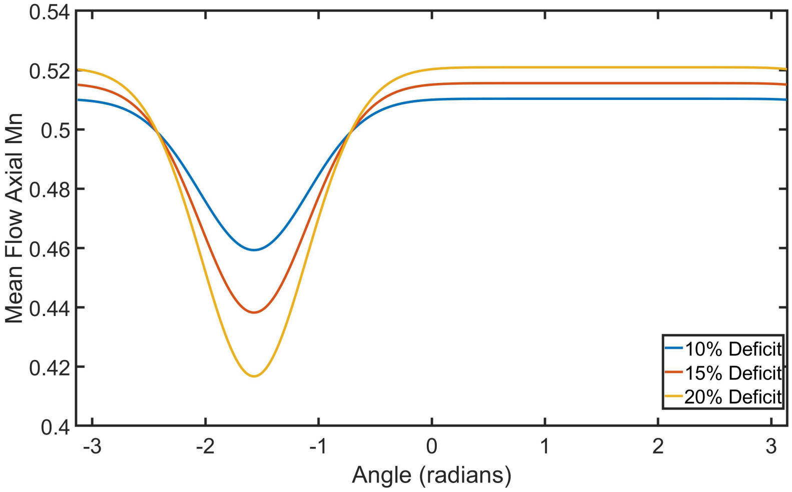


Figure 5.1: The Gaussian mean flow profiles used as test cases at $\sigma = 0.5$.

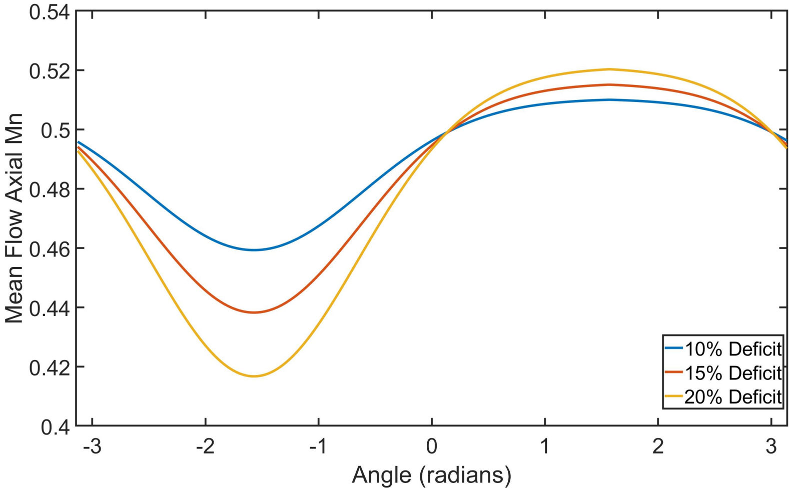


Figure 5.2: The Gaussian mean flow profiles used as test cases at $\sigma = 1$.

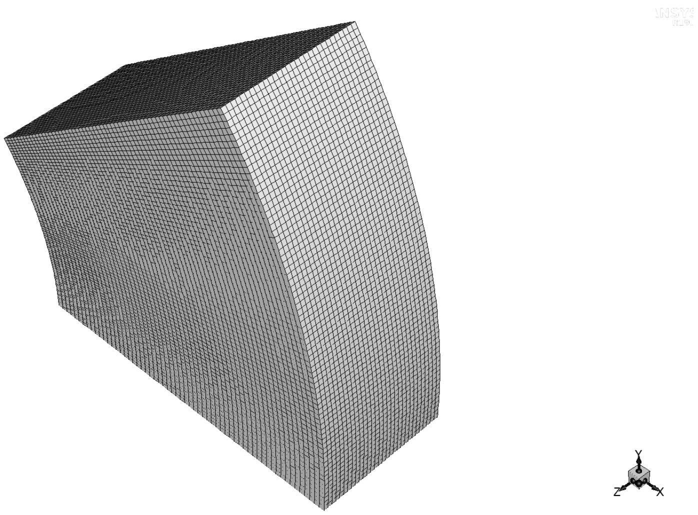


Figure 5.3: The three dimensional mesh used for the uniform flow calculations.

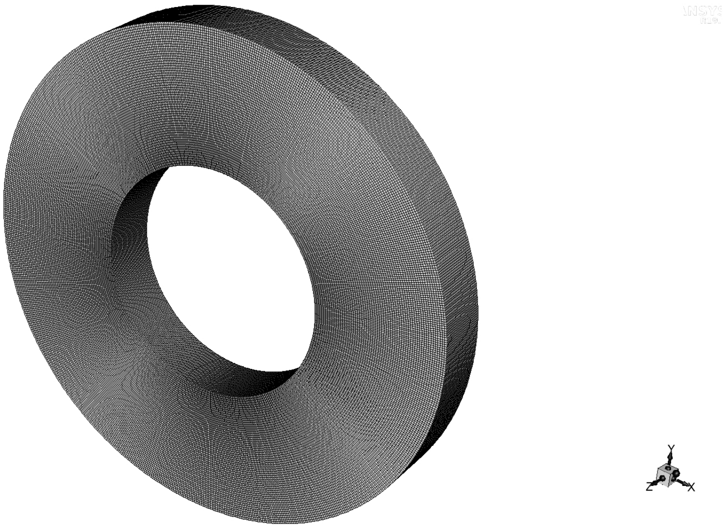
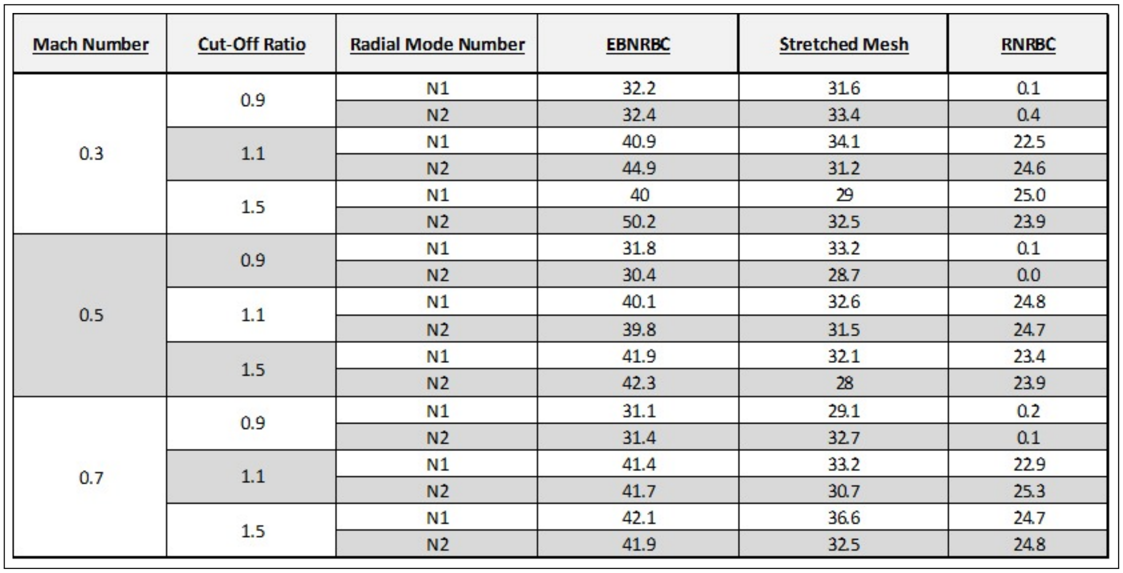


Figure 5.4: The three dimensional mesh used for the non-uniform flow calculations.

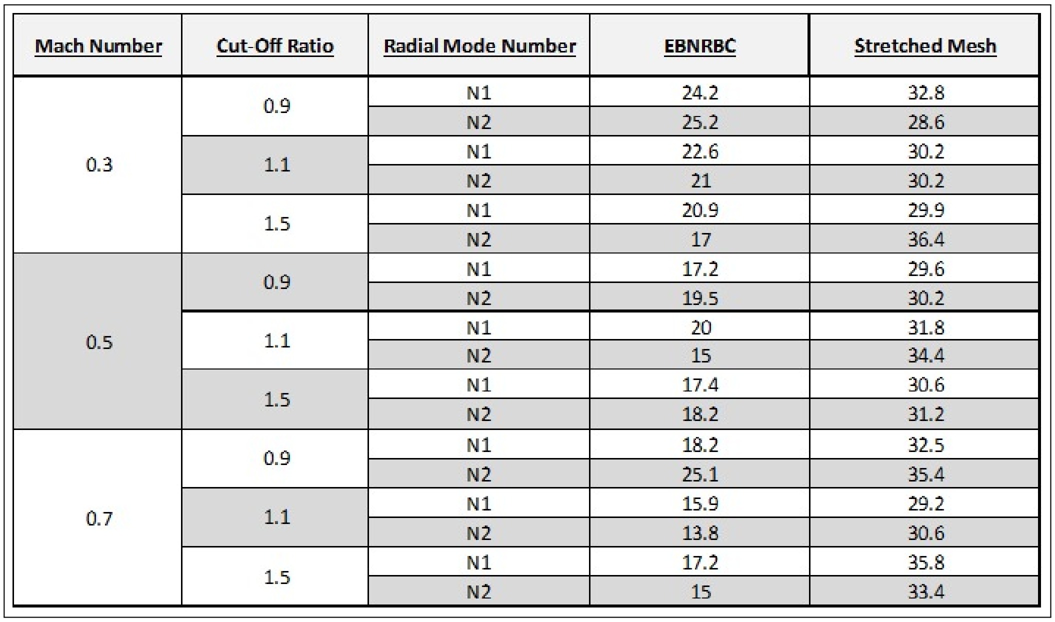
5.1 Results: Uniform Flow

Tables 5.1 and 5.2 show the results for all cases for uniform flow showing the reflection coefficients for the downstream and upstream travelling acoustic waves respectively.



Mach Number	Cut-Off Ratio	Radial Mode Number	EBNRBC	Stretched Mesh	RNRBC
0.3	0.9	N1	32.2	31.6	0.1
		N2	32.4	33.4	0.4
	1.1	N1	40.9	34.1	22.5
		N2	44.9	31.2	24.6
	1.5	N1	40	29	25.0
		N2	50.2	32.5	23.9
0.5	0.9	N1	31.8	33.2	0.1
		N2	30.4	28.7	0.0
	1.1	N1	40.1	32.6	24.8
		N2	39.8	31.5	24.7
	1.5	N1	41.9	32.1	23.4
		N2	42.3	28	23.9
0.7	0.9	N1	31.1	29.1	0.2
		N2	31.4	32.7	0.1
	1.1	N1	41.4	33.2	22.9
		N2	41.7	30.7	25.3
	1.5	N1	42.1	36.6	24.7
		N2	41.9	32.5	24.8

Table 5.1: Reflection coefficients calculated for uniform mean flow for the downstream cases. EBNRBC is the current eigen-based NRBC, RNRBC is the Reference NRBC.



Mach Number	Cut-Off Ratio	Radial Mode Number	EBNRBC	Stretched Mesh
0.3	0.9	N1	24.2	32.8
		N2	25.2	28.6
	1.1	N1	22.6	30.2
		N2	21	30.2
	1.5	N1	20.9	29.9
		N2	17	36.4
0.5	0.9	N1	17.2	29.6
		N2	19.5	30.2
	1.1	N1	20	31.8
		N2	15	34.4
	1.5	N1	17.4	30.6
		N2	18.2	31.2
0.7	0.9	N1	18.2	32.5
		N2	25.1	35.4
	1.1	N1	15.9	29.2
		N2	13.8	30.6
	1.5	N1	17.2	35.8
		N2	15	33.4

Table 5.2: Reflection coefficients calculated for uniform mean flow for the upstream cases.

5.1.1 Uniform Flow Boundary condition Performance: EBNRBC

The eigen-based NRBC performs well at the downstream boundary giving a reflection coefficient of between 40 and 50dB below the input signal for all cases at all Mach numbers, although it is noted this is below the -25dB noise floor. The upstream EBNRBC also performs well giving a reflection coefficient of between 14 and 25dB below the input signal for each case at all Mach numbers. Note that there is a small but sharp change in amplitude between the inlet plane and the rest of the calculation. The cause for this change is not yet known. It is more significant when the input signal is in the second radial mode (not shown) however in both cases the signal itself is close to the wave splitting noise floor relative to the input signal.

The following results at a cut-on ratio of $\omega/\omega_c = 1.1$ are representative of the behaviour across the range tested.

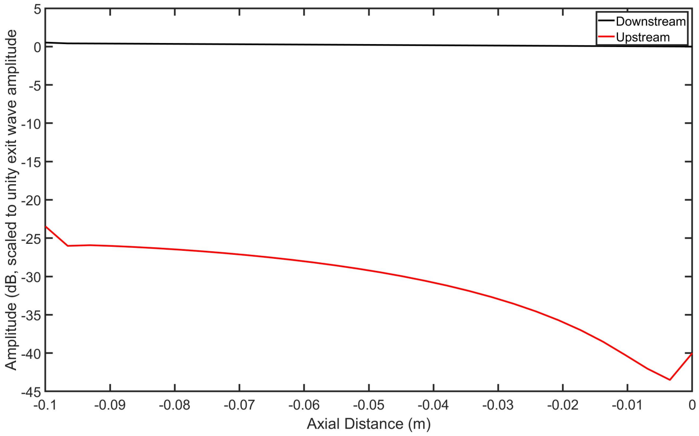


Figure 5.5: Results showing the effectiveness of the EBNRBC of a downstream wave at first radial mode for $Mn = 0.5$.

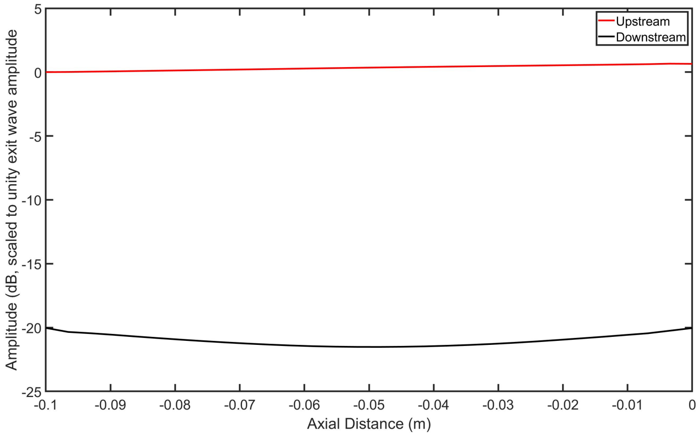


Figure 5.6: Results showing the effectiveness of the EBNRBC of an upstream wave at first radial mode for $Mn = 0.5$.

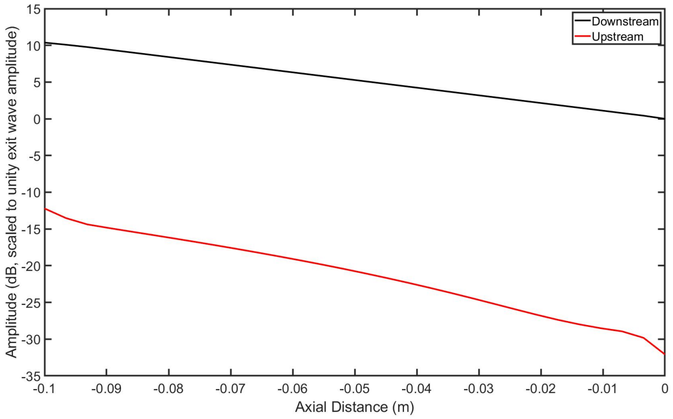


Figure 5.7: Results showing the effectiveness of the EBNRBC of a downstream cut-off wave at first radial mode for $Mn = 0.5$.

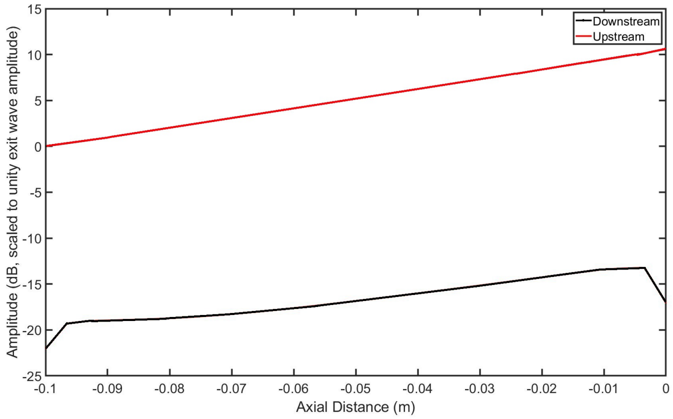


Figure 5.8: Results showing the effectiveness of the EBNRBC of an upstream cut-off wave at first radial mode for $Mn = 0.5$.

Figures 5.7 and 5.8 show an example of a cut-off calculation. Accordingly the input wave decreases in amplitude as it goes down the duct. In figure 5.7 this is the downstream wave (black line) and in figure 5.8 this is the upstream wave (red line). As can be seen the reflected modes seem to increase in amplitude as they travel along the duct, this implies that the apparent reflected waves are not genuine reflections, they are a spurious result from the wave splitting. It's notable that the amplitude of these apparent reflections are everywhere at least 20dB below the amplitude of the input signal. This is higher than the noise floor observed in the cut-on test cases but this is to be expected for signals close to the cut-on cut-off boundary: at the boundary itself the upstream and downstream waves are identical and therefore cannot be distinguished by wave splitting.

5.1.2 Uniform Boundary condition Performance: Stretched Grid

The stretched grid performs equally well downstream and upstream reducing reflections from anywhere between 28dB and 37dB for all cases at all Mach Numbers. This represents an improvement over the EBNRBC at the upstream boundary but lower effectiveness at the downstream boundary. The following results are at a cut-on ratio of $\omega/\omega_c = 1.1$ and are representative results of what was present in most cases.

