# A Multidomain Spectral Method For Computational Aeroacoustics

Dan Stanescu

A Thesis

in

The Department

of

Mechanical Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University
Montréal, Québec, Canada

February 1999

0-612-39028-4

# Abstract

## A Multidomain Spectral Method For Computational Aeroacoustics

**Dan Stanescu, Ph.D.**
**Concordia University, 1999**

This thesis presents a method for computational aeroacoustics, primarily aimed at computing sound propagation in, and radiation from, turbofan inlets. The physics of sound propagation is modeled by the system of partial differential equations that describe conservation of mass, momentum and energy in inviscid flows. The equations are solved numerically in the time domain as an initial and boundary value problem to obtain the time-dependent acoustic pressure in the flow field, from which sound pressure levels are obtained by integration.

A multidomain spectral method is used to discretize the space terms. Complex geometries are handled by the use of unstructured grids of non-overlapping hexahedra that may have curved boundaries. An isoparametric mapping is used to transform each hexahedron on the master element, on which an efficient collocation spectral approximation can be defined by the use of tensor products. Continuity of the solution in space is enforced as part of the solution process by the use of a set of staggered grids that do not involve the element corners.

A set of Runge-Kutta methods optimized for wave propagation and with minimal storage requirements are developed for integration in time. Several radiation boundary conditions are implemented and tested, and a way to construct the spectral grids within the elements starting from given edge descriptions is proposed. A transformation that alleviates the time step restriction while keeping the exponential accuracy is also discussed. Numerical results that validate the methodology are presented for several test cases representative of the fan noise problem. The thesis ends with a brief description of a modification that allows computation of noise superposed on a mean flow known from other sources, such as experiments, and its application to turbulent mixing noise from a supersonic jet.

# Acknowledgements

This work would not have been possible without the help received from Prof. W.G. Habashi, who not only took the chance of having a graduate student working on 'strange' (such as finite volume or spectral) methods, but also supported his participation to several CFD and Aeroacoustics forums that were most beneficial. It would not have been possible, as well, without the funding received through a series of NSERC and FCAR Graduate and Aerospace Scholarships and the funding from Pratt & Whitney Canada.

Many a times, advice from the colleagues at the CFD Lab, among whom Dr. D. Ait-Ali-Yahia, Dr. G. Baruzzi, M. Bogstad and Dr. Y. Bourgault, proved to be the only way around difficulties. Special thanks are due to Dr. L. Dutto for providing her excellent nonlinear GMRES solver, and to C. Lepage, who not only never failed in LaTeX, but also had a good advice whenever nothing else seemed to work. Last, but certainly not least, thanks are due to Prof. D. Kopriva from Florida State University, who imparted some useful advice on spectral methods and also provided an useful expression parser, and Prof. W.S. Don from Brown University, who suggested the way to implement filters in the algorithm.

*This thesis is dedicated to my wife, Adelaida.*

# Contents

# List of Figures

# List of Tables

# Nomenclature

## English symbols

| | |
|---|---|
| $A$ | Array of Runge-Kutta coefficients in usual Butcher notation |
| $\mathcal{A}$ | Amplitude of acoustic pressure perturbations |
| $B$ | Number of fan blades |
| $\mathcal{B}$ | Boundary operator |
| $\mathcal{C}$ | Courant-Friedrichs-Lewy number |
| $\mathrm{C}$ | Characteristic path for uni-dimensional Euler equations |
| $\mathcal{D}$ | Computational domain |
| $D$ | Generic element in $\mathcal{D}$ |
| $\mathbb{D}$ | Master element $[-1,1]^d$ |
| $\mathbf{D}$ | Differentiation matrix |
| $E$ | Number of elements in $\mathcal{D}$ |
| $E_m$ | Radial eigenfunction for spinning mode of azimuthal order $m$ |
| $\mathcal{E}$ | Set of points on a grid edge |
| $F$ | Flux tensor in physical space, components $(F_1, F_2, F_3)$ |
| $\tilde{F}$ | Flux tensor in computational space, components $(\tilde{F}_1, \tilde{F}_2, \tilde{F}_3)$ |
| $\mathfrak{F}$ | Set of points on a grid face |

| | |
|---|---|
| $\mathbb{G}$ | Set of Gauss-Chebyshev points $\bar{\xi}^j$, $j = 1, \ldots, N$ |
| $H$ | Source vector for axisymmetric flow |
| $I$ | Unit matrix |
| $\mathbf{I}$ | Interpolation matrix |
| $\Im$ | Imaginary part |
| $J_m$ | Bessel function of first kind of order $m$ |
| $\mathcal{J}$ | First Jacobian of transformation from computational space $\vec{\xi}$ to physical space $\vec{x}$ |
| $K_i$ | Intermediate values of derivatives in the Runge-Kutta discretization, usual form |
| $L$ | Reference length |
| $\mathbb{L}$ | Set of Gauss-Chebyshev-Lobatto points $\xi^j$, $j = 0, 1, \ldots, N$ |
| $\mathfrak{L}$ | Length of a grid edge |
| $M$ | Mach number |
| $N$ | Number of Gauss-Chebyshev points |
| $\mathfrak{N}$ | Number of degrees of freedom in the discretization |
| $\mathcal{N}$ | Fan rotational speed (revolutions per minute) |
| $\mathcal{O}$ | Order of a truncation error |