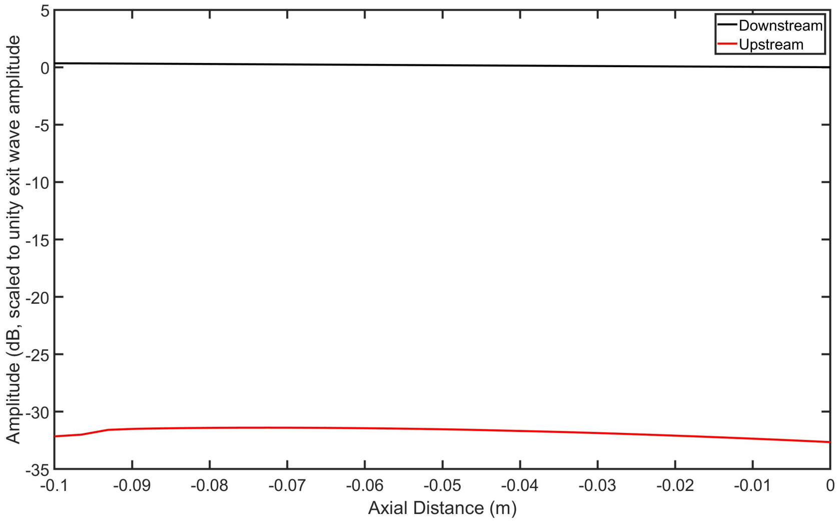


Figure 5.9: Results showing the effectiveness of the stretched mesh of a downstream wave at first radial mode for $Mn = 0.5$.

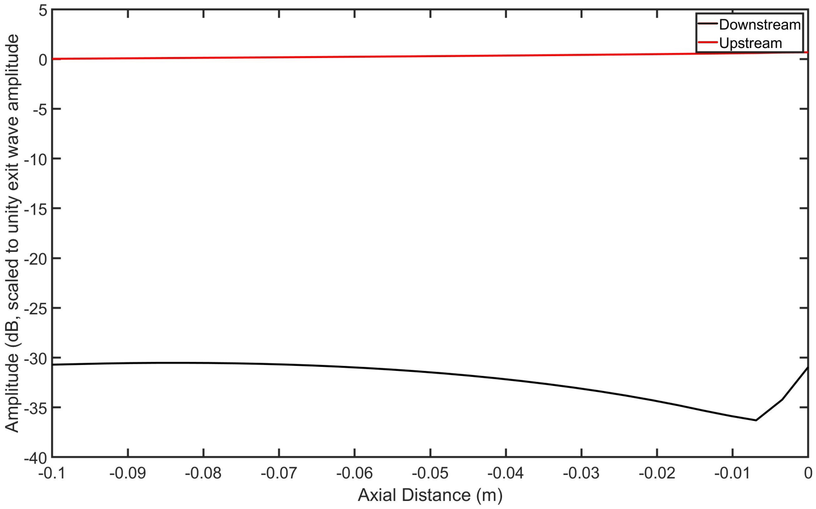


Figure 5.10: Results showing the effectiveness of the stretched mesh of an upstream wave at first radial mode for $Mn = 0.5$.

5.1.3 Uniform Boundary condition Performance: Reference NRBC

It should first be noted that the reference NRBC cannot be applied upstream. The Reference NRBC here performs less well than the EBNRBC downstream, producing a reflection coefficient of between 22 and 25dB below the input signal for all cut-on cases at all Mach Numbers. The boundary condition was found to be entirely reflective for cut-off modes as shown in figure 5.1. It should be noted that the second radial mode, while reflection coefficients are roughly similar to that of the first radial modes seem to suffer from sharp increases again at the inlet boundary. The following result displayed is at a cut-on ratio of $\omega/\omega_c = 1.1$. This again is representative of the results present in most cases.

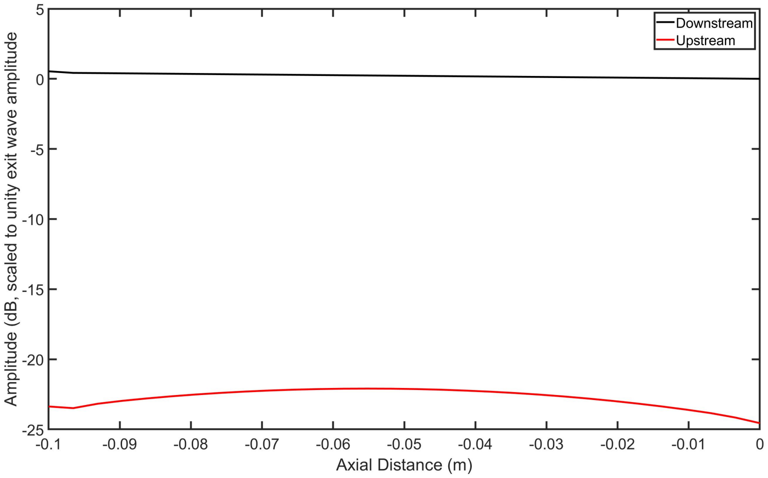
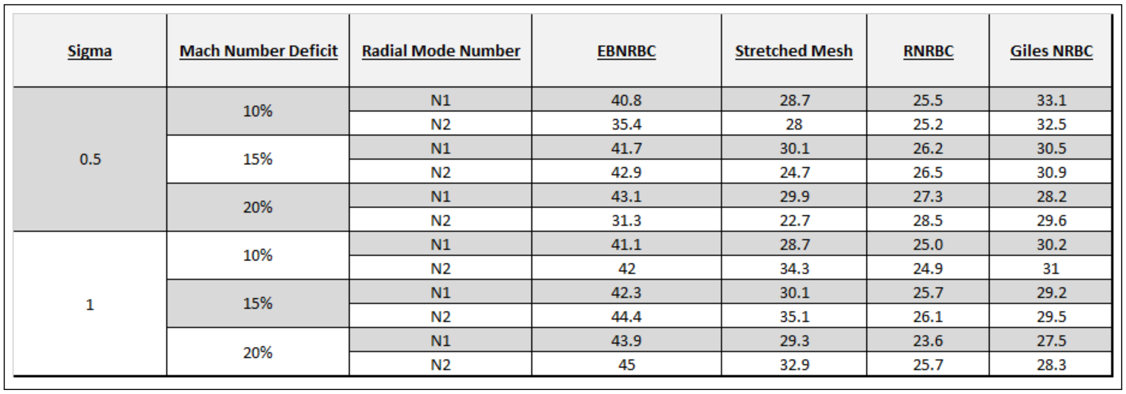


Figure 5.11: Results showing the effectiveness of the Reference NRBC of a downstream wave at first radial mode for $Mn = 0.5$.

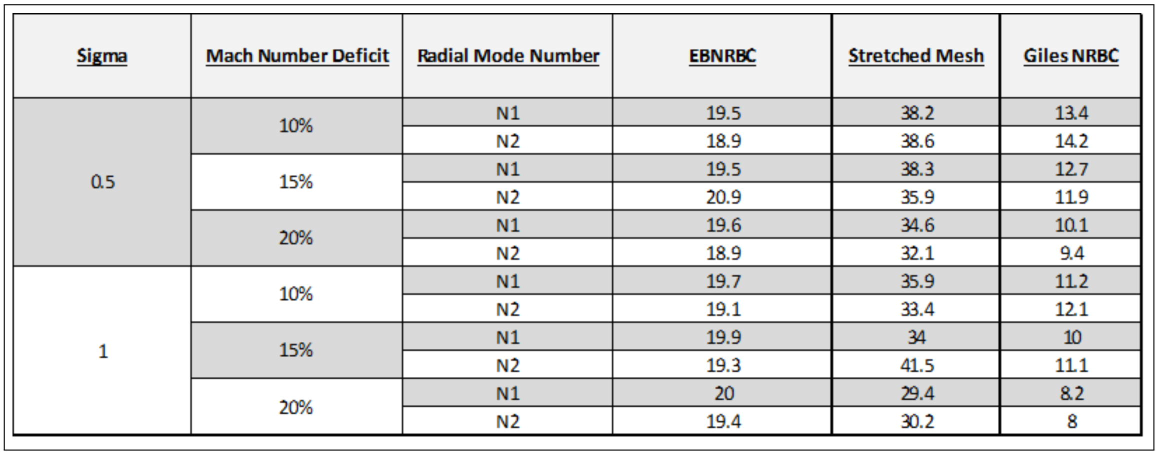
5.2 Results: Non-Uniform Mean Flow

The results displayed here are for a Gaussian width of $\sigma = 0.5$ with a 15% Mach number deficit for each boundary method as this case is representative of all test cases performed. The tables in figures 5.3 and 5.4 display the results for all non-uniform cases with downstream and upstream travelling waves respectively:



Sigma	Mach Number Deficit	Radial Mode Number	EBNRBC	Stretched Mesh	RNRBC	Giles NRBC
0.5	10%	N1	40.8	28.7	25.5	33.1
		N2	35.4	28	25.2	32.5
	15%	N1	41.7	30.1	26.2	30.5
		N2	42.9	24.7	26.5	30.9
	20%	N1	43.1	29.9	27.3	28.2
		N2	31.3	22.7	28.5	29.6
1	10%	N1	41.1	28.7	25.0	30.2
		N2	42	34.3	24.9	31
	15%	N1	42.3	30.1	25.7	29.2
		N2	44.4	35.1	26.1	29.5
	20%	N1	43.9	29.3	23.6	27.5
		N2	45	32.9	25.7	28.3

Table 5.3: Reflection coefficients calculated for non-uniform mean flow for the downstream acoustic waves.



Sigma	Mach Number Deficit	Radial Mode Number	EBNRBC	Stretched Mesh	Giles NRBC
0.5	10%	N1	19.5	38.2	13.4
		N2	18.9	38.6	14.2
	15%	N1	19.5	38.3	12.7
		N2	20.9	35.9	11.9
	20%	N1	19.6	34.6	10.1
		N2	18.9	32.1	9.4
1	10%	N1	19.7	35.9	11.2
		N2	19.1	33.4	12.1
	15%	N1	19.9	34	10
		N2	19.3	41.5	11.1
	20%	N1	20	29.4	8.2
		N2	19.4	30.2	8

Table 5.4: Reflection coefficients calculated for non-uniform mean flow for the upstream acoustic waves.

As can be seen from the tables, the downstream results always perform better than the upstream results for the EBNRBC. The stretched mesh on the other hand performs better for upstream flowing acoustic waves whilst the reference NRBC performs as well as it has been even for non-uniform flows. The RNRBC can not be applied upstream.

5.2.1 Non-Uniform Boundary Condition Performance: EBNRBC

The eigen-Based NRBC performs well downstream giving a reflection coefficient of between 35 and 45dB below the input signal and producing reflection coefficients of 20dB below the input signal for all upstream cases. As previously the downstream travelling waves are represented by the black line while the upstream waves are represented by the red line.

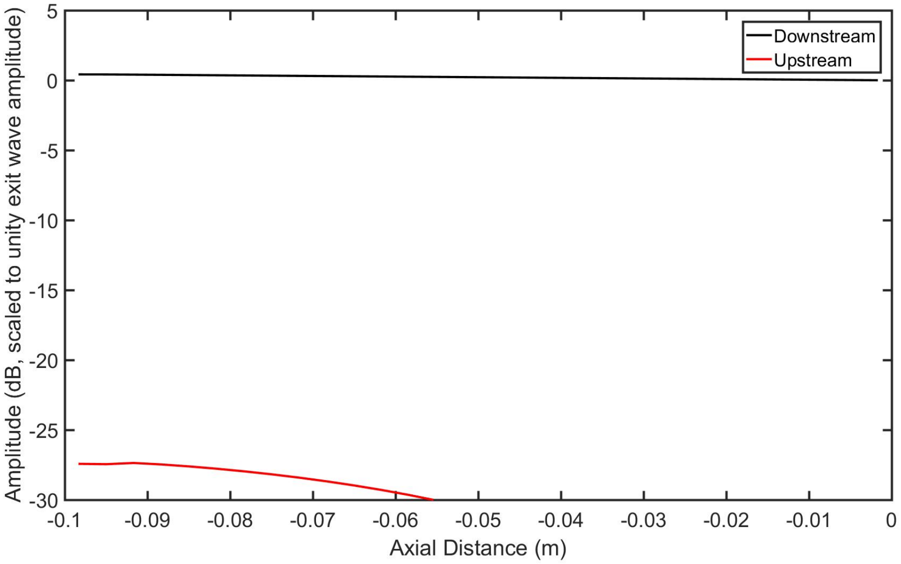


Figure 5.12: Results showing the effectiveness of the EBNRBC of a downstream wave at first radial mode for a Gaussian Mach number deficit of 15%.

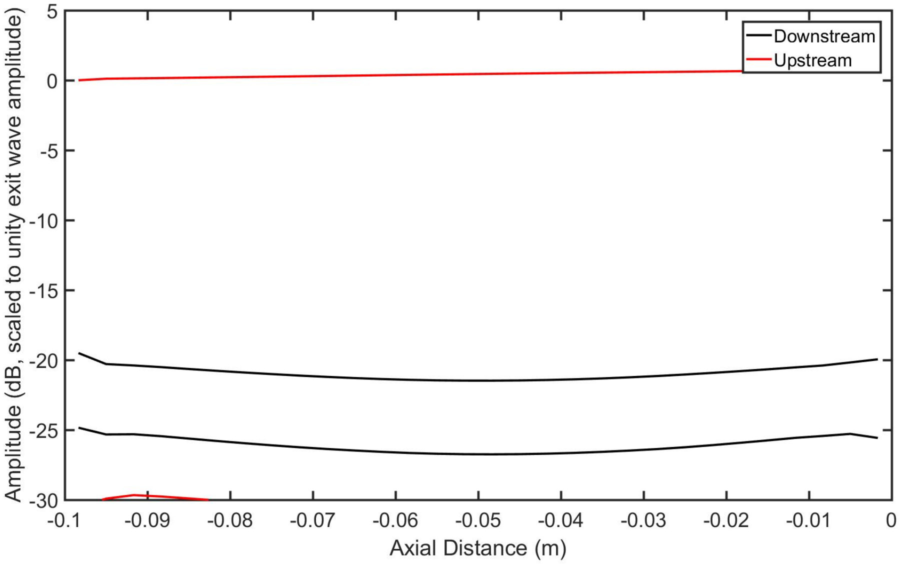


Figure 5.13: Results showing the effectiveness of the EBNRBC of an upstream wave at first radial mode for a Gaussian Mach number deficit of 15%.

5.2.2 Non-Uniform Boundary Condition Performance: Stretched Mesh

The stretched grid performs well producing reflection coefficients between 23dB and 35dB below the input signal for the downstream cases and between 30dB and 42dB for the upstream cases at all Mach Numbers. This represents an improvement over the EBNRBC at the upstream boundary but lower effectiveness at the downstream boundary.

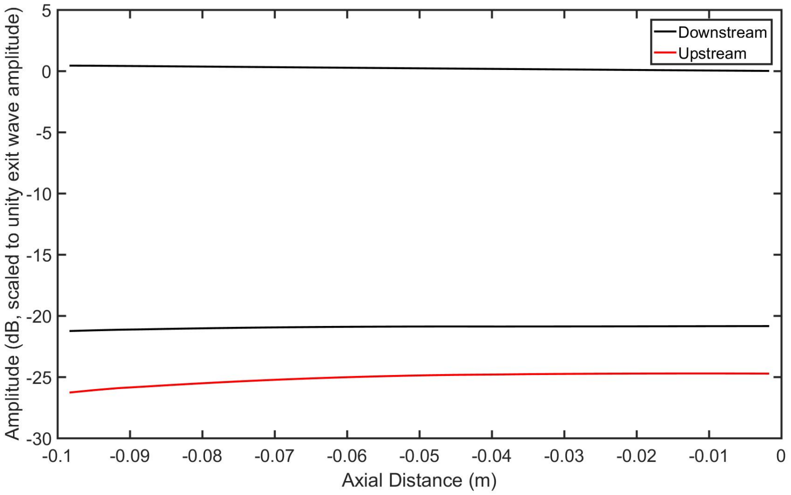


Figure 5.14: Results showing the effectiveness of the stretched mesh of a downstream wave at first radial mode for a Gaussian Mach number deficit of 15%.

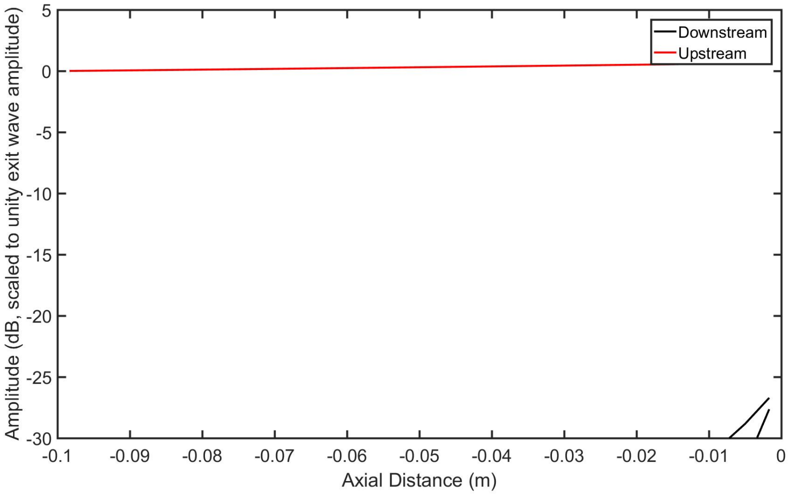


Figure 5.15: Results showing the effectiveness of the stretched mesh of an upstream wave at first radial mode for a Gaussian Mach number deficit of 15%.

5.2.3 Non-Uniform Boundary condition Performance: Reference NRBC

The Reference NRBC here performs less well downstream producing reflection coefficients between 24 and 29dB below the input signal for all cases.

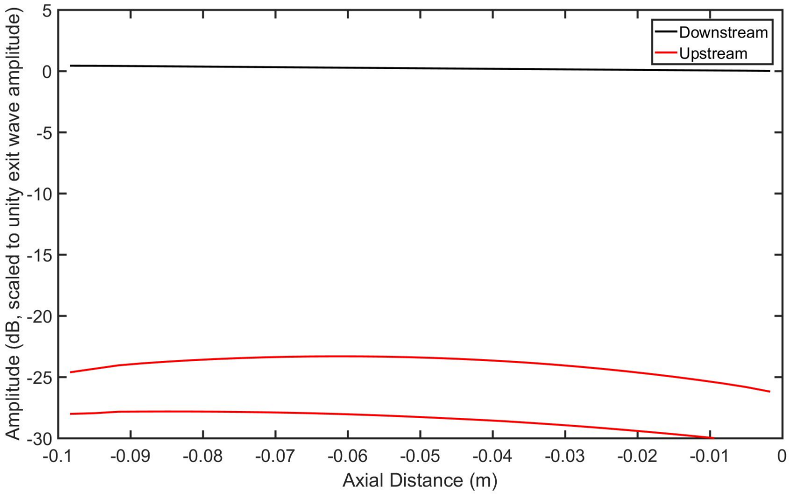


Figure 5.16: Results showing the effectiveness of the reference NRBC of a downstream wave at first radial mode for a Gaussian Mach number deficit of 15%.

5.2.4 Non-Uniform Boundary condition Performance: Giles EBNRBC

The Giles EBNRBC gives results that, at first, look inadequate. Recalling that non-uniform mean flow causes scattering of the mode into neighbouring harmonics, the boundary condition reduces reflection for the $m = 10$ harmonic however fails to be non-reflecting for the remaining modal content.

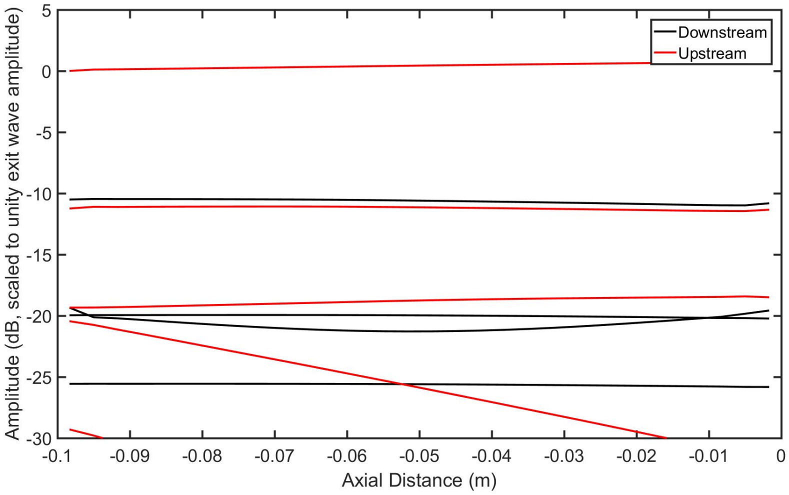


Figure 5.17: Results showing the effectiveness of the Giles EBNRBC of an upstream wave at first radial mode for a Gaussian Mach number deficit of 15% and variance $\sigma = 0.5$.

The image here shows a reduction in reflection of 10dB. Looking at the wide Gaussian of variance $\sigma = 1$ shows an even worse result.

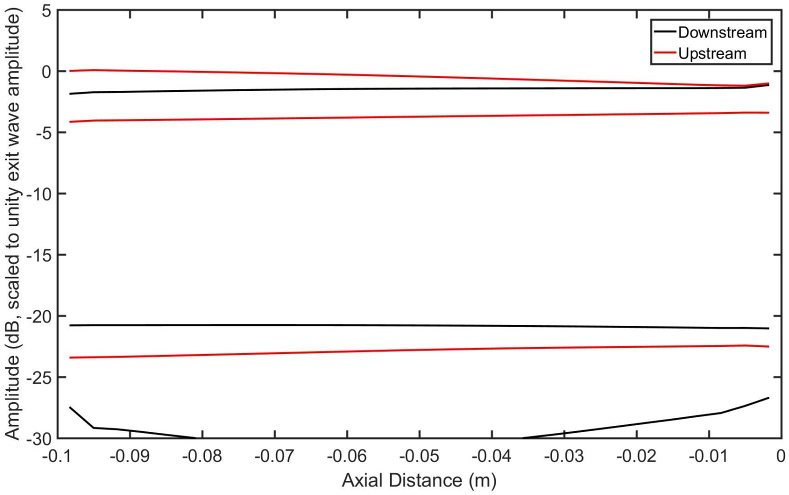


Figure 5.18: Results showing the effectiveness of the Giles EBNRBC of an upstream wave at first radial mode for a Gaussian Mach number deficit of 15% and variance $\sigma = 1$.

The reason for the high reflectivity in figures 5.17 and especially 5.18 is due to the fact that much of the harmonic content is left untreated. In figure 5.18 the harmonic being reflected is left untreated by the Giles EBNRBC meaning the reflected harmonic is not at

$m = 10$. In this case harmonic $m = 11$ has the highest amplitude. This harmonic is close to the cut-off point and when it meets the untreated downstream boundary the reflection causes a resonance causing the harmonic to have high amplitude. However upon looking at the reflected modes it is seen that the downstream and upstream waves have very similar profiles, and so interfere with each other destructively as shown in figures 5.19 and 5.20. That is the pressure amplitude is much smaller than the individual wave amplitudes.

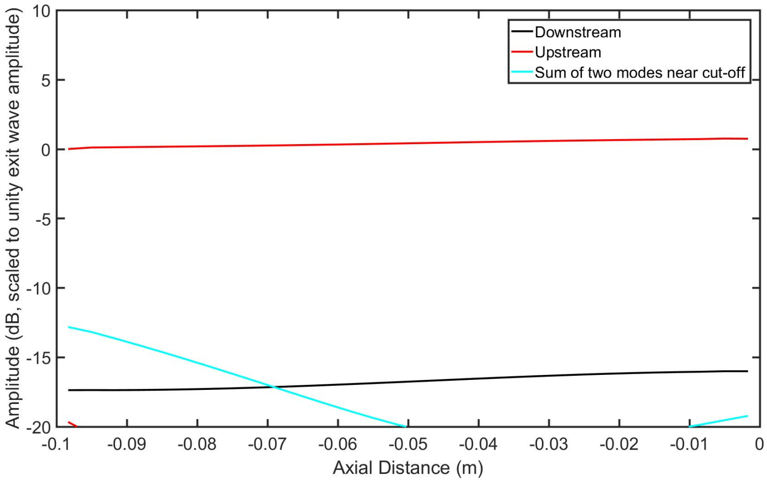


Figure 5.19: Giles EBNRBC results for a Gaussian Mach number deficit of 15% and variance $\sigma = 0.5$ this time with accounting for the interaction between reflected modes.

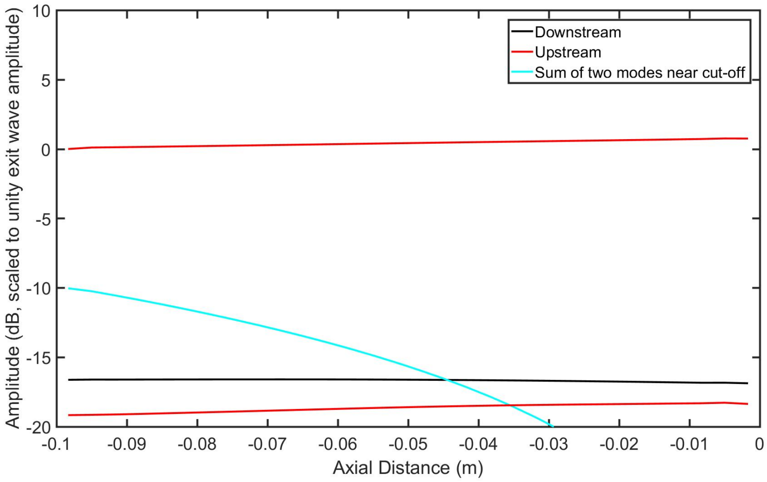


Figure 5.20: Giles EBNRBC results for a Gaussian Mach number deficit of 15% and variance $\sigma = 1$ this time with accounting for the interaction between reflected modes.

These graphs show the Giles EBNRBC are more effective than appeared initially. The amplitude of the reflection can now be seen to be -13dB amd -10dB for $\sigma = 0.5$ and 1 respectively. The downstream results are not shown as the Giles EBNRBC performs well for them. They give reflections in the given cases at the downstream boundary of about -30dB. This is because the harmonic scattering is not as severe for a mode travelling downstream as it is for a mode travelling upstream.

5.3 Results Summary

5.3.1 Uniform Mean Flow

In the case of uniform mean flow the EBNRBC yields reflection coefficients for downstream acoustic waves 40 to 50 dB below the input signal, although it should be noted that this is below the nominal noise floor of -25dB. This exceeds the performance of the stretched mesh (23 - 35dB) and the reference NRBC (20 - 25dB). For the upstream cases, the stretched mesh (30 - 42dB) performs better than the EBNRBC (15 - 20dB) while the reference NRBC is limited to being applied at the outlet only.

5.3.2 Non-Uniform Mean Flow

The results for non-uniform mean flow were qualitatively similar to the uniform mean flow cases. The EBNRBC works best producing reflection coefficients of about 40dB below the input signal for downstream acoustic waves while the stretched mesh follows with reductions of about 30dB. The reference NRBC is least effective giving reflection coefficients of between 26 and 30dB below the input signal. For the upstream cases, the stretched mesh (28 - 40dB) again outperforms the EBNRBC (20dB). The Giles EBNRBC, while not shown here performed well for downstream travelling acoustic waves. For upstream travelling acoustic waves the harmonic scattering caused by non-uniform flows is much more severe causing the boundary condition at the upstream boundary to perform inadequately. Modes can be scattered into neighbouring harmonics which may be close to the cut off boundary. This gives rise to modes with high amplitude but low acoustic energy. As the downstream boundary in these cases were untreated it allows the harmonic close to cut-off to resonate resulting in the high harmonic content seen in figure 5.18. The profile however to the up and downstream travelling waves are similar and destructively interfere with one another. This means that the true reflection isn't as high as one may initially think and so the Giles EBNRBC are still effective for non-uniform mean flow for upstream travelling waves however not as effective as the new EBNRBC developed here. Kousen [32] and Moiner et al [84] can help produce boundary conditions for flows with swirl allowing an element of non-uniformity of the flow. The reference boundary conditions work well for well cut-on modes and fail for cut-off. These boundary conditions work quickly however can only be applied downstream.

Overall, the EBNRBC outperformed both of the comparison treatments for waves in the downstream direction. In the upstream direction it was less effective than the stretched mesh treatment, however it is not subject to the limitations of zonal methods explained previously, that is, the requirement for a larger calculation domain to be able to implement it and signals cannot be input at the same boundary to which the treatment is applied.

The results for each boundary condition tested across all of the Mach numbers, cut off ratios, radial mode numbers, mean flow Gaussian depths and mean flow Gaussian widths are remarkably consistent as shown by the narrow ranges in the reflection coefficients shown in the previous sections. The only exception to this is the reference NRBC which is ineffective for cut-off modes.

5.4 Bandpass Filter Effects

5.4.1 Filter Phase Lag

The use of a filter creates a lag in the phase. This causes the EBNRBC to perform inadequately if ignored. This was tested during the full annulus trial calculations at $m = 5$ with uniform mean flow as illustrated in figure 5.21 for the case where the cut off ratio is $\omega/\omega_c = 1.5$. Without the phase correction the amplitude of the reflected wave is only 19dB below the input signal. With the phase correction the amplitude of the reflected wave is 39dB below the input signal. The phase correction is made up of two components, firstly the phase lag of the filter itself which was determined by passing an artificially generated signal through the filter and comparing the phase of the input and output signals and the 1 timestep lag due to the boundary condition being based on the flow variables of the previous timestep. In general the phase correction will be a function of the number of timesteps per wave period. Hence a constant 400 timesteps per period was used in the three dimensional case for each case in both uniform and non-uniform flows while 200 timesteps per period was used in the two dimensional calculations. The phase correction was 0.11

radians in two dimensions and 0.077 in three dimensions. In three dimsnions 0.073 radians was due to the filter while 0.004 radians was due to the timestep showing a much larger contribution by the filter.

An effect present in only the two dimensional runs was that the closer the frequency was to cut off the longer it took for the solution to converge with roughly 6000 timesteps for $\omega/\omega_c = 1.05$ and only 800 timesteps for $\omega/\omega_c = 2$ which is a significant difference. This effect is not seen in the three dimensional runs.

5.4.2 Band Pass Width

The filter, by nature, has a response time. This results in an increase in calculation time. The response time is dependant on how tight the bandpass filter is. The filter used had a normalised frequency band of 0.9 - 1.1 multiplied by the target frequency. The response time can be decreased by widening the band gap whilst still preserving the accuracy. The phase correction is also dependent on the band gap and therefore new phase corrections had to be calculated for each different band gap value. Table 7.1 shows the number of timesteps required for each bandpass gap tested to get the same level of accuracy as that present with a band gap of 0.2. Not shown in the table is the band gap of 0.3 as this began to show an increase in timesteps and a reduction in accuracy. This plays a role in how competitive the EBNRBC is in terms of compute time and power when compared to alternative methods which is discussed further in the next section.

Band Gap	Number of Timesteps
0.2	9000
0.22	7500
0.24	6000
0.26	5500
0.28	5000

Table 5.5: This table shows the number of timesteps required to get the same level of accuracy when the band pass gap is 0.2

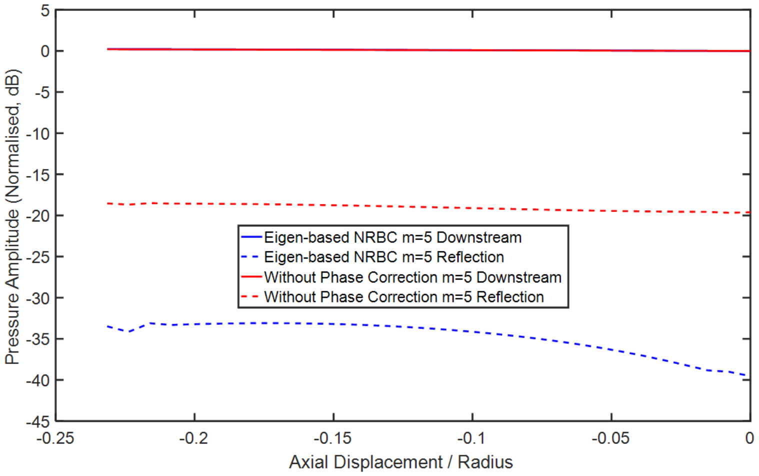


Figure 5.21: This figure shows the effectiveness of the eigen-based solver with and without the phase correction for a uniform mean flow case where $\omega/\omega_c = 1.5$.

5.5 Calculation Time

Although calculation time was not the focus of this work some useful observations were made as to the relative calculation efficiency of the boundary treatments.

The eigen-based boundary conditions take an increased number of timesteps to get to the level of accuracy shown in the results. The EBNRBC takes roughly 9,000 timesteps to achieve these levels in both the uniform mean flow and non-uniform mean flow cases, whilst the reference NRBC requires only 2000 and the stretched mesh roughly 4000. The preliminary steady state calculation, however, for the stretched mesh takes much longer to converge than both the EBNRBC and the reference NRBC which both use the same mesh. The number of iterations required to get a well converged steady state is in the range of 70,000 for the stretched mesh while for the EBNRBC and the reference NRBC it is in the range of 3500. As a result the total time required by both the stretched mesh and the EBNRBC, including the calculation for the steady state, is roughly equal whilst the Reference NRBC is the quickest over all.

The length of time for the EBNRBC to reach convergence is due to the response time of the filter. Investigations were performed on the current Butterworth bandpass filter to improve response times. The investigation, looked at in the previous section, focused primarily on widening the bandpass gap of the filter and results showed the wider the bandpass the faster the response times.

When widening the filter band gap it is seen that the EBNRBC run time is more competitive with the stretched mesh and when considering the run time of the steady case the total run time to obtain an accurate solution is less than that of the stretched mesh. A way in which one can further decrease the run time of the EBNRBC is to cut the calculation shorter as the filter responds to the incoming wave. It is not always necessary to reduce reflections by 40dB. So one can cut the calculation time by simply stopping the calculation at the desired level of accuracy. However it should be noted that the calculation should be left alone long enough such that reflections resulting from the filter response time can leave the domain. For example a reduction in reflection of 26dB can be achieved in 2000 timesteps however the initial reflections present due to the incorrect boundary condition as the filter responds are still present in the domain. This results in having to run the calculation long enough to ensure the reflected waves have left the domain. Taking all this into account the lowest number of timesteps achieved was approximately 4000. This was with a band pass gap of 0.2 and a reflection reduction of around 30dB. Further work is required to find the optimal filter for this type of calculation. It should be noted that while widening the band gap allows the user to reach a solution quicker it is not a suitable solution if one requires boundary conditions for multiple tones at closely spaced frequencies. In this case a tighter filter for each frequency will be required.

Chapter 6

Demonstration for Prediction of Wake/Stator Interaction Tone Noise

6.1 Three dimensional bladed turbo-machinery test case

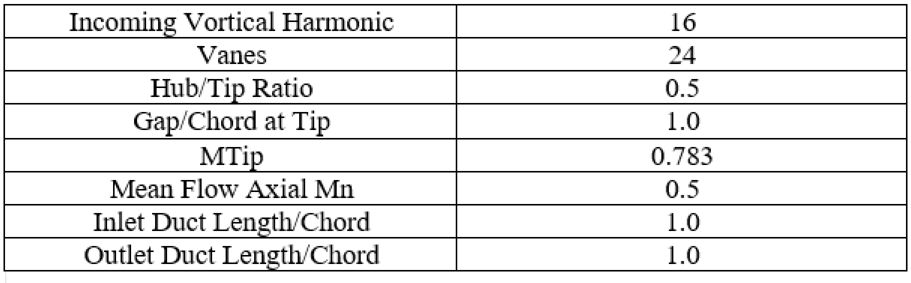
This test case was chosen to test the performance of the EBNRBC under more realistic conditions. The test case consists of a uniform annular duct containing 24 axially aligned flat plate stator blades. The wake interacting with the stators is modelled as a vortical mode introduced at the upstream boundary. The vortical wake interacts with the stators to produce upstream and downstream acoustic waves in Tyler-Sofrin modes [85]; That is Bessel-Fourier harmonics at the same frequency as the input vortical mode and circumferential mode number,

$$m = m_v \pm kV \tag{6.1}$$

where m_v is the mode number of the input vortical wave, k is an integer and V is the number of stators.

This is a simplified test case in that the mean flow is axial, there is no camber or thickness in the stator vanes and the mean flow is treated as inviscid however, the physics of the interaction between the vortical mode and the stator blades is preserved. The interaction also causes the vortical mode to scatter into upstream and downstream travelling acoustic waves with multiple modes in each direction as in a real case. These waves impinge on the upstream and downstream boundary and hence a non-reflecting boundary treatment is essential for correct noise acoustic prediction.

The geometry used was is taken from the category 4 benchmark problem from the third computational aeroacoustics workshop on bench mark problems (Cleveland, Ohio, Nov 8-10 1999) as described by Dahl [86]. The geometry of the test cases shown in table 6.1.



Incoming Vortical Harmonic	16
Vanes	24
Hub/Tip Ratio	0.5
Gap/Chord at Tip	1.0
MTip	0.783
Mean Flow Axial Mn	0.5
Inlet Duct Length/Chord	1.0
Outlet Duct Length/Chord	1.0

Table 6.1: The geometry of the wake-stator interaction test case. Mtip refers to the tip Mach number of a conceptual rotor generating the input vortical mode.