| | |
|---|---|
| $\mathcal{P}$ | Root mean square pressure |
| $Q$ | Vector of conserved quantities in physical space |
| $\tilde{Q}$ | Vector of conserved quantities in computational space |
| $\mathcal{R}$ | Reynolds number |
| $\mathbb{R}$ | Real numbers set |
| $\mathrm{R}$ | Riemann invariant in locally isentropic flow |
| $\Re$ | Real part |
| $\mathbf{S}$ | Smoothing matrix |
| $T_j(x)$ | Chebyshev polynomial of degree $j$ |
| $V$ | Number of stator vanes |
| $\vec{V}$ | Velocity, components $(v_1, v_2, v_3)$ or $(u, v, w)$ interchangeably |
| $Y_m$ | Bessel function of second kind of order $m$ |
| $a$ | Wave speed for linear convection equation |
| $a_{ij}$ | Runge-Kutta array coefficients, entries of the matrix $A$ |
| $b_i$ | Runge-Kutta stage coefficients |
| $c$ | Sound speed |
| $c_i$ | Runge-Kutta coefficients for interstage advancement of independent variable, in this thesis time |

| | |
|---|---|
| $d$ | Number of spatial dimensions |
| $e$ | Element index |
| $h_i$ | Lagrange interpolant based on the points in $\mathbb{L}$ |
| $\bar{h}_i$ | Lagrange interpolant based on the points in $\mathbb{G}$ |
| $k_x, k_y$ | Wave numbers in the respective directions |
| $\vec{n}$ | Normal vector |
| $n_c$ | Filter cut-off frequency |
| $o$ | Order of Runge-Kutta time discretization |
| $o_f$ | Order of spatial filter |
| $p$ | Pressure |
| $\bar{p}$ | Mean pressure |
| $q$ | Numerical approximation to $Q$ |
| $r(\mathbf{z})$ | Amplification factor of the Runge-Kutta method |
| $r_e(\mathbf{z})$ | Exact amplification factor for the ordinary differential equation governing time advancement |
| $s$ | Number of stages of the Runge-Kutta method |
| $t$ | Time |
| $\vec{x}$ | Position vector, components $(x_1, x_2, x_3)$ or $(x, y, z)$ in Cartesian, $(x, r)$ in axisymmetric |

| | |
|---|---|
| $\Delta t$ | Time step |
| $\Delta x_m$ | Minimum spacing, $\mathcal{O}(N^{-2})$ for Chebyshev methods |

**Greek symbols**

| | |
|---|---|
| $\alpha_i, \beta_i$ | Low-storage Runge-Kutta coefficients |
| $\beta$ | Exponent for the power law defining the variable damping |
| $\gamma$ | Specific heats ratio |
| $\delta_{ij}$ | Kronecker symbol |
| $\epsilon$ | Total energy per unit mass |
| $\varepsilon$ | Machine precision |
| $\theta$ | Far field directivity angle |
| $\lambda$ | Acoustic wavelength |
| $\nu$ | Kinematic viscosity |
| $\vec{\xi}$ | Position vector in computational space, components $(\xi, \eta, \zeta)$ |
| $\varpi$ | Parameter in the Kosloff–Tal-Ezer transform |
| $\rho$ | Density |
| $\varrho$ | Filter function |
| $\sigma(\vec{x})$ | Variable damping in the damping layer |
| $\sigma_M$ | Maximum value of the damping parameter |

| | |
|---|---|
| $\varsigma$ | Cut-off ratio |
| $\upsilon$ | Integer switch for axisymmetric flow in two spatial dimensions |
| $\chi$ | Kosloff–Tal-Ezer transform coordinate |
| $\omega$ | Sound source angular frequncy |
| $\Gamma$ | Boundary of $\mathcal{D}$ |
| $\Theta, \Lambda, \Xi$ | Linear interpolation functions for transfinite interpolation |
| $\Upsilon$ | Amplitude error factor for Runge-Kutta methods |
| $\Psi$ | Phase error factor for Runge-Kutta methods |

**Subscripts**

| | |
|---|---|
| $L$ | Left state for Riemann problem |
| $R$ | Right state for Riemann problem |
| $b$ | Value at a boundary point |
| $d$ | Value interpolated from the interior of the domain |
| $m$ | Azimuthal order of a spinning mode |
| $o$ | Outlet value |
| $\infty$ | Far field value |
| $\mu$ | Radial order of a spinning mode |