This test case has been used previously by Namba and Schulten [87] to test calculation methods and by Wilson to test boundary conditions. The original test case was designed with a high amplitude input vortical wave leading to acoustic waves of amplitudes approaching 160dB SPL [88]. At these high amplitudes non-linear effects can occur. Previous authors used linear methods which precluded these linear effects. In the current works a non-linear method is used and in order to avoid the complexities of non-linear effects the amplitude of the input signal was reduced such that the variation in circumferential velocity was 1% of mean flow axial velocity.

The input signal was a vortical wave at circumferential mode 16 and frequency $\omega = 4152.97$. At this frequency only 2 of the acoustic Tyler-Sofrin modes are cut-on which are the first and second radial harmonics at $m = -8$ [88]. Both of these modes are treated in the EBNRBC's.

The mesh was defined by 64 equispaced radial points and 768 equispaced circumferential points and 112 equispaced axial points. This gives almost 50 points per wavelength for the input vortical mode in both the circumferential and axial directions. The input wave was radially in phase meaning the radial discretisation is not relevant. The two cut-on acoustic modes were captured with nearly 100 points per wavelength circumferentially for both radial mode numbers and almost 40 points per wavelength axially for the first radial mode and almost 60 for the second radial mode. This discretisation is sufficient to ensure that discretisation errors in the CFD calculation are small.

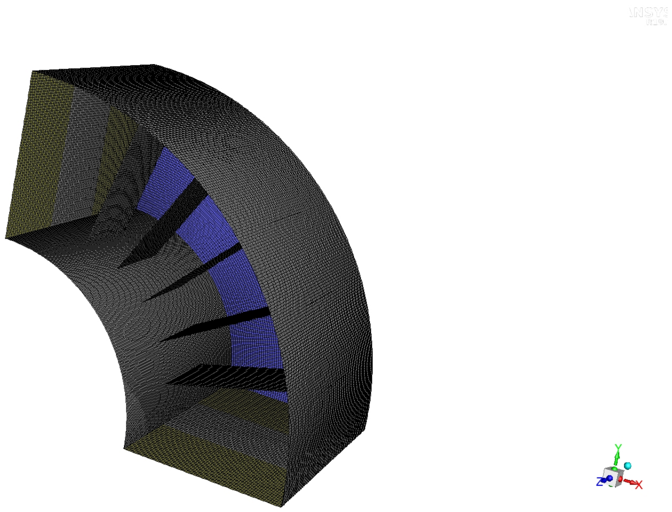


Figure 6.1: The three dimensional mesh used for the uniform flow test calculation. The outlet and interior mesh have been removed to allow proper visual of the blades.

The analysis of the results is split, where the domain before the fans is dubbed the upstream part of the duct and the domain after the domain is dubbed the downstream part of the duct. Here the acoustic signal, depending on the boundary condition can reflect off the domain boundaries and reflect partially again off the blades. This means in order to get accurate results, the boundary conditions need to be activated simultaneously and need to perform adequately at both the inlet and the outlet. As acoustic disturbances are travelling both upstream and downstream, any waves not properly treated at the boundary can reflect and constructively or destructively interact with the acoustic waves created due to the vortical input spoiling the solution.

In discussing the validation of the EBNRBC in its prediction of wake/stator interaction tone noise, wave splitting was performed at the upstream end of the duct before the blades and the downstream end of the duct after the blades, an example of what is meant being seen in figure 6.2.

A number of calculations were performed with a number of different boundary conditions:

- No NRBC (fixed total pressure at inlet and fixed static pressure at outlet).
- Fixed total pressure at inlet and Reference NRBC (RNRBC) at outlet.
- Fixed total pressure at inlet and EBNRBC at outlet.
- EBNRBC at both inlet and outlet.

In theory in applying the correct boundary condition downstream, the acoustic waves travel down the duct and leave the domain. The upstream travelling waves however reflect off the inlet boundary and a portion of the acoustic energy then reflects off the blades. The portion that travels through the blades then leaves the domain through the outlet contaminating the prediction of the generated sound power in the downstream direction. The same effect is seen when applying the RNRBC. It is expected therefore that the results in these cases will be less accurate than when applying the EBNRBC at both the inlet and outlet.

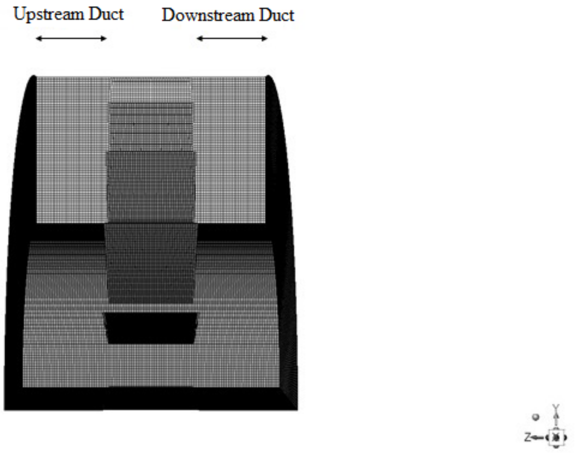


Figure 6.2: The wave splitting performed for analysis of the downstream boundary conditions were done in the downstream part of the duct, while the wave splitting for the upstream boundary conditions were done in the upstream part of the duct.

6.2 Results: Constant Pressure Boundary Condition

This boundary condition involves setting the total pressure at the inlet as well as the flow angles (as described in section 3.2.2) and static pressure at outlet. This boundary condition is reflective and thus gives spurious reflections that spoil the solution. Figure 6.3 shows the wave split modal amplitudes in the downstream portion of the duct. The two cut-on modes are represented by the lines at the top of the figure and the reflected mode amplitudes exactly match those of the incident modes therefore it can be seen that the constant pressure boundary condition is entirely reflective giving the wrong solution. The red lines sloping downward represent cut-off modes all of which decay strongly such that the influence on the result at the boundary is negligible. The green diagonal line represents the reflection of the strongest cut-off mode however the amplitude is too small to be significant.

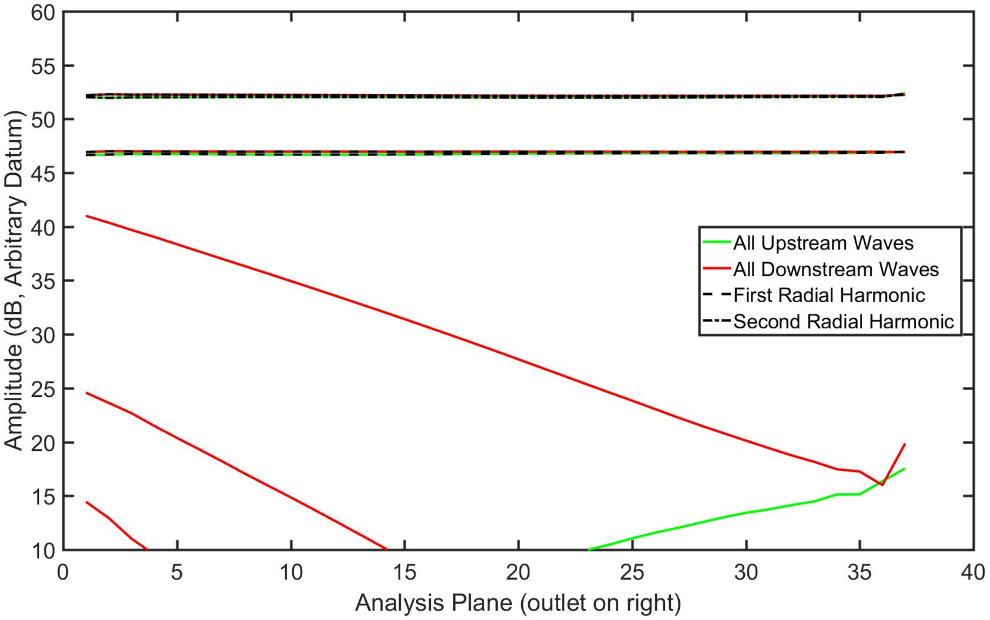


Figure 6.3: Analysis of the constant pressure boundary condition in the downstream portion of the duct. The lines at around 52dB represent the second radial harmonic, travelling in both the upstream and the downstream directions (upstream and downstream lines overlay exactly) while the lines at around 46dB represent the first radial harmonics travelling in both the upstream and downstream directions (again upstream and downstream lines overlay exactly).

Figure 6.4 shows the wave split modal amplitudes in the upstream portion of the duct. It can be seen that reflected wave is reduced in amplitude by about 5dB when compared to the incident wave in the second radial harmonic and the amplitude of the reflected wave is about 7dB lower than the incident wave in the first radial harmonic. Hence the fixed total pressure boundary condition is more non-reflective than the outlet boundary condition but still too reflective to give accurate results.

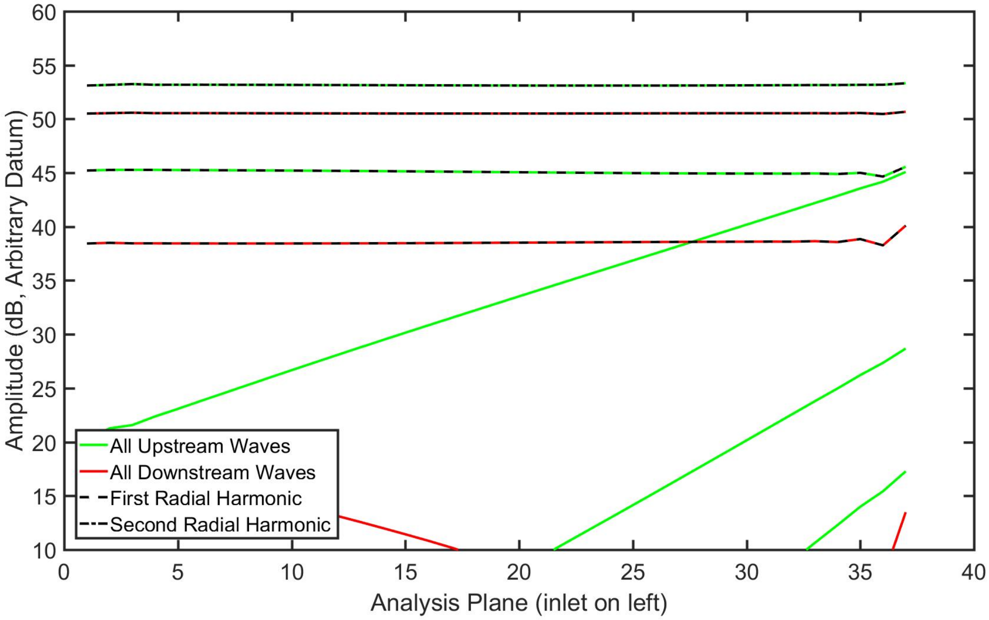


Figure 6.4: Analysis of the constant pressure boundary condition in the upstream portion of the duct.

6.3 Results: RNRBC

This boundary condition is limited to being applied only at the outlet and cannot be applied at the inlet, thus there is no active NRBC at the inlet of the duct, only the input vortical wave.

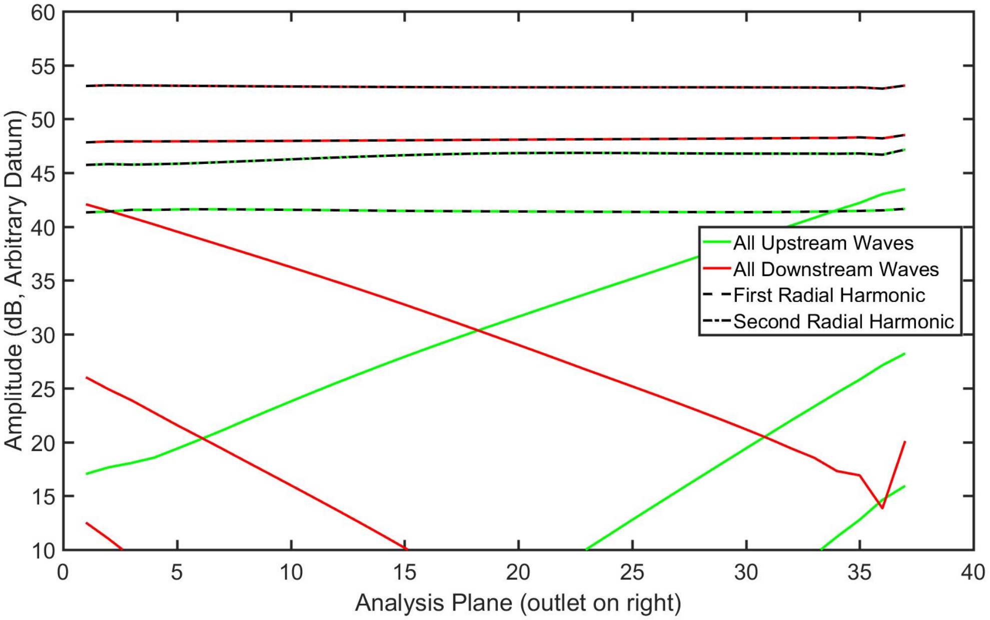


Figure 6.5: Analysis of the RNRBC in the downstream portion of the duct.

The reference NRBC here reduces reflections however its performance fails to match that of the performance seen in the validation in two dimensions and three dimensions. In the downstream portion of the duct the first radial mode is reduced in amplitude by roughly 6dB whilst the second radial mode is reduced by roughly 8dB. The reason for this is unclear but it is noted that in this test case unlike the test cases in chapter 4 and 5 there are residual vortical modes leaving the domain as well as acoustic waves. This may be the reason for the reduction in performance of the RNRBC but further work is required to confirm this.

At the upstream end of the duct a similar story is told to that of figure 6.4. There is no non reflective boundary and so the upstream acoustics are reflected back downstream with a small reduction in amplitude due to the same reason mentioned before.

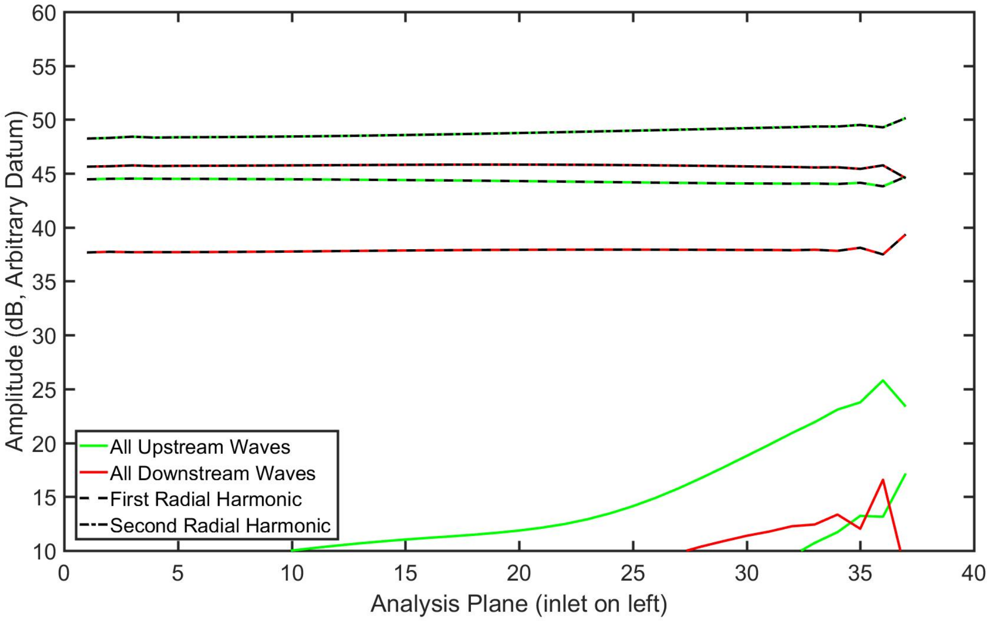


Figure 6.6: Analysis of the RNRBC in the upstream portion of the duct.

6.4 Results: EBNRBC at Downstream Boundary Only

In this case the NRBC was set only at the downstream boundary. As can be seen in figure 6.7, the first radial mode is reduced in reflections by about 39dB whilst the second radial mode is reduced in reflections by roughly 27.5dB. Again the upstream portion of the duct tells the same story as the previous cases.

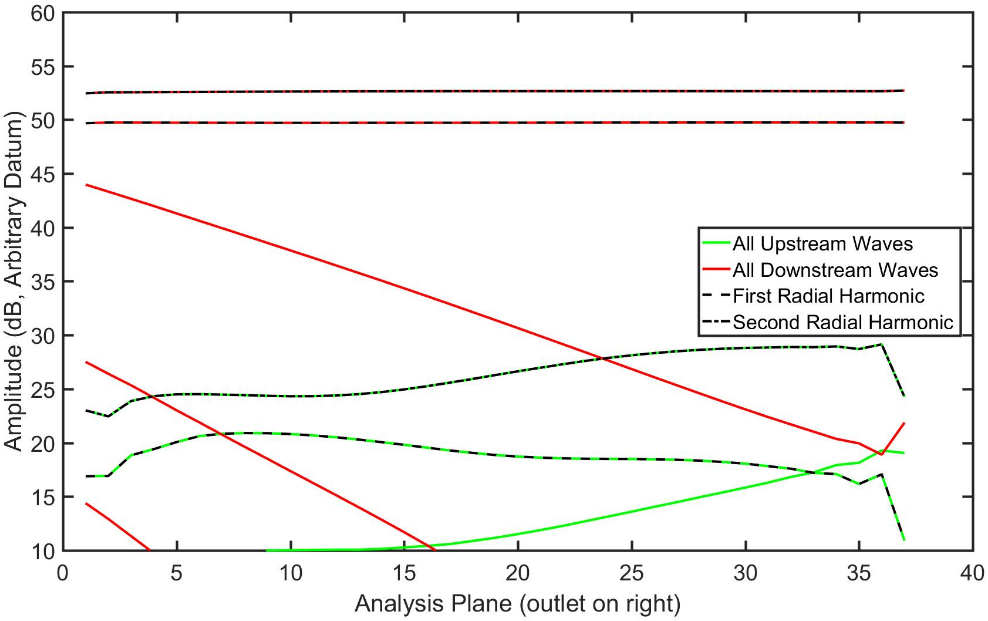


Figure 6.7: Analysis of the EBNRBC when applied only at the outlet in the downstream portion of the duct.