**Superscripts**

| | |
|---|---|
| $T$ | Transpose |

# Chapter 1

# Introduction

Soon after the introduction of jet engines, it became clear that their successful use on commercial aircraft was largely dependent on the possibility of reducing the noise associated with the high subsonic Mach number turbulent jets which for those early designs were the only source of thrust. Research aimed at understanding the basic mechanisms of jet noise led to the development of the so-called 'acoustic analogy' by Lighthill [1] and the dawn of a new discipline, aeroacoustics. By applying a dimensional analysis to the turbulent sources in the jet shear layer, Lighthill showed in a subsequent paper [2] that the acoustic power radiated by a subsonic jet is proportional to the eighth power of the jet exit velocity. This most important theoretical result contributed to the evolution of the subsonic jet engine from the small area, high exit velocity noisy jets to the modern turbofans of large bypass ratio.

The acoustic analogy, as well as its later generalizations to account for the presence of solid surfaces in the flow, rely on a specification of the acoustic sources and as such are not completely deductive methods. If the sources are known, from experiments or numerical computations for example, the radiated field can be calculated by performing a volume integral over the source domain. The recent increases in computer power and advances in computational methods have opened the way for the computation of sound sources from first principles, using the fundamental equations of fluid dynamics. Thus, largely relying on experience gained from computational fluid dynamics (CFD), computational aeroacoustics (CAA) has emerged as a novel approach to aeroacoustics problems. Its aim, as stated by Tam [3] is "the application of computational methods to aeroacoustics problems for the purpose of understanding the physics of noise generation and propagation".

An algorithm for CAA must take into consideration aspects that are not usually encountered in CFD, such as the inherent unsteadiness of the problem, the wide range of time and length scales and the correct propagation of high-frequency acoustic waves of very small amplitude over large distances and across the boundaries of the computational domain. It is not hard to appreciate that no single method can ever be designed to perfectly suit the whole array of problems encountered in aeroacoustics. The object of the present thesis is to design an algorithm for the study of noise propagation through fan inlets and their near acoustic field. However, it will be shown that other problems, as for example turbulent mixing noise from jets, can be addressed with only minor modifications to the proposed algorithm.

## 1.1 Fan noise mechanisms

For the large bypass ratio turbofans used on commercial aircraft nowadays, the jet is not necessarily the most important source of noise in all flight regimes. Instead, during an approach flight, and even during take-off for high values of the bypass ratio, the fan is usually the dominant source due to the shielding effect of the unheated low-speed secondary jet flow on the core jet noise. Both broadband noise, mainly due to random turbulent disturbances interacting with the fan, and tone noise (rotor-alone and rotor-stator interaction noise) at multiples of the blade passage frequency characterize the fan acoustic signature. Moreover, multiple pure tones, another component of fan noise, occur at supersonic tip speeds due to the presence of a rotor-locked shock system [4] and to blade geometry irregularities arising from either manufacturing or assembly [5]. As a complete deterministic approach to broadband noise cannot be envisaged due to its stochastic nature, the present thesis is limited to the study of tone noise. This is also motivated by the fact that the broadband noise has a relatively low level over most of the frequency spectrum, whereas the drastic increase in noise intensity at the discrete frequencies involved in tonal noise represents a main contribution to community noise exposure.

In a frame of reference that rotates with the fan, when fluid flows through the bypass duct, the periodicity of the blades generates a pressure distribution that repeats itself $B$ times around the circumference, $B$ being the number of fan blades. This pressure distribution, which can be thought of as a $B$-lobed pattern, rotates with the fan generating periodic flow disturbances superposed on the mean duct flow. As opposed to propellers, for which the pressure disturbances propagate in free space, in the case of the fan the duct acts as a waveguide within which the sound field is made up of a discrete series of acoustic eigenmodes. The propagation characteristics of these modes depend on the driving frequency, and a landmark paper by Tyler and Sofrin [6] showed how this can be restated in terms of the rotor tip speed. As long as the tip speed is subsonic with respect to the ambient speed of sound, the generated modes are evanescent and decay exponentially with axial distance from the source, or equivalently they are cut-off by the duct. However, when the tip speed is supersonic, which is usually the case at high power settings (take-off and climb), the generated modes can propagate through the duct without decay if no acoustic treatment is applied to the walls. The resulting sound field is known as 'rotor-alone noise' and the lobed pressure patterns that sweep the duct walls along helical curves while propagating axially are commonly termed 'spinning modes'. Since the tip speed is supersonic, the spinning pattern may also contain in this case a rotor-locked shock system which can produce very intense acoustic radiation. The far-field noise spectrum has in this case discrete peaks at all the multiples of the shaft frequencies [4].

For an approach flight the engines usually run at a lower power setting, such that the fan tip speed is in the subsonic range and the rotor-alone noise is cut-off. The rotating pressure field and the fan viscous wakes impinge, however, on the nearby stator, while the stator potential flow field is periodically disturbed by the rotor blades. These are the main mechanisms generating rotor-stator interaction noise. A completely similar interaction takes place between the rotor and the struts. The theory developed by Tyler and Sofrin [6] also provides a qualitative description of the interaction noise when the number of vanes of the stator, $V$, and the number of blades of the rotor, $B$, are given. The basic interaction mechanisms can only generate patterns with $s = nB + kV$ lobes, where $k$ and $n$ are arbitrary integers. Depending on the particular combination of $k$ and $n$, these patterns may spin with higher speed than the rotor and hence can become cut-on. In practice, $B$ and $V$ can be chosen such that the fundamental frequency (also called the blade passage frequency, $B\mathcal{N}/60$) is cut-off. Higher harmonics usually are not, since this would require impractically large values for $B$ and $V$. Due to the much longer time of exposure of the environment to approach noise than to take-off noise which results from the small slope of the glide path, a better control of the rotor-stator and rotor-strut interaction noise is an essential factor for modern turbofan design to meet increasingly stringent noise regulations.

## 1.2 Fan noise prediction methods

In principle one can use the fluid dynamics equations to solve for the disturbances generated by the rotor, their interaction with the nearby components, and their propagation to the far field as noise, simultaneously. As turbulence should be accounted for and the geometries involved are quite complicated, such an approach is obviously impractical. It is therefore useful to consider the two components of the problem, source modeling and duct propagation and radiation, separately, with the understanding that various simplifying assumptions allow in some cases their simultaneous resolution.

Most source models currently in use for the fan are based on two-dimensional (2D) analyses, and have been developed soon after the Tyler-Sofrin [6] theory. Blades are usually considered cascades of flat plates, and the linearized equations for conservation of mass and momentum in compressible flows are used to describe the acoustic field. The disturbances corresponding to the fan blade wakes are considered to be in the form of gusts convected downstream by the mean flow, usually given by empirical correlations. Solutions can be obtained under these conditions for the unsteady pressure jump across the blades [7], which is equivalent to specifying a chord-wise distribution of acoustic dipole sources. Coupling with the duct of the unsteady distributions of both the rotor and the stator determined with proper phase and amplitude easily allows calculation of the propagating modes.

Three-dimensional (3D) effects play an important role on the sound generated by a blade row, in particular when it interacts with disturbances that have large radial variations. A model for the interaction of a rotating annular cascade in an uniform subsonic flow with circumferentially periodic distortions was developed by Namba [8]. He found that, as a rule, under the assumptions of the method, the 2D model is not adequate for low frequencies but its predictions improve as the frequency of the incident distortions increases.

An extension of Lighthill's acoustic analogy by Ffowcs Williams and Hawkings [9] through the use of generalized functions allowed researchers to take into account the effects of blade geometry such as thickness and planform. A method developed by Schulten [10, 11] computes both the field generated by the blade geometry and that resulting from interaction of the blades with an incident field. The solution is given in terms of Green's functions that incorporate such details as the presence of a hub and the acoustic treatment of the duct wall with liners. The method allows for the study of stagger angle, camber and vane sweep effects on the resulting noise.

All preceding theories can model both the sources and the propagation of sound in simple duct geometries, but are valid only for uniform subsonic flow.

When the blade row has supersonic tip speeds, the generated noise characteristics can change drastically. Recent studies [12] show that the response of an airfoil in realistic non-uniform transonic flow is considerably larger than that of a flat plate. As analytical solutions in this case are excluded, researchers are now considering numerical approaches that solve the governing equations for the stator and rotor simultaneously while taking into account their relative motion [13].

Radiation of specified acoustic modes was first predicted using analytical methods based on Wiener-Hopf techniques, both for two-dimensional [14] and axisymmetric [15] infinitely thin ducts. A numerical method using finite elements in the near acoustic field and wave envelope elements in the far field in order to reduce the number of grid points has been developed mainly by Eversman [16, 17]. The computed variable is the complex spatial acoustic pressure amplitude, or equivalently the acoustic velocity potential, and a weighted residual formulation allows an axisymmetric model to be used even for the higher order spinning modes. This is appropriate for most fan inlet geometries and helps reduce the needed computer resources. Since wave elements incorporate the proper decay of the spherical field with radius from the source, good results have been obtained for the amplitude, but not for the phase of the solution. More recently, Caruthers [18] employed free space Green functions to obtain very accurate discretizations, for both phase and amplitude, of the Helmholtz equation with minimal requirements for the number of points.