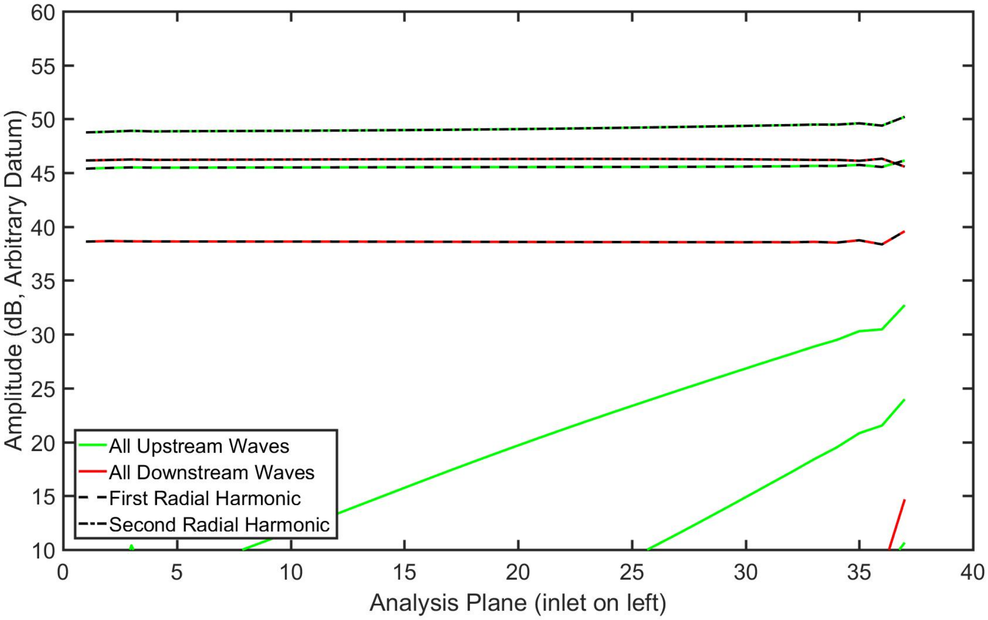


Figure 6.8: Analysis of the EBNRBC when applied only at the outlet in the upstream portion of the duct.

6.5 Results: EBNRBC at Downstream And Upstream Boundaries

In this case the EBNRBC was applied at both the inlet and the outlet, theoretically allowing for both upstream and downstream waves to exit the domain. In the downstream duct in figure 6.9 it can be seen that the first radial mode is reduced in amplitude by roughly 22dB while the second radial mode is reduced in reflection by roughly 18dB.

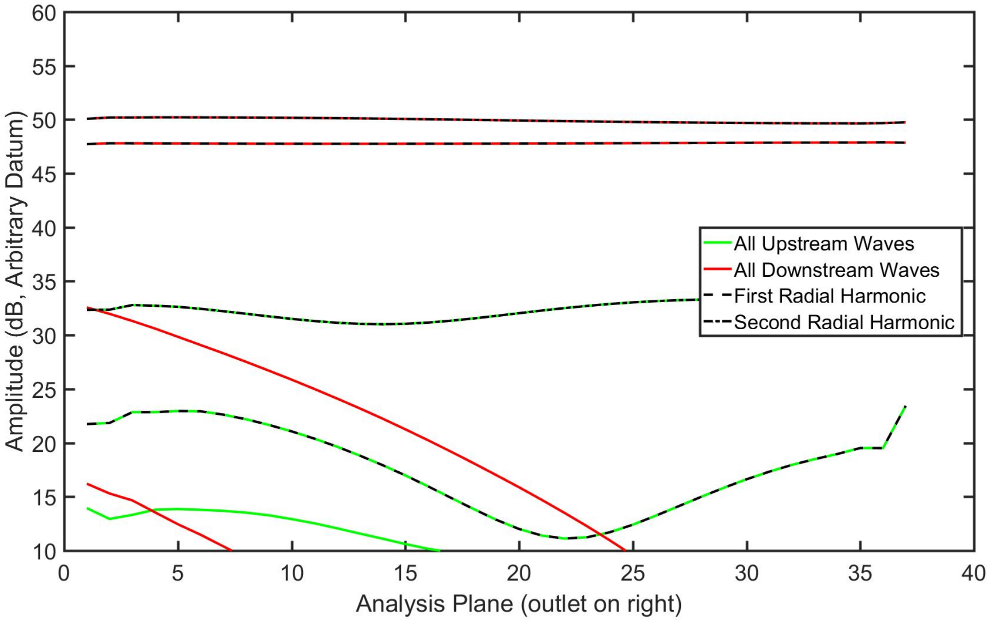


Figure 6.9: Analysis of the EBNRBC in the downstream portion of the duct when applied at both boundaries.

The upstream portion of the duct (figure 6.10) shows similar success in achieving an NRBC as the first radial mode is reduced in amplitude by 18dB whilst the second radial mode is reduced in amplitude by roughly 20dB.

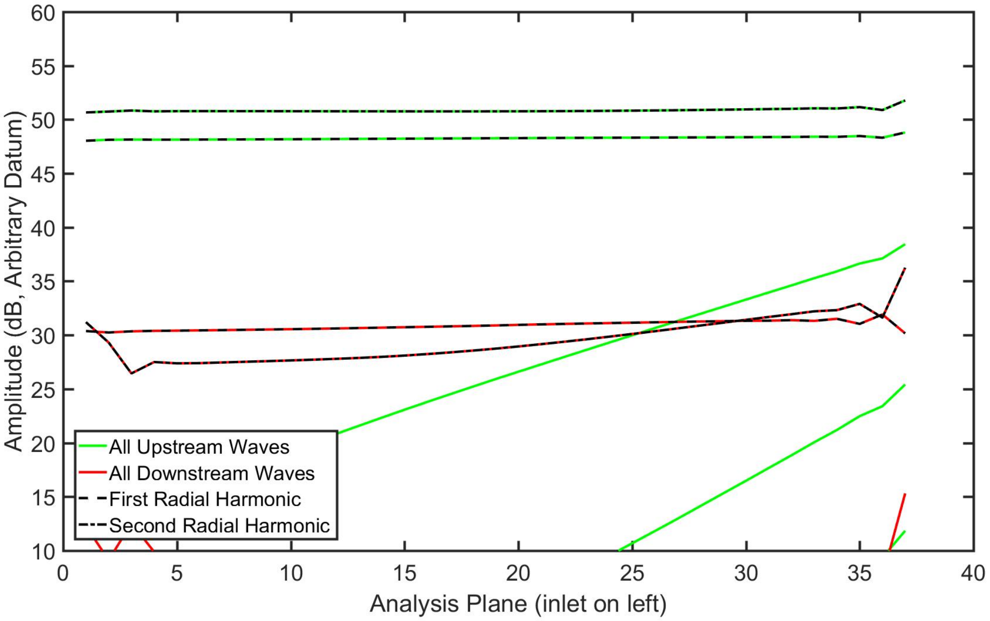


Figure 6.10: Analysis of the EBNRBC in the upstream portion of the duct when applied at both boundaries.

6.6 Results Summary: Blade Noise

The performance of the constant pressure boundary condition is inadequate. The reflections at the downstream end have the same amplitude as the incident waves and this effects the perceived amplitude of the outgoing waves at the upstream end (54dB in figure 6.4 compared with 51dB in figure 6.10). The fixed total pressure boundary condition at inlet is slightly better but still gives strong reflections. In this case these reflected waves travel through to the downstream part of the duct leading to an error in the outgoing wave (52dB in figure 6.3 compared with 50dB in figure 6.9).

The RNRBC, which has previously performed well, failed to give adequate results, and as mentioned is limited to being applied at the downstream boundary only, meaning the upstream signal is fully reflected.

The EBNRBC performs best here. When only the downstream boundary condition is active it reduces the signal by approximately 39dB and 27.5dB in both the first and second radial harmonics respectively at the downstream end of the duct. Of course the upstream end is still reflective and thus fails to let out the acoustic disturbance. When activated at both upstream and downstream boundaries the reduction in reflection is now 22dB and 18dB for the first and second radial modes at the downstream end of the duct while the upstream end of the duct reduces reflections by 18dB and 20dB in the first and second radial modes respectively showing good performance in allowing waves to leave the domain. The reason for the increase in reflection coefficient is not clear, but in both cases it is effective as an NRBC.

The results presented here show the importance of applying non-reflective boundary conditions at both the upstream and downstream boundaries. Although zonal boundary treatments were shown to work well in the three dimensional test cases in chapter 5 they were not tested here because of the necessity of applying an input vortical signal at the upstream boundary.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

A new two and three dimensional Non-Reflecting Boundary Condition (NRBC) has been developed for acoustic modes of discrete frequencies propagating through non-uniform ducted flows for the upstream and downstream boundary. The boundary conditions dubbed Eigen-Based Non-Reflecting Boundary Condition (EBNRBC) were developed and tested using a commercial CFD solver (ANSYS FLUENT) for many uniform and non-uniform flow cases in both two and three dimensions. The cases varied in Mach number and mode number and in the case of non-uniform flow, the mean flow profile. The EBNRBC when applied at the downstream boundary performed best when compared to alternative methods, reducing reflections to around 40dB below the incident wave amplitude while at the upstream boundary reflections are reduced to 20dB below the incident wave amplitude.

At the downstream boundary the EBNRBC generally performs better then the Reference NRBC (RNRBC) based on the method of Poinsot and Lele [2]. The RNRBC reduces reflections of cut-on waves to around 25dB below the incident wave amplitude which will be sufficient in many applications, however the RNRBC fails at cut-off frequencies and frequencies close to cut-off. In the wake/stator interaction test case the RNRBC fails to perform adequately only reducing reflections to around 6-8dB below the incident wave amplitude. The EBNRBC can also be applied at both the inlet boundary whilst the RNRBC can only be applied at the downstream boundary.

In uniform axial mean flow the EBNRBC reduces to the Giles NRBC [18]. In non-uniform mean flow the acoustic modes are no longer pure Bessel-Fourier modes. Treating the boundary using Giles NRBC with only the Bessel-Fourier mode closest in shape to the non-uniform flow mode was found to be insufficiently non-reflective for upstream travelling waves but performed adequately for downstream travelling waves where there is less harmonic scattering. The Giles NRBC cannot be applied in cases where the geometry is non-uniform.

The stretching mesh in the downstream cases performed nearly as well as the EBNRBC. The upstream stretching mesh is more effective in reducing reflections for upstream travelling waves however the calculation time of the EBNRBC is shorter in total, including calculation of the steady state, than the stretched mesh. It's also not possible to input an acoustic signal on the same boundary as the stretched mesh boundary treatment. Application of the stretched mesh also requires knowledge of the geometry just outside the domain which may not always be available.

In developing these boundary condition a bandpass filter was used. The filter however has a response time which slows the calculation time. Widening the band in the band pass filter can help improve response times and hence calculation times making this method more competitive with alternate treatments in terms of time taken to solve.

The choice of boundary condition is dependent on the particular application. The EBNRBC developed in this thesis would not be appropriate for the prediction of broadband noise or complex noise fields with large numbers of acoustic frequencies and wave numbers. In these cases the reference boundary condition works adequately at the downstream end. The reference boundary condition also does not require a preprocessing stage making it the most time efficient method. Domain extension boundary treatments such as a stretching mesh can be used at either end but cannot be used with an input acoustic or vortical signal.

The EBNRBC has been demonstrated to meet all the requirements laid out at the beginning of the thesis for a non-reflecting boundary treatment for the prediction of acoustic tones in ducted geometry:

- To be applicable at the inlet and outlet surface.
- To be applicable on the boundary surface itself.

- To allow for an acoustic or vortical input signal at the same plane as the non-reflecting boundary treatment.
- To provide accurate results in the presence of circumferentially (as well as radially) varying mean flow for both cut-on and cut-off incident waves.
- To be applicable to non-uniform duct geometry including non-planar boundary surfaces (this last capability has not been demonstrated in this thesis - see next section).

7.2 Future Work

The reference NRBC for the realistic test case in chapter 6 was less effective than for the simplified test cases in chapter 5. It was surmised that this was due to the residual vortical wave at the downstream boundary but further work is required to confirm this conclusion.

The Giles boundary condition was demonstrated in chapter 5 using a single Bessel-Fourier harmonic closest to that of the non-uniform mean flow mode. There is also the possibility of applying the Giles boundary condition with multiple harmonics to better cover the scattered acoustic field. Note that while a wide range of modes can be chosen relevant to the scattered mode being input, the relationship between the pressure and velocities in uniform flow is different to that in non-uniform flow thereby making the outgoing modal coefficients different to what one would have using the EAGCC method where the non-uniform flow is accounted for.

The response time of the filter used in the current method strongly effects the calculation time required to achieve convergence as discussed in chapter 5. Therefore alternative types of filters, such as the Chebyshev band pass filter, should be explored with the aim of finding shorter response times reducing overall compute cost.

The EBNRBC method, as developed in this thesis, is immediately applicable to acoustically lined ducts however this capability has not been demonstrated or validated in this thesis.

The EBNRBC method has been validated for non-uniform mean flow but not non-uniform duct geometry. In order to use the method with non-uniform geometry a small extension to the method is required to interpolate the flow data between the CFD and the EBNRBC meshes.

The eigenanalysis method underlying the EBNRBC is also applicable to external flows in the vicinity of upstream end of the duct as shown by Wilson [39]. The EBNRBC method as coded should work directly with this case. Further development work would be required, however, to extend the method to the region downstream of a turbomachinery duct, in view of the very localised shear layer associated with the exhaust jet.

The test cases in this thesis were all set up with a slip wall (inviscid) boundary condition. This will have negligible effect when the wall boundary layer is small compared to the acoustic wavelengths being predicted. The EBNRBC method, as developed in this thesis, can be applied also to calculations with no slip boundary condition however further demonstration/validation will be required to apply the method in these cases.

Acknowledgements

This work was funded by the Engineering and Physical Sciences Research Council and sponsored by Rolls-Royce as part of the Rolls-Royce University Technology Centre in Propulsion Systems Noise at the University of Southampton.

Bibliography

[1] D. L. Sutliff, "A 20 year retrospective of the advanced noise control fan–contributions to turbofan noise research," in *AIAA Propulsion and Energy 2019 Forum*, p. 3824, 2019.

[2] T. J. Poinsot and S. K. Lele, "Boundary-conditions for direct simulations of compressible viscous flows," *Journal of Computational Physics*, vol. 101, no. 1, pp. 104–129, 1992.

[3] S. W. Rienstra, "Sound transmission in slowly varying circular and annular lined ducts with flow," *Journal of Fluid Mechanics*, vol. 380, pp. 279–296, 1999.

[4] G. Kreiss, B. Krank, and G. Efraimsson, "Analysis of stretched grids as buffer zones in simulations of wave propagation," *Applied Numerical Mathematics*, vol. 107, pp. 1–17, 2016.

[5] A. Mani, "On the reflectivity of sponge zones in compressible flow simulations," *Center for Turbulence Research, Annual Research Briefs Stanford, CA*, pp. 117–133, 2010.

[6] S. G. Johnson, "Notes on perfectly matched layers (pmls)," *arXiv preprint arXiv:2108.05348*, 2021.

[7] D. Schlüß, C. Frey, and G. Ashcroft, "Consistent non-refleflecting boundary conditions for both steady and unsteady flow simulations in turbomachinery applictations," *Proceedings of the 7th Europ*, 2016.

[8] M. Bertsch, "Evaluation of impedance boundary conditions in ansys fluent," 2017.

[9] J. Anker, B. Schrader, U. Seybold, J. Mayer, and M. Casey, "A three-dimensional non-reflecting boundary condition treatment for steady-state flow simulations," in *44th AIAA Aerospace Sciences Meeting and Exhibit*, p. 1275, 2006.

[10] S. Medida, *Curvilinear extension to the Giles non-reflecting boundary conditions for wall-bounded flows.* PhD thesis, University of Toledo, 2007.

[11] J. D. Anderson and J. Wendt, *Computational fluid dynamics*, vol. 206. Springer, 1995.

[12] R. Eymard, T. Gallouët, and R. Herbin, "Finite volume methods," *Handbook of numerical analysis*, vol. 7, pp. 713–1018, 2000.

[13] J. N. Reddy, "An introduction to the finite element method," *New York*, vol. 27, 1993.

[14] B. E. CP985938RF and A. Majda, "Absorbing boundary conditions for the numerical simulation of waves," *Math. Comp*, vol. 31, p. 629, 1977.

[15] D. Givoli, "Non-reflecting boundary conditions," *Journal of computational physics*, vol. 94, no. 1, pp. 1–29, 1991.

[16] S. V. Tsynkov, "Numerical solution of problems on unbounded domains. a review," *Applied Numerical Mathematics*, vol. 27, no. 4, pp. 465–532, 1998.

[17] L. Ting and M. J. Miksis, "Exact boundary conditions for scattering problems," *The Journal of the Acoustical Society of America*, vol. 80, no. 6, pp. 1825–1827, 1986.

[18] M. Giles, *Non-reflecting boundary conditions for the Euler equations.* Computational Fluid Dynamics Laboratory, Department of Aeronautics and . . . , 1988.

[19] A. Sescu, "On deriving nonreflecting boundary conditions in generalized curvilinear coordinates," *arXiv preprint arXiv:1501.06392*, 2015.

[20] P. Moinier and M. B. Giles, "Eigenmode analysis for turbomachinery applications," *Journal of Propulsion and Power*, vol. 21, no. 6, pp. 973–978, 2005.

[21] S. W. Rienstra, *Cut-on, cut-off transition of sound in slowly varying flow ducts.* Citeseer, 2001.

[22] S. W. Rienstra, "Sound propagation in slowly varying lined flow ducts of arbitrary cross-section," *Journal of Fluid Mechanics*, vol. 495, pp. 157–173, 2003.

[23] N. Peake and A. J. Cooper, "Acoustic propagation in ducts with slowly varying elliptic cross-section," *Journal of Sound and Vibration*, vol. 243, no. 3, pp. 381–401, 2001.

[24] A. J. Cooper and N. Peake, “Propagation of unsteady disturbances in a slowly varying duct with mean swirling flow,” *Journal of Fluid Mechanics*, vol. 445, pp. 207–234, 2001.

[25] S. W. Rienstra, *Basics of Acoustics.* 2017.

[26] S. W. Rienstra, “Fundamentals of duct acoustics,” *Von Karman Institute Lecture Notes*, 2015.

[27] A. G. Wilson, “Propagation of acoustic perturbations in non-uniform ducts with non-uniform mean flow using eigen analysis in general curvilinear coordinate systems,” *Journal of Sound and Vibration*, vol. 443, pp. 605–636, 2019.