## 1.3 Computational Aeroacoustics approach

The methods presented above have a number of advantages. In particular, they allow easy parametric study in the design process, since the influence of a number of factors (row spacing, blade cross section, etc.) on the generated sound can be easily calculated. These advantages rely however on heavy assumptions such as the existence of a uniform flow field and the very small level of the perturbations (linearity). In the inlet and exhaust ducts of the fan the mean flow is in reality very complex and, in particular for high power settings, the level of the pressure perturbations is extremely high. In this case, the nonlinear effects become important, as energy transfers appear between different frequencies. These facts recently lead researchers [19, 20] to consider the nonlinear Euler/Navier-Stokes equations for modeling acoustic propagation in fan inlets.

The approach considered in this thesis as a first step towards the development of a system for fan noise prediction supposes the simultaneous use of CFD and CAA for a complete specification of the sources of noise and the computation of their sound field. The flow field through the rotor can be determined if a steady analysis in a rotating frame is performed. The results can be used to generate the disturbances that are associated with the rotor alone (in the form of a function of time and space) as seen on a plane a certain distance upstream of the rotor, hereafter called the source plane. A more complete CFD analysis of the unsteady interaction between the rotor and the stator can specify the disturbances corresponding to both the rotor-alone and rotor-stator interaction sources. For reasons of accuracy and cost, this analysis must be however limited to small regions surrounding the sources. Beyond the source plane, the use of the same CFD solver is still possible but would incur either large errors or else impractically large numbers of grid points. Therefore, the computation of the acoustic field propagated beyond the source plane is to be done using a CAA algorithm, as suggested also in [19, 20]. Such an algorithm is developed in this thesis.

As far as sound propagation is concerned, the molecular viscosity is unimportant unless very large propagation distances are involved. However, the refraction effects caused by the large gradients of the mean flow in the region of the boundary layers may be important and ultimately should not be neglected. Hence, the physical model must describe the possibly nonlinear propagation of acoustic waves superposed on the existing mean flow. To this end, the time-dependent Euler equations are used here. For convenience, the term 'Euler equations' is used henceforth to refer to the system of conservation laws for mass, momentum and energy in inviscid fluid flow.

For the flow in the fan inlet, in the case other solutions are not available or for large engines when the boundary layer flow occupies a relatively small region, the mean flow can be considered in a first approximation inviscid. In this case, the algorithm developed in the thesis allows the computation of both the mean flow and the sound field. First the mean flow is computed, as a steady solution of the equations subject to time-averaged boundary conditions. Afterwards, the boundary conditions are imposed as a function of time, reflecting the incidence of the acoustic perturbation on the mean flow, and the acoustic variables are obtained as the difference between the unsteady flow variables and the mean flow, once a periodic state has been obtained over the flow field. In the case of a fan exhaust, or for small engines, the mean flow must incorporate viscous effects. In this case, a minor modification of the algorithm allows calculating the nonlinear propagation of disturbances on a given mean flow provided by other means.

## 1.4 The need for spectral methods

When the mean flow and the acoustic perturbations are computed together as the solution of the same system of partial differential equations, a very important problem related to accuracy arises. It is due to the fact that the perturbations can be many orders of magnitude smaller than the mean flow, at least at some distance from the source. Minor errors in the computation can therefore affect the computed sound field to the point of rendering the solution meaningless. It makes sense therefore to use numerical methods that can offer a very high level of accuracy at a reasonable cost.

After the introduction of the dispersion-relation-preserving schemes by Tam and Webb [21] and the compact schemes with spectral-like resolution by Lele [22], finite difference methods started to be widely used for CAA. They offer a number of advantages, such as ease of analysis and programming and computational efficiency. Their use ranged from computing propagation of both linear [21] and nonlinear [23] acoustic waves to computation of sound radiation from fans [19] and modeling of jet screech noise [24]. The use of finite difference methods becomes however difficult when the grids have a multiblock structure, which is almost always the case for industrial applications. In this case, methods that offer a larger geometrical flexibility are a good alternative.

The initial effort of this research was focused on finite volume methods of very high order of accuracy. Structured mesh essentially non-oscillatory finite volume methods had been developed by Casper [25] and used for aeroacoustic computations [26]. Finite volume methods show clear advantages over finite difference methods in complex geometries. To fully exploit their potential, one should however be able to use unstructured meshes. A long time has therefore been spent on unstructured grid finite volume methods, of the essentially non-oscillatory type [27] and of central differencing type under a least-squares approach. Although very high accuracy was finally obtained (grid refinements on a plane wave propagation problem in 2D showed up to fourth order for the essentially non-oscillatory and sixth order for the least-squares methods, respectively), this accuracy proved still insufficient for acoustic modeling. In addition, the CPU time requirements of the methods were prohibitive.

It is well known from numerical analysis that approximations based on partial sums of Fourier (or Chebyshev, for the non periodic case) series have exponential convergence (or spectral accuracy) properties. That is, given an infinitely differentiable function $q$, the error between the approximate $q_N$ based on $N$ terms in the series and the real function is bounded as $\lim_{N\to\infty} N^n||q_N - q|| = 0$, for any positive value of $n$. From a practical point of view, this means that very high accuracy can be obtained with a small number of grid points. A plot of the error norm obtained from both spectral and finite difference discretizations (equivalent to the finite volume methods above for this case) of the unidimensional (1D) linear wave equation versus the total number of points is given in figure 1.1. It is obvious that for 3D computations of high frequency waves, when the number of grid points might reach several millions even for spectral methods, most finite volume/finite difference methods might need many more points and would henceforth be excluded. These results, although obtained at an arduous cost, definitely turned the course of this research, albeit late, towards spectral methods.

For the computation of propagating waves, an important requirement, as underlined by Tam [21], is that the numerical scheme must reproduce the dispersion relation of the governing partial differential equations. The quality of a given spa-

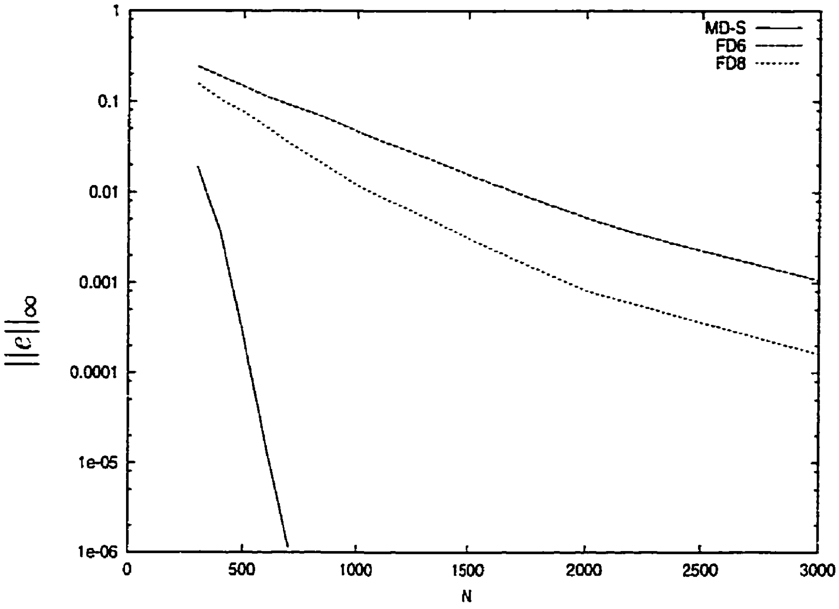


Figure 1.1: Norm of the error versus total number of points for the convection of a Gaussian pulse using the 1D linear wave equation and multidomain spectral (MD-S, 20 elements), 6th order (FD6) and 8th order (FD8) finite difference spatial discretizations. The same time integration method is used for all spatial discretizations.

tial discretization is assessed by studying it in wavenumber space. The numerical wavenumber generally differs very little from the exact one for small frequency waves but increasingly deviates from it as the frequency is larger than a certain limit. In other words, given a certain numerical discretization of the spatial terms in the governing equations, only the waves with the wavelength above a certain threshold can be represented properly. For classical fourth-order finite difference discretizations, for example, the wavelength must be larger than about $12\Delta x$, and larger than $6\Delta x$ for the optimized fourth-order methods devised in [21]. Spectral methods offer again the best alternative, with wavelengths as small as $2\Delta x$ (the

Nyquist frequency) being correctly represented by Fourier methods, and $\pi\Delta x$ by Chebyshev methods [28]. Here $\Delta x$ is the grid spacing, uniform for the finite difference and Fourier case and in an average sense for the Chebyshev case.

Spectral methods, yet, have a number of drawbacks. For nonlinear problems, the natural approach is to use the so-called 'pseudospectral' methods [29], wherein the nonlinear terms are evaluated in physical space. This, however, leads to aliasing errors in the computation of the convolution sums. Aliasing errors are usually removed by the use of techniques such as padding or phase shifts [29]. Another disadvantage when using spectral methods for the space terms is that implicit methods for the discretization of the time derivative are very difficult to construct. For explicit methods on the other hand, the time step, being proportional to the smallest grid spacing, can be severely limited due to the quadratic clustering of the grid points towards the end of the domain in the Chebyshev case. Stability problems as well as slow convergence in computing steady state solutions also characterize spectral methods.