[28] H.-O. Kreiss, “Initial boundary value problems for hyperbolic systems,” *Communications on Pure and Applied Mathematics*, vol. 23, no. 3, pp. 277–298, 1970.

[29] H. O. Kreiss, “Initial boundary value problems for hyperbolic partial differential equations,” in *Proceedings of the Fourth International Conference on Numerical Methods in Fluid Dynamics*, pp. 22–33, Springer, 1975.

[30] R. L. Higdon, “Initial-boundary value problems for linear hyperbolic systems,” *SIAM review*, vol. 28, no. 2, pp. 177–217, 1986.

[31] L. Selle, F. Nicoud, and T. Poinsot, “Actual impedance of nonreflecting boundary conditions: Implications for computation of resonators,” *AIAA journal*, vol. 42, no. 5, pp. 958–964, 2004.

[32] K. A. Kousen, “Eigenmodes of ducted flows with radially-dependent axial and swirl velocity components,” 1999.

[33] M. B. Giles, “Eigenmode analysis of unsteady one-dimensional euler equations,” 1983.

[34] D. L. Darmofal, P. Moinier, and M. B. Giles, “Eigenmode analysis of boundary conditions for the one-dimensional preconditioned euler equations,” *Journal of Computational Physics*, vol. 160, no. 1, pp. 369–384, 2000.

[35] R. Beam, R. Warming, and H. C. Yee, “Stability analysis of numerical boundary conditions and implicit difference approximations for hyperbolic equations,” *Journal of Computational Physics*, vol. 48, no. 2, pp. 200–222, 1982.

[36] M. Giles and W. T. Thompkins, "Asymptotic analysis of numerical wave propagation in finite difference equations," tech. rep., Cambridge, Mass.: Gas Turbine & Plasma Dynamics Laboratory, Massachusetts . . . , 1983.

[37] R. Vichnevetsky, "Energy and group velocity in semi discretizations of hyperbolic equations," 1981.

[38] M. E. Goldstein, "Aeroacoustics," *mhi*, 1976.

[39] A. G. Wilson, "Calculation of acoustic transfer functions for smooth ducts of varying cross section and non-uniform mean flow using eigen analysis in general curvilinear coordinates," in *2018 AIAA/CEAS Aeroacoustics Conference*, p. 3776, 2018.

[40] A. G. Wilson, "Non-linear acoustic propagation in circumferentially non-uniform mean flow," in *25th AIAA/CEAS Aeroacoustics Conference*, p. 2448, 2019.

[41] P. J. Petrie-Repar, "Three-dimensional non-reflecting boundary condition for linearized flow solvers," in *ASME Turbo Expo 2010: Power for Land, Sea, and Air*, pp. 1247–1252, American Society of Mechanical Engineers, 2010.

[42] A. P. Saxer and M. B. Giles, "Quasi-three-dimensional nonreflecting boundary conditions for euler equations calculations," *Journal of Propulsion and Power*, vol. 9, no. 2, pp. 263–271, 1993.

[43] V. Ryaben'Kii, "Nonreflecting time-dependent boundary conditions on artificial boundaries of varying location and shape," *Applied Numerical Mathematics*, vol. 33, no. 1-4, pp. 481–492, 2000.

[44] L. Caretto, "Grid transformations for cfd calculations," 2002.

[45] A. Sommerfeld, "Lectures on theoretical physics, vol. i," *I (New York: Academic)*, 1964.

[46] L. N. Trefethen and L. Halpern, "Well-posedness of one-way wave equations and absorbing boundary conditions," *Mathematics of computation*, vol. 47, no. 176, pp. 421–435, 1986.

[47] F. Treves, *Introduction to pseudodifferential and Fourier integral operators Volume 2: Fourier integral operators*, vol. 2. Springer Science & Business Media, 1980.

[48] M. Ruzhansky, "Introduction to pseudo-differential operators," *Imperial College, London*, 2014.

[49] L. Halpern and L. N. Trefethen, "Wide-angle one-way wave equations," *The Journal of the Acoustical Society of America*, vol. 84, no. 4, pp. 1397–1404, 1988.

[50] M. J. Grote, "Nonreflecting boundary conditions for time dependent wave propagation," in *Research Report/Seminar für Angewandte Mathematik*, vol. 2000, Eidgenössische Technische Hochschule, Seminar für Angewandte mathematik, 2000.

[51] I. L. Sofronov, "Conditions for complete transparency on a sphere for a three-dimensional wave equation," in *Doklady Akademii Nauk*, vol. 326, pp. 953–957, Russian Academy of Sciences, 1992.

[52] I. L. Sofronov, "Artificial boundary conditions of absolute transparency for two-and three-dimensional external time-dependent scattering problems," *European Journal of Applied Mathematics*, vol. 9, no. 6, pp. 561–588, 1998.

[53] I. Sofronov, "Generation of 2d and 3d artificial boundary conditions transparent for waves outgoing to infinity," *preprint*, pp. 96–09, 1996.

[54] I. Sofronov, "Condition of absolute transparency on sphere for wave equation," *Finite Elements in Fluids: New Trends and Applications. CIMNE, Barcelona*, 1993.

[55] I. Sofronov, "Artificial boundary conditions which are adequate to the wave equation outside the sphere," *Keldysh Inst. Appl. Math., Russ. Acad. Sci*, 1992.

[56] B. Gustafsson, "Far-field boundary conditions for time-dependent hyperbolic systems," *SIAM journal on scientific and statistical computing*, vol. 9, no. 5, pp. 812–828, 1988.

[57] G. R. Putland, "Exact derivation of Kirchhoff's integral theorem and diffraction formula using high-school math," Apr. 2020.

[58] T. Colonius, "Modeling artificial boundary conditions for compressible flow," *Annu. Rev. Fluid Mech.*, vol. 36, pp. 315–345, 2004.

[59] S. V. Tsynkov, "Artificial boundary conditions for the numerical simulation of unsteady acoustic waves," *Journal of Computational Physics*, vol. 189, no. 2, pp. 626–650, 2003.

[60] V. Ryaben'kii, S. V. Tsynkov, and V. I. Turchaninov, "Global discrete artificial boundary conditions for time-dependent wave propagation," *Journal of Computational Physics*, vol. 174, no. 2, pp. 712–758, 2001.

[61] S. V. Tsynkov, "External boundary conditions for three-dimensional problems of computational aerodynamics," *SIAM Journal on Scientific Computing*, vol. 21, no. 1, pp. 166–206, 1999.

[62] K. W. Thompson, "Time dependent boundary conditions for hyperbolic systems," *Journal of computational physics*, vol. 68, no. 1, pp. 1–24, 1987.

[63] D. R. Hixon, "Improved mean flow boundary conditions for computational aeroacoustics," in *23rd AIAA/CEAS Aeroacoustics Conference*, p. 3503, 2017.

[64] K. W. Thompson, "Time-dependent boundary conditions for hyperbolic systems, ii," *Journal of computational physics*, vol. 89, no. 2, pp. 439–461, 1990.

[65] M. Myers, "On the acoustic boundary condition in the presence of flow," *Journal of Sound and Vibration*, vol. 71, no. 3, pp. 429–434, 1980.

[66] A. H. Nayfeh, *Introduction to perturbation techniques.* John Wiley & Sons, 2011.

[67] A. J. Cooper and N. Peake, "Propagation of unsteady disturbances in a slowly varying duct with mean swirling flow," *Journal of Fluid Mechanics*, vol. 445, pp. 207–234, 2001.

[68] V. V. Golubev and H. M. Atassi, "Acoustic–vorticity waves in swirling flows," *Journal of Sound and Vibration*, vol. 209, no. 2, pp. 203–222, 1998.

[69] R. Vichnevetsky, "Wave propagation and reflection in irregular grids for hyperbolic equations," *Applied numerical mathematics*, vol. 3, pp. 133–166, 1987.

[70] R. Vichnevetsky and L. H. Turner, "Spurious scattering from discontinuously stretching grids in computational fluid-dynamics," *Applied Numerical Mathematics*, vol. 8, no. 3, pp. 289–299, 1991.

[71] T. Colonius, S. K. Lele, and P. Moin, "Boundary conditions for direct computation of aerodynamic sound generation," *AIAA journal*, vol. 31, no. 9, pp. 1574–1582, 1993.

[72] S. Ta'asan and D. Nark, "An absorbing buffer zone technique for acoustic wave propagation," in *33rd Aerospace Sciences Meeting and Exhibit*, p. 164, 1995.

[73] F. Q. Hu, "On absorbing boundary conditions for linearized euler equations by a perfectly matched layer," 1995.

[74] J.-P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *Journal of computational physics*, vol. 114, no. 2, pp. 185–200, 1994.

[75] D. J. Bodony, "Analysis of sponge zones for computational fluid mechanics," *Journal of Computational Physics*, vol. 212, no. 2, pp. 681–702, 2006.

[76] W. C. Chew and W. H. Weedon, "A 3d perfectly matched medium from modified maxwell's equations with stretched coordinates," *Microwave and optical technology letters*, vol. 7, no. 13, pp. 599–604, 1994.

[77] R. Mittra and U. Pekel, "A new look at the perfectly matched layer (pml) concept for the reflectionless absorption of electromagnetic waves," *IEEE Microwave and Guided Wave Letters*, vol. 5, no. 3, pp. 84–86, 1995.

[78] C. Ma, "Perfect matched layer for acoustic wave equations,"

[79] D. Appelö, T. Hagstrom, and G. Kreiss, "Perfectly matched layers for hyperbolic systems: general formulation, well-posedness, and stability," *SIAM Journal on Applied Mathematics*, vol. 67, no. 1, pp. 1–23, 2006.

[80] F. Q. Hu, "Development of pml absorbing boundary conditions for computational aeroacoustics: A progress review," *Computers & Fluids*, vol. 37, no. 4, pp. 336–348, 2008.

[81] F. Q. Hu, X. D. Li, and D. K. Lin, "Absorbing boundary conditions for nonlinear euler and navier-stokes equations based on the perfectly matched layer technique," *Journal of Computational Physics*, vol. 227, no. 9, pp. 4398–4424, 2008.

[82] Q.-H. Liu and J. Tao, "The perfectly matched layer for acoustic waves in absorptive media," *The Journal of the Acoustical Society of America*, vol. 102, no. 4, pp. 2072–2082, 1997.

[83] E. Turkel and A. Yefet, "Absorbing pml boundary layers for wave-like equations," *Applied Numerical Mathematics*, vol. 27, no. 4, pp. 533–557, 1998.

[84] P. Moinier, M. B. Giles, and J. Coupland, "Three-dimensional nonreflecting boundary conditions for swirling flow in turbomachinery," *Journal of Propulsion and Power*, vol. 23, no. 5, pp. 981–986, 2007.

[85] J. M. Tyler and T. G. Sofrin, “Axial flow compressor noise studies,” tech. rep., SAE Technical Paper, 1962.

[86] J. Hardin, D. Huff, C. Tam, and M. Dahl, “Third computational aeroacoustics (caa) workshop on benchmark problems,” in *NASA Conference Publication*, vol. 209790, 2000.

[87] J. Schulten and M. Namba, “Numerical results of lifting surface theory-cat. 4 benchmark problem 3rd caa workshop,” 2000.

[88] A. Wilson, “Application of cfd to wake/aerofoil interaction noise-a flat plate validation case,” in *7th AIAA/CEAS Aeroacoustics Conference and Exhibit*, p. 2135, 2001.

Appendix A

EBNRBC Source Code

```
#include ”udf.h”
#include ”math.h”
#include ”mem.h”
#include ”/home/kz2e16/Boundary_Conditions/meshv1_transfer/compop.h”
#include <stdio.h>

#define MTC        2
#define NUMMOD     1
#define FOURCO     2
#define FILTOR     5
#define FILT       2
#define NOVAR      4
#define PI         3.14159265359
#define DTH        0.008181230869
#define BOUNDARY   2
#define NOR        64
#define MODE       -8
#define SEGNUM     4
#define rsh 1.39902416   /* Gamma (from FLUENT defaults) */
#define rgas 287.05  /* Rgas (SI units) */
#define qfac 0.0     /* Number of Phase changes from bottom to top (+ve
    => top leading) */
#define mna 0.5      /* Mean flow axial Mn */
#define mntip 0.783  /* Rotor tip tangential Mn */
#define nblades 16.0 /* Number of rotors */
#define uthdashouabar 0.01  /* Unsteady utheta normalised by mean ua */
```

```
#define tt 288.0      /* Inlet total temperature (K) */
#define ps 101325.0   /* Inlet static pressure (Pa) */
#define row 1.0       /* outer wall radius (m) */
#define riw 0.5       /* inner wall radius (m) */
#define pi 3.141592653589793
#define nr 64      /* Number of cells (NOT nodes!) in radial direction
    on the boundary */
#define nth 192    /* Number of cells (NOT nodes!) in circdirection on
    the boundary */
#define bdryid 20 /* Zone ID of boundary */

#define fname_indices  "/home/kz2e16/Boundary_Conditions/
    meshv1_transfer/DS_caa4mesh_v1muscl_bdry_indices.txt"    /* Filename
     for point number information */
#define fname_ps       "/home/kz2e16/Boundary_Conditions/
    meshv1_transfer/DS_caa4mesh_v1muscl_bdry_ps.txt"         /* Filename
     for steady pressure profile */
#define fname_jacob    "/home/kz2e16/Boundary_Conditions/
    meshv1_transfer/DS_caa4mesh_v1muscl_bdry_jacob.txt"      /* Filename
     for jacobian */
#define fname_invjacob "/home/kz2e16/Boundary_Conditions/
    meshv1_transfer/DS_caa4mesh_v1muscl_bdry_invjacob.txt"   /* Filename
     for inverse jacobian */

#define pointtol 1e-6 /* tolerance on calculated vs read in face
    centroids */

/*(4.9/400)*(2*pi)*/

static int modes[NUMMOD] = {0};
static Complex rev[NOR][NUMMOD][NOVAR][MTC][BOUNDARY]  = {{{{{{ 0
    }}}}}}; /* 0 = Downstream, 1 = Upstream */
static Complex lev[NOR][NUMMOD][NOVAR][MTC][BOUNDARY]  = {{{{{{ 0
    }}}}}}; /* 0 = Downstream, 1 = Upstream */
/*static double fnyquist = 0.5/1.570796326794896e-06; 400 steps/cycle
        */                    /*   //fnyquist = 0.5/timestep */
real omegas = 8.790547282462599e+03;
                                                               /*Angular
    frequency  9.252720619513942e-6 */
```

```
real c = 331.4952496191566;
                                                                                          /*
                //Speed of sound in air */
real M = 0.5;
int m = -8;

    /*Mode Number of the Signal */
float Rout = 1;
float Rin =  0.5;
static real ffilt[NOR][NUMMOD][FOURCO][NOVAR][FILTOR][FILT][BOUNDARY] =
     {{{{{{{{ 0 }}}}}}}};
static real fip[NOR][NUMMOD][FOURCO][NOVAR][BOUNDARY] = {{{{{{{ 0
    }}}}}}};
real aout[5] = {1, -3.995069131096948, 5.985706376640888,
    -3.986204158075490, 0.995566972065976};
real bout[5] = {0.246193054746015e-05, 0, -0.492386109492030e-05, 0,
    0.246193054746015e-05};
static Complex cco[NOR][NUMMOD][NOVAR][BOUNDARY] = {{{{{ 0 }}}}};
static Complex crev[NOR][NUMMOD][NOVAR][MTC][BOUNDARY] = {{{{{ 0 }}}}};
      /* DIMENSION 0 = pressure, 1 = Uz, 2 = Uth. BOUNDARY 0 = INLET, 1
    = OUTLET */
Complex phase = {0};
static Complex ocrev[NOR][NUMMOD][NOVAR][MTC][BOUNDARY] = {{{{{ 0
    }}}}};
Complex levc[BOUNDARY][MTC] = {{{0}}};
double totpressure = 18854.89;
real bdry_fc[nr][nth][3];
real bdry_ps[nr][nth];
real bdry_jacob[nr][nth][3][3];
real bdry_invjacob[nr][nth][3][3];




/*

    */
/*                                    CREATE ALL RELEVENT FUNCTIONS
                              */
/*
```

```
    */



/*

    */
/*                                  CREATE FILTER VARIABLES
                                     */
/*

    */

void testft(int boundary)
{
real RUTH = 0;
real UR = 0;
double x[ND_ND];
double x_1;
double y;
double z;
double theta;
int i;
int j;
int k;
int l;
int iradlp;
int id = 0;
int bound = 0;
double power = rsh/(rsh-1);
double profile = 0;
real machsquareds = 0;

#if !RP_HOST
if(boundary == 0)
{id = 17; bound = 2;}
else if(boundary == 1)
{id = 20; bound = 3;}
```

```
Domain *d;
face_t f;
Thread *thread;
d = Get_Domain(1);
thread = Lookup_Thread(d, id);

for( iradlp = 0; iradlp <= NOR-1; iradlp ++)
{
for( i = 0; i <= NUMMOD-1; i = i +1)
{
for( j = 0; j <= 1; j = j +1)
{
for( k = 0; k <= NOVAR-1; k = k +1)
{
fip[iradlp][i][j][k][boundary] = 0;
}
}
}
}


/* Step 1. Access current boundary values and decompose into fourier
   series fip  */


begin_f_loop(f, thread)
{
if(PRINCIPAL_FACE_P(f, thread))
{
    F_CENTROID(x,f,thread);
    x_1 = x[0];
    y = x[1];
    z = x[2];
    theta = atan2(y,x_1);
    int irad = 0;
    real rsquared = pow(x_1,2) + pow(y,2);
    float r = sqrt(rsquared);
    RUTH = -r*(-F_U(f,thread)*sin(theta) + F_V(f,thread)*cos(theta));
    UR = F_U(f,thread)*cos(theta) + F_V(f,thread)*sin(theta);
```

```
irad = floor(((r - Rin)/(Rout - Rin)*NOR));
profile = 0*5847.854485832853*exp((-theta*theta)/(2*0.5*0.5));
machsquareds = (pow((totpressure-profile+ps)/ps,(1/power)) - 1)
    *(2/(rsh - 1));
for( i = 0; i <= NUMMOD-1; i = i +1)
{
  if (modes[i]==0)
  {
    fip[irad][i][0][0][boundary] = fip[irad][i][0][0][boundary] +
       SEGNUM*(1/PI*(UR/c)*DTH);
    fip[irad][i][0][1][boundary] = fip[irad][i][0][1][boundary] +
       SEGNUM*(1/PI*(RUTH/c)*DTH);
    fip[irad][i][0][2][boundary] = fip[irad][i][0][2][boundary] +
       SEGNUM*(1/PI*((F_W(f, thread) - (c*machsquareds))/c)*DTH);
    fip[irad][i][0][3][boundary] = fip[irad][i][0][3][boundary] +
       SEGNUM*(1/PI*(F_UDMI(f, thread, bound)/(rsh*ps))*DTH);
    fip[irad][i][1][0][boundary] = 0;
    fip[irad][i][1][1][boundary] = 0;
    fip[irad][i][1][2][boundary] = 0;
        fip[irad][i][1][3][boundary] = 0;
  }
  else
  {
    fip[irad][i][0][0][boundary] = fip[irad][i][0][0][boundary] +
       SEGNUM*(1/PI*(UR/c)*cos(modes[i]*theta)*DTH);
    fip[irad][i][1][0][boundary] = fip[irad][i][1][0][boundary] +
       SEGNUM*(1/PI*(UR/c)*sin(modes[i]*theta)*DTH);
    fip[irad][i][0][1][boundary] = fip[irad][i][0][1][boundary] +
       SEGNUM*(1/PI*(RUTH/c)*cos(modes[i]*theta)*DTH);
    fip[irad][i][1][1][boundary] = fip[irad][i][1][1][boundary] +
       SEGNUM*(1/PI*(RUTH/c)*sin(modes[i]*theta)*DTH);
    fip[irad][i][0][2][boundary] = fip[irad][i][0][2][boundary] +
       SEGNUM*(1/PI*((F_W(f, thread) - (c*machsquareds))/c)*cos(
       modes[i]*theta)*DTH);
    fip[irad][i][1][2][boundary] = fip[irad][i][1][2][boundary] +
       SEGNUM*(1/PI*((F_W(f, thread) - (c*machsquareds))/c)*sin(
       modes[i]*theta)*DTH);
    fip[irad][i][0][3][boundary] = fip[irad][i][0][3][boundary] +
       SEGNUM*(1/PI*(F_UDMI(f, thread, bound)/(rsh*ps))*cos(modes[i
            ]*theta)*DTH);
             fip[irad][i][1][3][boundary] = fip[irad][i][1][3][boundary]
                 + SEGNUM*(1/PI*(F_UDMI(f, thread, bound)/(rsh*ps))*sin(
               modes[i]*theta)*DTH);
      }
    }
  }
}
  end_f_loop(f, thread)



 #if RP_NODE
for( iradlp = 0; iradlp <= NOR-1; iradlp ++)
{
for( i = 0; i <= NUMMOD-1; i = i +1)
{
for( j = 0; j <= 1; j = j +1)
{
for( k = 0; k <= NOVAR-1; k = k +1)
{
fip[iradlp][i][j][k][boundary] = PRF_GRSUM1(fip[iradlp][i][j][k][
   boundary]);
}
}
}
}
  #endif
#endif
}

void testfilter(int boundary){
int j;
int k;
int iradlp;
int llp;
int mlp;
int i;
```

```
testft(boundary);
/*
   ***************************************************************************
   */
/* Step 2. Apply bandpass filter in time
                                      */
/*
   ***************************************************************************
   */
#if !RP_HOST
aout[0] = 0;

/*Shift time history one step to the right and add current values from
   fip */

for( iradlp = 0; iradlp <= NOR-1; iradlp = iradlp +1)
{
for( i = 0; i <= NUMMOD-1; i = i +1)
{
for( j = 0; j <= 1; j = j +1)
{
for( k = 0; k <= NOVAR-1; k = k +1)
{
for( llp = FILTOR-1; llp >= 1; llp = llp -1)
{
for( mlp = 0; mlp <= 1; mlp = mlp +1)
{
ffilt[iradlp][i][j][k][llp][mlp][boundary]=ffilt[iradlp][i][j][k][llp
   -1][mlp][boundary];
}
}
ffilt[iradlp][i][j][k][0][0][boundary] = fip[iradlp][i][j][k][boundary
   ];
ffilt[iradlp][i][j][k][0][1][boundary] = bout[0]*ffilt[iradlp][i][j][k
   ][0][0][boundary];
for( llp = FILTOR-1; llp >= 1; llp = llp -1)
{
ffilt[iradlp][i][j][k][0][1][boundary] = ffilt[iradlp][i][j][k][0][1][
   boundary] - aout[llp]*ffilt[iradlp][i][j][k][llp][1][boundary] +
    bout[llp]*ffilt[iradlp][i][j][k][llp][0][boundary];
}
}
}
}
}
aout[0] = 1;

/*
    ************************************************************************
    */
/* Step 3. Convert fourier harmonics into single vector
                              */
/*
    ************************************************************************
    */


for( iradlp = 0; iradlp <= NOR-1; iradlp ++)
{
for( i = 0; i <= NUMMOD-1; i = i + 1)
{
for( k = 0; k <= NOVAR-1; k = k +1)
{
cco[iradlp][i][k][boundary].real = ffilt[iradlp][i][0][k][0][1][
    boundary];
cco[iradlp][i][k][boundary].imag = ffilt[iradlp][i][1][k][0][1][
    boundary];
}
}
}
#endif
}

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