The use of spectral methods has usually been restricted to simple geometries and, in the Fourier case, to periodic boundary conditions. This is due to the fact that a multidimensional spectral approximation is built using tensor products of 1D approximations in order to remain efficient [29]. Hence, the computational domain has to be either a cube or a domain that can be mapped on the unit cube by a coordinate transformation. To account for more complicated shapes, multidomain (patching) methods and spectral element methods have been developed only relatively recently. In this case, the computational domain is divided into smaller non-overlapping pieces, henceforth called elements, each of which has a stand-alone spectral approximation. In general, spectral element methods are considered to be those that use a variational or a weighted residual form of the equations over each element, as the Galerkin method that is also very much in use for finite element techniques, while multidomain methods are usually based on collocation. Continuity of the function and all its derivatives up to an order lower than those appearing in the governing equations must be enforced explicitly in multidomain methods.

The first successful spectral element method was introduced by Patera [30] for the incompressible Navier-Stokes equations. For compressible flow, Cai *et al.* [31] have shown, for a global spectral approximation, how to modify the spectral sums to account for the presence of discontinuities. Spectral element methods using the same principle have been developed by Sidilkover and Karniadakis [32]. The main difficulty that arises in this case is upwinding when several element corners meet together, the methods becoming unstable if it is not done properly. Elegant solutions have been found recently for this problem, for example the use of staggered grids proposed by Kopriva and Kolias [33] and the penalty method developed by Hesthaven [34].

Multidomain spectral methods have, however, distinct advantages over a global spectral approximation once this difficulty is solved. The first one is that a local approximation might be preferred to a global one due to the character of the governing equations, as is actually the case for the hyperbolic Euler system. The second one is that, the computational domain being already subdivided, an algorithm based on these methods can be implemented in a very natural way on parallel machines simply by assigning a number of elements to each processor. Also, the time step for explicit time marching methods can be much larger for a multidomain method using several elements with a smaller number of grid points inside each of them, than for a global discretization using only one domain with a very large number of points. A multidomain spectral method has therefore been considered appropriate in this work for the discretization of the spatial terms in the governing equations.

## 1.5 Thesis outline

This thesis describes an algorithm for computational aeroacoustics, mainly aimed at accurately describing noise propagation in fan inlets and its radiation in the near acoustic field. It also presents its application to a set of standard acoustic problems for the purpose of validation, as well as to several aeroacoustics problems of interest in gas turbine engineering.

The algorithm can be thought of as being made of several parts. First, it contains a highly accurate (spectral) discretization of the spatial terms in the partial differential equations. Second, for unsteady problems, it offers the possibility of using a discretization of the time derivative that is optimized for wave propagation. Finally, the algorithm is enhanced with the capability of treating acoustic propagation, which includes calculating the acoustic variables from the time-dependent flow variables as well as imposing proper boundary conditions for them. The computer code thus obtained can be used either as a steady state solver for the purpose of computing a mean flow or as an unsteady solver for computing acoustic propagation superposed on this mean flow.

The second chapter of this thesis shortly presents the Euler equations used as the mathematical expression of the physical laws governing sound propagation. The approximation of the spatial terms by a Chebyshev spectral sum is then described, using both a formalism based on Lagrange interpolants and the actual matrix-vector multiplication operations actually implemented in the code. The staggered grid initially proposed by Kopriva and Kolias [33] is used to transfer information between elements without involving element corners.

The third chapter focuses on the issues related to the time discretization. The properties of the Runge-Kutta methods for solving ordinary differential equations are discussed, and a way to construct methods optimized for wave propagation and using only a minimum amount of storage even for nonlinear problems is introduced. The accuracy of the new methods is tested on model problems.

The fourth chapter is devoted to the boundary conditions used for the computation of the mean flow as well as those for the computation of acoustic waves. Although at the source plane the approach allows specification of a general function of time and space for incoming waves, a single incoming duct mode is specified for the calculations considered in this thesis.

The fifth chapter discusses the relative advantages and the actual methodology for generating the needed 3D grids inside each element within the same computer code, from given geometrical information about its edges and faces. A technique for increasing the time step by mapping the grid points inside the elements is also described.

The sixth chapter presents numerical results obtained with the computer implementation of the algorithm. Comparisons of the obtained solutions with analytical ones are first performed for linear cases which form a subset of the governing equations. This is a necessary step for the validation of any nonlinear solver. Results are then given for several cases where analytical solutions are not possible but where the full capabilities of the algorithm are tested. Some of these cases are studies of acoustic radiation from a fan inlet currently under production at Pratt&Whitney Canada. The algorithm described herein is in further development under a joint effort from both the CFD Laboratory of Concordia University and the CFD Group at Pratt&Whitney Canada, the final goal being an industrial environment code for the study of nonlinear aeroacoustics of turbofans.

The seventh chapter shows how the method can be modified to account for mean flows that are not obtained as a solution of the Euler equations by the same solver and with the same spectral discretization, and applies it for simulating the turbulence mixing noise from a perfectly expanded supersonic jet. Conclusions and possible directions for future work are presented in the eighth chapter.