```
    */
/*                                      INPUT SIGNAL
                                               */
/*

    */
/*
    *****************************************************************************
    */

void signaldata(int boundary){/* novar = 0 is Ur, 1 is Uth, 2 is Uz, 3
    is Pressure */

int iradlp;
int i;
int j;
for( iradlp = 0; iradlp <= NOR-1; iradlp++)
{
for( i = 0; i <= NUMMOD-1; i++)
{
for(j = 0; j <= NOVAR-1; j++)
{
for(l = 0; l <= MTC-1; l++)
{
crev[iradlp][i][j][l][boundary] = cmultcr(rev[iradlp][i][j][l][boundary
    ], 100/(rsh*ps));
}
}
}
}
}


/*
    *************************************************************************
    */
/*   Step 4. Apply left eigenvectors to determine ingoing mode
    amplitudes     */
/*
```

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


void ogv(int boundary){   /*ogv = Out Going Vector */
Complex funkcoeff[BOUNDARY][MTC] = {{{0}}};
Complex fc[BOUNDARY][MTC]        = {{{0}}};
Complex cfc[BOUNDARY][MTC]       = {{{0}}};
int iradlp;
int i;
int k;
int novar;
int l;
int j = 0;

if (boundary == 0)
  {j = 1;}
else if (boundary == 1)
  {j = 0;}

for( l = 0; l <= MTC-1; l++)
{
for( iradlp = 0; iradlp <= NOR-1; iradlp ++)
{
for( i = 0; i <= NUMMOD-1; i = i +1)
{
for( k = 0; k <= NOVAR-1; k = k +1)
{
levc[boundary][l] = cmult(lev[iradlp][i][k][l][j], cco[iradlp][i][k][
   boundary]);
funkcoeff[boundary][l].real = funkcoeff[boundary][l].real + levc[
   boundary][l].real;
funkcoeff[boundary][l].imag = funkcoeff[boundary][l].imag + levc[
   boundary][l].imag;
}
}
}
}
```

```
fc[boundary][0] = funkcoeff[boundary][0];
fc[boundary][1] = funkcoeff[boundary][1];
phase.real = cos(0.073126493278062);
phase.imag = sin(0.073126493278062);
cfc[boundary][0] = cmult(fc[boundary][0], phase);
cfc[boundary][1] = cmult(fc[boundary][1], phase);
for( l = 0; l <= MTC-1; l++)
{
for( iradlp = 0; iradlp <= NOR-1; iradlp ++)
{
for( i = 0; i <= NUMMOD-1; i++)
{
for( novar = 0; novar <= NOVAR-1; novar++)
{
ocrev[iradlp][i][novar][l][boundary] = cmult(cfc[boundary][l], rev[
    iradlp][i][novar][l][j]);
}
}
}
}
}




/*

    */
/*                                        SET UP CALCULATION
                                         */
/*

    */

DEFINE_ON_DEMAND(zerop)
{
int i;
int j;
int k;
int iradlp;
int llp;
int fm = -8;
for (i = 0; i <= NUMMOD-1; i++ )
{
  modes[i] = fm;
  fm++;
}

#if !RP_HOST
Domain *d1;
face_t f1;
Thread *t1;
d1 = Get_Domain(1);
int id1 = 20;
t1 = Lookup_Thread(d1, id1);
begin_f_loop(f1, t1)
{
if(PRINCIPAL_FACE_P(f1, t1))
{
F_UDMI(f1, t1, 3) = 0;
}
}
end_f_loop(f1,t1)
#endif

testft(1);

#if !RP_HOST

/*Shift time history one step to the right and add current values from
   fip */
for( iradlp = 0; iradlp <= NOR-1; iradlp ++)
{
for( i = 0; i <= NUMMOD-1; i = i +1)
{
for( j = 0; j <= 1; j = j +1)
{
```

```
for( k = 0; k <= NOVAR-1; k = k +1)
{
for( llp = FILTOR-1; llp >= 0; llp = llp -1)      /*Note reverse
    direction as shifting to the right*/
{
ffilt[iradlp][i][j][k][llp][0][1] = fip[iradlp][i][j][k][1];
ffilt[iradlp][i][j][k][llp][1][1] = 0;
}
}
}
}
}
#endif
#if !RP_NODE
Message("Success␣but␣PLEASE␣REMEMBER␣TO␣CHECK␣THE␣BOUNDARY␣FACE␣ID'S\n"
    );
#endif
}

/*                                      READ IN FILES
                                             */




DEFINE_ON_DEMAND(file_read)
{
int l;
int i;
int j;
int k;

FILE *fp, *fi, *fo, *fu, *fl, *fh, *fj, *fk;

fp = fopen("DSLEVr.txt","r"); fi = fopen("DSLEVi.txt","r"); fo = fopen(
    "DSREVr.txt","r"); fu = fopen("DSREVi.txt","r"); fl = fopen("USLEVr.
    txt","r"); fh = fopen("USLEVi.txt","r"); fj = fopen("USREVr.txt","r"
    ); fk = fopen("USREVi.txt","r");

      if(fp == NULL){Message("Unable␣to␣open␣'LEVr.txt'␣file");}
```

```
else if(fi == NULL){Message("Unable␣to␣open␣'LEVi.txt'␣file");}
else if(fo == NULL){Message("Unable␣to␣open␣'REVr.txt'␣file");}
else if(fu == NULL){Message("Unable␣to␣open␣'REVi.txt'␣file");}
else{
for( l = 0; l <= MTC-1; l++)
{
for( i = 0; i <= NOVAR-1; i++)
{
for( k = 0; k <= NUMMOD-1; k++)
{
for( j = 0; j <= NOR-1; j++)
{
fscanf (fp, "%le", &lev[j][k][i][l][0].real);
fscanf (fi, "%le", &lev[j][k][i][l][0].imag);
fscanf (fo, "%le", &rev[j][k][i][l][0].real);
fscanf (fu, "%le", &rev[j][k][i][l][0].imag);
fscanf (fl, "%le", &lev[j][k][i][l][1].real);
fscanf (fh, "%le", &lev[j][k][i][l][1].imag);
fscanf (fj, "%le", &rev[j][k][i][l][1].real);
fscanf (fk, "%le", &rev[j][k][i][l][1].imag);
}
}
}
}
fclose(fp); fclose(fi); fclose(fo); fclose(fu); fclose(fl); fclose(fh);
    fclose(fj); fclose(fk);
}

signaldata(0);
signaldata(1);

#if !RP_NODE
Message("Read␣in␣complete.\n");
#endif
}


/*                              WRITE FFILT DATA
                                  */
```

```
DEFINE_ON_DEMAND(WRITE_filter_data)
{
int i;
int j;
int k;
int iradlp;
int llp;
int mlp;
int boundary;
FILE *fp;
fp = fopen(”overwrite.txt”, ”r”);
if(fp == NULL){
#if !RP_NODE
Message(”\n␣Cannot␣write␣filter␣data.␣Please␣create␣an␣empty␣txt␣file␣
    titled␣’overwrite’\n”);
#endif
}
else
{fclose(fp);
for( iradlp = 0; iradlp <= NOR-1; iradlp = iradlp +1)
{
for( i = 0; i <= NUMMOD-1; i = i +1)
{
for( j = 0; j <= 1; j = j +1)
{
for( k = 0; k <= NOVAR-1; k = k +1)
{
for( llp = 0; llp <= FILTOR-1; llp = llp +1)
{
for( mlp = 0; mlp <= 1; mlp = mlp +1)
{
for( boundary = 0; boundary <= BOUNDARY-1; boundary = boundary +1)
{
node_to_host_real_1(ffilt[iradlp][i][j][k][llp][mlp][boundary]);

}
}
}
}
}
}
}
#if !RP_NODE
FILE *filter;
filter = fopen(”filter_data.txt”,”w”);
if(filter == NULL){Message(”Unable␣to␣create␣’filter_data’␣file”);}
else{
for( iradlp = 0; iradlp <= NOR-1; iradlp = iradlp +1)
{
for( i = 0; i <= NUMMOD-1; i = i +1)
{
for( j = 0; j <= 1; j = j +1)
{
for( k = 0; k <= NOVAR-1; k = k +1)
{
for( llp = 0; llp <= FILTOR-1; llp = llp +1)
{
for( mlp = 0; mlp <= 1; mlp = mlp +1)
{
for( boundary = 0; boundary <= BOUNDARY-1; boundary = boundary +1)
{
printf(”ffilt[%d][%d][%d][%d][%d][%d][%d]␣WRITE␣%e\n”, iradlp, i, j, k,
     llp, mlp, boundary, ffilt[iradlp][i][j][k][llp][mlp][boundary]);

scanf(”%le”, &ffilt[iradlp][i][j][k][llp][mlp][boundary]);
fprintf(filter, ”%le␣\n”, ffilt[iradlp][i][j][k][llp][mlp][boundary]);
}
}
}
}
}
}
}
Message(”Write␣complete.\n”);
}
fclose(filter);
Message(”Done\n␣PLEASE␣DELETE␣OVERWRITE␣FILE”);
```

```
#endif
}
}

/*

    */
/*                                    READ FFILT DATA
                                         */
/*

    */

DEFINE_ON_DEMAND(READ_filter_data)
{
int i;
int j;
int k;
int iradlp;
int llp;
int mlp;
int boundary;
FILE *filter;
filter = fopen("filter_data.txt","r");
if(filter == NULL){Message("Unable to create 'filter_data' file");}
else{
for( iradlp = 0; iradlp <= NOR-1; iradlp = iradlp +1)
{
for( i = 0; i <= NUMMOD-1; i = i +1)
{
for( j = 0; j <= 1; j = j +1)
{
for( k = 0; k <= NOVAR-1; k = k +1)
{
for( llp = 0; llp <= FILTOR-1; llp = llp +1)
{
for( mlp = 0; mlp <= 1; mlp = mlp +1)
{
for( boundary = 0; boundary <= BOUNDARY-1; boundary = boundary +1)
{
fscanf(filter, ”%le”, &ffilt[iradlp][i][j][k][llp][mlp][boundary]);
printf(”ffilt[%d][%d][%d][%d][%d][%d][%d]␣READ␣%e\n”, iradlp, i, j, k,
    llp, mlp, boundary, ffilt[iradlp][i][j][k][llp][mlp][boundary]);
}
}
}
}
}
}
}
#if !RP_NODE
Message(”Read␣complete.\n”);
#endif
}
fclose(filter);
}


DEFINE_ON_DEMAND(print)
{
int i;
int j;
int k;
int iradlp;
int llp;
int mlp;
int boundary;
for( iradlp = 0; iradlp <= NOR-1; iradlp = iradlp +1)
{
for( i = 0; i <= NUMMOD-1; i = i +1)
{
for( j = 0; j <= 1; j = j +1)
{
for( k = 0; k <= NOVAR-1; k = k +1)
{
for( llp = 0; llp <= FILTOR-1; llp = llp +1)
{
for( mlp = 0; mlp <= 1; mlp = mlp +1)
```

```
{
for( boundary = 0; boundary <= BOUNDARY−1; boundary = boundary +1)
{
node_to_host_real_1(ffilt[iradlp][i][j][k][llp][mlp][boundary]);

}
}
}
}
}
}
}
#if !RP_NODE
for( iradlp = 0; iradlp <= NOR−1; iradlp = iradlp +1)
{
for( i = 0; i <= NUMMOD−1; i = i +1)
{
for( j = 0; j <= 1; j = j +1)
{
for( k = 0; k <= NOVAR−1; k = k +1)
{
for( llp = 0; llp <= FILTOR−1; llp = llp +1)
{
for( mlp = 0; mlp <= 1; mlp = mlp +1)
{
for( boundary = 0; boundary <= BOUNDARY−1; boundary = boundary +1)
{
printf(”ffilt[%d][%d][%d][%d][%d][%d][%d].real␣%e\n”, iradlp, i, j, k,
    llp, mlp, boundary, ffilt[iradlp][i][j][k][llp][mlp][boundary]);

}
}
}
}
}
}
}
#endif
}
```

```
/*

    */
/*                                    RUN CALCULATION
                                       */
/*

    */




DEFINE_EXECUTE_AT_END(FilterVariables)
{
testfilter(0);
testfilter(1);
}


/*
    *****************************************************************************
    */
/*

    */
/*                                   APPLICATION OF NRBC
                                  */
/*

    */
/*
    *****************************************************************************
    */
DEFINE_PROFILE(OUTLET_3dnrbc, thread, position)
{
  face_t f;

      /*Atmospheric Pressure (1 atm)*/
  real t = CURRENT_TIME;

     /*Angular frequeny normalised by speed of sound c

     Duct Radius */
  double p = 0;
  double signal = 0;

     /*Mode Number of the Signal */
  double profile;
  double power = rsh/(rsh-1);
  double pin = 0;
  double uz = 0;
  double uth = 0;
  double ur = 0;
  double x[ND_ND];
  double x_1;
  double y;
  double z;
  double theta;
  real scalar;
  real machsquareds;
  real machsquaredi;
  double pres;
  int i;
  int l;


  #if !RP_HOST
  ogv(1);
/*
   ***********************************************************************
   */
/* Step 5. Apply right eigenvectors to set boundary values from
   outgoing modes*/
/*        Currently hardwired for exit boundary condition
                        */
/*
   ***********************************************************************
```

```
    */

begin_f_loop(f, thread)
{
if(PRINCIPAL_FACE_P(f, thread))
{
F_CENTROID(x,f,thread);
x_1 = x[0];
y = x[1];
z = x[2];
theta = atan2(y,x_1);
real rsquared = pow(x_1,2) + pow(y,2);
real r = sqrt(rsquared);
int irad = 0;
profile = 0*5847.854485832853*exp((-theta*theta)/(2*0.5*0.5));
machsquareds = (pow((totpressure-profile+ps)/ps,(1/power)) - 1)*(2/(rsh
    - 1));
pin = 0;
uz = 0;
uth = 0;
ur = 0;
p = 0;
irad = floor(((r - Rin)/(Rout - Rin)*NOR));
for( l = 0; l <= MTC-1; l++)
{
for( i = 0; i <= NUMMOD-1; i = i +1)
{
pin = pin + crev[irad][i][3][l][1].real*cos(modes[i]*theta-omegas*t) +
    crev[irad][i][3][l][1].imag*sin(modes[i]*theta-omegas*t);
uz  = uz  + crev[irad][i][2][l][1].real*cos(modes[i]*theta-omegas*t) +
    crev[irad][i][2][l][1].imag*sin(modes[i]*theta-omegas*t);
uth = uth + crev[irad][i][1][l][1].real*cos(modes[i]*theta-omegas*t) +
    crev[irad][i][1][l][1].imag*sin(modes[i]*theta-omegas*t);
ur  = ur  + crev[irad][i][0][l][1].real*cos(modes[i]*theta-omegas*t) +
    crev[irad][i][0][l][1].imag*sin(modes[i]*theta-omegas*t);
p   = p   + ocrev[irad][i][3][l][1].real*cos(modes[i]*theta) + ocrev[
    irad][i][3][l][1].imag*sin(modes[i]*theta);
}
}
machsquaredi = machsquareds + pow(uz,2) + pow(uth/r,2) + 2*sqrt(
    machsquareds)*uz + pow(ur,2);
scalar = pow((1 + ((rsh - 1)/2)*machsquaredi),power);
signal = pin*rsh*ps;
pres = p*rsh*ps + 0*signal;
F_PROFILE(f, thread, position) = pres;
F_UDMI(f, thread, 3) = pres;
fflush(stdout);
}
}
end_f_loop(f, thread)
#endif
}


DEFINE_PROFILE(INLET_3dnrbc, thread, position)
{
face_t f;
real t = CURRENT_TIME;
double p = 0;
double signal = 0;
double profile;
double power = rsh/(rsh-1);
double pin = 0;
double uz = 0;
double uth = 0;
double ur = 0;
double x[ND_ND];
double x_1;
double y;
double z;
double theta;
real scalar;
real machsquareds;
real machsquaredi;
real p_stat;
real p_totg;
real p_tot;
real tsmean;
real c;
real omega;
real rfac;
real duth;
real dua;
real uusqogrtt;
real umn;
real upt;
double pres = 0;
int i;
int l;

#if !RP_HOST /* MOVE THIS */
ogv(0);

/*
   **************************************************************************
   */
/* Step 5. Apply right eigenvectors to set boundary values from ingoing
     modes */
/*          Currently hardwired for exit boundary condition
                         */
/*
   **************************************************************************
   */

begin_f_loop(f, thread)
{
if(PRINCIPAL_FACE_P(f, thread))
{
F_CENTROID(x,f,thread);
int irad = 0;
x_1 = x[0];
y = x[1];
z = x[2];
theta = atan2(y,x_1);
real rsquared = pow(x_1,2) + pow(y,2);
real r = sqrt(rsquared);
profile = 0*5847.854485832853*exp((-theta*theta)/(2*0.5*0.5));
```

```
machsquareds = (pow((totpressure-profile+ps)/ps,(1/power)) - 1)*(2/(rsh
     - 1));
pin = 0;
uz = 0;
uth = 0;
ur = 0;
p = 0;
irad = floor(((r - Rin)/(Rout - Rin)*NOR));
for( l = 0; l <= MTC-1; l++)
{
for( i = 0; i <= NUMMOD-1; i = i +1)
{
pin = pin + crev[irad][i][3][l][0].real*cos(modes[i]*theta-omegas*t) +
    crev[irad][i][3][l][0].imag*sin(modes[i]*theta-omegas*t);
uz  = uz  + crev[irad][i][2][l][0].real*cos(modes[i]*theta-omegas*t) +
    crev[irad][i][2][l][0].imag*sin(modes[i]*theta-omegas*t);
uth = uth + crev[irad][i][1][l][0].real*cos(modes[i]*theta-omegas*t) +
    crev[irad][i][1][l][0].imag*sin(modes[i]*theta-omegas*t);
ur  = ur  + crev[irad][i][0][l][0].real*cos(modes[i]*theta-omegas*t) +
    crev[irad][i][0][l][0].imag*sin(modes[i]*theta-omegas*t);
p   = p   + ocrev[irad][i][3][l][0].real*cos(modes[i]*theta) + ocrev[
    irad][i][3][l][0].imag*sin(modes[i]*theta);
}
}
tsmean=tt*(1-(rsh-1)/2*mna*mna);
c=sqrt(rsh*rgas*tsmean);
omega=nblades*mntip*c/row;
rfac=(r-riw)/(row-riw)-1;  /* So from -1 to 0 */
duth=uthdashouabar*mna*c*cos(omega*t-nblades*theta+qfac*2*pi*rfac);
dua = duth*mna/(mntip*r/row);        /* Pure gust at each height in 2D */
uusqogrtt=((mna*c-dua)*(mna*c-dua)+duth*duth)/rsh/rgas/tt;
umn=sqrt(uusqogrtt/(1-(rsh-1)/2*uusqogrtt));
upt=ps*pow((1+(rsh-1)/2*umn*umn),(rsh/(rsh-1)));
machsquaredi = machsquareds + pow(uz,2) + pow(uth/r,2) + 2*sqrt(
    machsquareds)*uz + pow(ur,2);
scalar = pow((1 + ((rsh - 1)/2)*machsquaredi),power);
p_stat = ps + pin*rsh*ps;
p_totg = p_stat*scalar;
p_tot = p_totg - ps;
```

```
signal = p_tot;
pres = p*rsh*ps + 0*totpressure - 0*profile + 0*signal + upt-ps;

F_PROFILE(f, thread, position) = pres;
F_UDMI(f, thread, 2) = pres;
fflush(stdout);
}
}
end_f_loop(f, thread)
#endif
}

DEFINE_PROFILE(INLET_steady_pressure, thread, position)
{
face_t f;

   /*        //Mode Number of the Signal */
double profile;
double x[ND_ND];
double x_1;
double y;
double z;
double theta;
#if !RP_HOST
begin_f_loop(f, thread)
{
F_CENTROID(x,f,thread);
x_1 = x[0];
y = x[1];
z = x[2];
theta = atan2(y,x_1);
profile = 5847.854485832853*exp((-theta*theta)/(2*0.5*0.5));
                                    /*       //Defining the non-uniform
   pressure distribution around the circumference.*/
F_PROFILE(f, thread, position) = 20127.46324385585 - profile;
   }
  end_f_loop(f, thread)
  #endif
}
```

```
DEFINE_PROFILE(INLET_temperature, thread, position)
{
face_t f;
real p_totin;
double profile;
double x[ND_ND];
double x_1;
double y;
double z;
double theta;
real t_s = 300/(1+((rsh-1)/2)*pow(0.5,2));
real t_tot;
#if !RP_HOST
begin_f_loop(f, thread)
{
if(PRINCIPAL_FACE_P(f, thread))
{
F_CENTROID(x,f,thread);
x_1 = x[0];
y = x[1];
z = x[2];
theta = atan2(y,x_1);
profile = 5847.854485832853*exp((-theta*theta)/(2*0.5*0.5));
p_totin = 20127.46324385585-profile;
t_tot = t_s*pow((p_totin+ps)/ps,(rsh-1)/rsh);
F_PROFILE(f, thread, position) = t_tot;
}
}
end_f_loop(f, thread)
#endif
}




DEFINE_PROFILE(INLET_UZ, thread, position)
{
face_t f;
```

```
real t = CURRENT_TIME;
double uzd = 0;
double uzu = 0;
double profile;
double power = rsh/(rsh-1);
double x[ND_ND];
double x_1;
double y;
double z;
double theta;
real machsquared;
real mach;
int i;
int l;
real tsmean;
real cs;
real omega;
real rfac;
real duth;
real dua;
#if !RP_HOST
ogv(0);
begin_f_loop(f, thread)
{
if(PRINCIPAL_FACE_P(f, thread))
{
F_CENTROID(x,f,thread);
int irad = 0;
x_1 = x[0];
y = x[1];
z = x[2];
theta = atan2(y,x_1);
profile = 0*5847.854485832853*exp((-theta*theta)/(2*0.5*0.5));
machsquared = (pow((totpressure-profile+ps)/ps,(1/power)) - 1)*(2/(rsh
   - 1));
mach = sqrt(machsquared);
uzd = 0;
uzu = 0;
real rsquared = pow(x_1,2) + pow(y,2);
real r = sqrt(rsquared);
irad = floor(((r - Rin)/(Rout - Rin)*NOR));
for( l = 0; l <= MTC-1; l++)
{
for( i = 0; i <= NUMMOD-1; i = i +1)
{
uzd = uzd + crev[irad][i][2][l][0].real*cos(modes[i]*theta-omegas*t) +
    crev[irad][i][2][l][0].imag*sin(modes[i]*theta-omegas*t);
uzu = uzu + ocrev[irad][i][2][l][0].real*cos(modes[i]*theta) + ocrev[
    irad][i][2][l][0].imag*sin(modes[i]*theta);
}
}
tsmean=tt*(1-(rsh-1)/2*mna*mna);
cs=sqrt(rsh*rgas*tsmean);
omega=nblades*mntip*cs/row;
rfac=(r-riw)/(row-riw)-1;  /* So from -1 to 0 */
duth=uthdashouabar*mna*cs*cos(omega*t-nblades*theta+qfac*2*pi*rfac);
dua=duth*mna/(mntip*r/row);       /* Pure gust at each height in 2D */
F_PROFILE(f, thread, position) = mach*c - dua + 0*uzd*c + uzu*c;
}
}
end_f_loop(f, thread)
#endif
}

DEFINE_PROFILE(INLET_UTH, thread, position)
{
face_t f;
real t = CURRENT_TIME;
double uthd = 0;
double uthu = 0;
double x[ND_ND];
double x_1;
double y;
double z;
double theta;
int i;
int l;
real tsmean;
real cs;
real omega;
real rfac;
real duth;
#if !RP_HOST
ogv(0);
begin_f_loop(f, thread)
{
if(PRINCIPAL_FACE_P(f, thread))
{
F_CENTROID(x,f,thread);
int irad = 0;
x_1 = x[0];
y = x[1];
z = x[2];
theta = atan2(y,x_1);
real rsquared = pow(x_1,2) + pow(y,2);
real r = sqrt(rsquared);
uthd = 0;
uthu = 0;
irad = floor(((r - Rin)/(Rout - Rin)*NOR));
for( l = 0; l <= MTC-1; l++)
{
for( i = 0; i <= NUMMOD-1; i = i +1)
{
uthd = uthd + crev[irad][i][1][l][0].real*cos(modes[i]*theta-omegas*t)
    + crev[irad][i][1][l][0].imag*sin(modes[i]*theta-omegas*t);
uthu = uthu + ocrev[irad][i][1][l][0].real*cos(modes[i]*theta) + ocrev[
    irad][i][1][l][0].imag*sin(modes[i]*theta);
}
}
tsmean=tt*(1-(rsh-1)/2*mna*mna);
cs=sqrt(rsh*rgas*tsmean);
omega=nblades*mntip*cs/row;
rfac=(r-riw)/(row-riw)-1;  /* So from -1 to 0 */
duth=uthdashouabar*mna*cs*cos(omega*t-nblades*theta+qfac*2*pi*rfac);
F_PROFILE(f, thread, position) =  -0*(uthd*c/r) - (uthu*c/r) + duth;
}
}
```

```
end_f_loop(f, thread)
#endif
}

DEFINE_PROFILE(INLET_UR, thread, position)
{
face_t f;
real t = CURRENT_TIME;
double urd = 0;
double uru = 0;
double x[ND_ND];
double x_1;
double y;
double z;
double theta;
int i;
int l;
#if !RP_HOST
ogv(0);
begin_f_loop(f, thread)
{
if(PRINCIPAL_FACE_P(f, thread))
{
F_CENTROID(x,f,thread);
int irad = 0;
x_1 = x[0];
y = x[1];
z = x[2];
theta = atan2(y,x_1);
real rsquared = pow(x_1,2) + pow(y,2);
real r = sqrt(rsquared);
urd = 0;
uru = 0;
irad = floor(((r - Rin)/(Rout - Rin)*NOR));
for( l = 0; l <= MTC-1; l++)
{
for( i = 0; i <= NUMMOD-1; i = i +1)
{
urd = urd + crev[irad][i][0][0].real*cos(modes[i]*theta-omegas*t) +
    crev[irad][i][0][0].imag*sin(modes[i]*theta-omegas*t);
uru = uru + ocrev[irad][i][0][1][0].real*cos(modes[i]*theta) + ocrev[
    irad][i][0][1][0].imag*sin(modes[i]*theta);
}
}
F_PROFILE(f, thread, position) = 0*(urd*c) + (uru*c);
}
}
end_f_loop(f, thread)
#endif
}









DEFINE_EXECUTE_AFTER_DATA(bdry_setup,libname)
{
Domain *mydomain;
Thread *tf;
FILE *fp;
FILE *fp2;
face_t f;
real x, y, z;
real xc[3];
real dxsq, dysq, dzsq, mindistsq, temp;
int i, j, ixi, ixj, id1, id2;

fp = fopen(fname_indices, ”r”);
for (i = 0; i < nr; i++)
{
for (j = 0; j < nth; j++)
{
fscanf(fp, ”%lg␣%lg␣%lg”, &x, &y, &z);
bdry_fc[i][j][0] = x;
bdry_fc[i][j][1] = y;
```

```
bdry_fc[i][j][2] = z;
/* Message(”%03d %03d %f %f %f \n”, i, j, bdry_fc[i][j][0], bdry_fc[i][
    j][1], bdry_fc[i][j][2]); */
}
}
fclose(fp);

Message(”Indices␣Read␣\n”);

/* Now identify closest point and save i and j indices in F_UDMI */
/*NB this routine isnt optimised at all, as only need to run it at the
    start of each run*/
mydomain = Get_Domain(1);
tf = Lookup_Thread(mydomain, bdryid);
begin_f_loop(f, tf)
{
F_CENTROID(xc, f, tf);
mindistsq = 1e8;                    /* Start with impossibly high distance
    */
ixi = -1;
ixj = -1;
for (i = 0; i < nr; i++)
{
for (j = 0; j < nth; j++)
{
dxsq = (bdry_fc[i][j][0] - xc[0])*(bdry_fc[i][j][0] - xc[0]);
dysq = (bdry_fc[i][j][1] - xc[1])*(bdry_fc[i][j][1] - xc[1]);
dzsq = (bdry_fc[i][j][2] - xc[2])*(bdry_fc[i][j][2] - xc[2]);
if (dxsq + dysq + dzsq < mindistsq)
{
mindistsq = dxsq + dysq + dzsq;
ixi = i;
ixj = j;
}
}
}
F_UDMI(f, tf, 0) = (double)ixi;
F_UDMI(f, tf, 1) = (double)ixj;
/* Message(”%f %f %f %f %f %f %f %f \n”, F_UDMI(f, tf, 0), F_UDMI(f, tf
    , 1), xc[0], xc[1], xc[2], mindistsq, xc[0], xc[1]); */
}
end_f_loop(f, tf)

Message("Index␣Matching␣Complete␣\n");

/*  Read boundary pressures */
fp = fopen(fname_ps, "r");
for (i = 0; i < nr; i++)
{
for (j = 0; j < nth; j++)
{
fscanf(fp, "%lg", &temp);
bdry_ps[i][j] = temp;
/* Message("%03d %03d %f \n", i, j, bdry_ps[i][j]); */
}
}
fclose(fp);

Message("Pressures␣Read␣\n");

/*  Read jacobian and invjacobian arrays */
fp = fopen(fname_jacob, "r");
fp2 = fopen(fname_invjacob, "r");
for (id1 = 0; i < 3; id1++)
{
for (id2 = 0; i < 3; id2++)
{
for (i = 0; i < nr; i++)
{
for (j = 0; j < nth; j++)
{
fscanf(fp, "%lg", &temp);
bdry_jacob[i][j][id1][id2] = temp;
fscanf(fp2, "%lg", &temp);
bdry_invjacob[i][j][id1][id2] = temp;
/* Message("%03d %03d %f \n", i, j, bdry_ps[i][j]); */
}
}
```

```
}
}
fclose(fp);
fclose(fp2);

Message("Jacobian␣Arrays␣Read␣\n");

}

DEFINE_PROFILE(ps_exit_steady_profile, tf, ivar)
{

/* Thread *tf; */
face_t f;
int i;
int j;

begin_f_loop(f, tf)
{
i = (int)F_UDMI(f, tf, 0);
j = (int)F_UDMI(f, tf, 1);
F_PROFILE(f, tf, ivar) = bdry_ps[i][j];
}
end_f_loop(f, tf)

}
```

A.1 Custom Made Header File For Operations With Complex Numbers

```
#include "udf.h"
#include "math.h"
#include "mem.h"
#include "complex.h"

typedef struct COMPLEX {
        double real;
        double imag;
}Complex;
```

```
Complex cadd(Complex x,Complex y){

        Complex z;
        z.real = x.real + y.real;
        z.imag = x.imag + y.imag;

        return z;

}
Complex caddcr(Complex x, double y){

        Complex z;
        z.real = x.real + y;
        z.imag = x.imag;

        return z;

}

Complex cmin(Complex x,Complex y){

        Complex z;
        z.real = x.real - y.real;
        z.imag = x.imag - y.imag;

        return z;

}
Complex cmincr(Complex x, double y){

        Complex z;
        z.real = x.real - y;
        z.imag = x.imag;

        return z;

}
```

```
Complex cmult(Complex x,Complex y){

        Complex z;
        z.real = x.real * y.real - x.imag * y.imag;
        z.imag = x.real * y.imag + x.imag * y.real;

        return z;

}
Complex cmultcr(Complex x, double y){

        Complex z;
        z.real = x.real * y;
        z.imag = x.imag * y;

        return z;

}
```