# Chapter 2

# Numerical model and space discretization

## 2.1 Physical model and governing equations

The sensation of sound is produced by pressure disturbances reaching the eardrum. Their propagation through the surrounding medium is governed by the Navier-Stokes equations, which fluid flow is considered to obey. For frequencies of most practical interest, however, molecular viscosity effects on sound propagation can be neglected, the pressure representing a much more important stress field than the viscous shear stresses. The ratio of the two stresses is given by the relevant Reynolds number $\mathcal{R} = \omega\lambda^2/\nu$, which is around $10^8$ for sound at most audible frequency in usual ambient conditions. For viscous effects to become important, sound must travel $\mathcal{R}$ wavelengths [35, 36]. Since regions of much smaller size are studied herein, viscosity is neglected in the Navier-Stokes equations. The resulting system of first order partial differential equations, denoted here as the Euler equations, describes the conservation of mass, momentum and energy in an inviscid fluid flow. It accounts for any refraction effects due to non-uniform mean flow on the propagation of sound, as well as possible nonlinear effects due to large source amplitude.

The Euler equations can be written under an unified notation, for both the Cartesian coordinates and the axisymmetric case, by the use of an integer switch that takes two discrete values. Considering the computational domain of interest $\mathcal{D} \subset \mathbb{R}^d$ with its boundary $\Gamma$, the vectorial form of the equations is:

$$\frac{\partial Q}{\partial t} + \sum_{i=1}^{d} \frac{\partial F_i}{\partial x_i} + vH\delta_{2d} = 0. \tag{2.1}$$

For Cartesian coordinates $v = 0$ and $d$ is the number of spatial dimensions. Setting $v = 1$ and $d = 2$ recovers the case of axisymmetric flows, which are actually three-dimensional flows but can be studied using a planar grid due to their symmetry. To ease the presentation, the position vector $\vec{x}$ is considered to have either the components $(x_1, x_2, x_3)$ or equivalently $(x, y, z)$ for the Cartesian coordinate system and $(x, r)$ for axial symmetry. The notation $\delta_{ij}$ denotes the Kronecker symbol while $Q$ is the vector of conserved state variables and $\vec{F}(F_1, F_2, F_3)$ and $H$ are the flux tensor and the source term vector, respectively. Their explicit form is:

$$Q = \begin{pmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho v_3 \\ \rho\epsilon \end{pmatrix}, \quad F_i = \begin{pmatrix} \rho v_i \\ \rho v_1 v_i + p\delta_{1i} \\ \rho v_2 v_i + p\delta_{2i} \\ \rho v_3 v_i + p\delta_{3i} \\ (\rho\epsilon + p)v_i \end{pmatrix}, \quad H = \frac{1}{x_2} \begin{pmatrix} \rho v_2 \\ \rho v_1 v_2 \\ \rho v_2^2 \\ (\rho\epsilon + p)v_2 \end{pmatrix}. \tag{2.2}$$

The primitive flow variables appearing in the above equation are the density $\rho$, the three components $v_i$ of the velocity vector $\vec{V}$ and the pressure $p$, while $\epsilon$ denotes the total energy per unit mass. Alternately, the velocity components will be denoted by $(u, v, w)$. The equation of state for an ideal gas

$$p = \rho(\gamma - 1)\left(\epsilon - \frac{\vec{V}^2}{2}\right) \tag{2.3}$$

where $\gamma$ is the ratio of the specific heats, is used to relate the total specific energy to the primitive variables and close the system. Both the flux vector components and the source term are functions of the state vector $Q$, i.e. $\vec{F} = \vec{F}(Q)$ and $H = H(Q)$.

The system of equations (2.1) has a hyperbolic character in time and any of the space variables [37]. An initial and boundary value problem (IBVP) is obtained when solutions to (2.1) are sought that satisfy initial conditions of the form

$$Q(\vec{x}, t_0) = Q_0(\vec{x}), \tag{2.4}$$

with $Q_0$ a given function, as well as appropriate boundary conditions written here in operator form

$$\mathcal{B}\left[Q(\vec{x}, t)\right] = 0 \quad \text{for } \vec{x} \in \Gamma. \tag{2.5}$$

## 2.2 Non-dimensional form of governing equations

In the Euler equations (2.1), the variables have their usual dimensions. This is not appropriate for a numerical algorithm, since the disparity in the magnitude of the physical variables can generate undesirable errors. Therefore the dependent variables in the equations as well as the time and space coordinates are set in a non-dimensional form denoted by an asterisk superscript as follows:

$$\begin{aligned} x_i{}^* = \frac{x_i}{L}, \quad t^* = \frac{tc_\infty}{L}, \quad v_i{}^* = \frac{v_i}{c_\infty}, \\ \rho^* = \frac{\rho}{\rho_\infty}, \quad p^* = \frac{p}{\rho_\infty c_\infty^2}, \quad \epsilon^* = \frac{\epsilon}{c_\infty^2}. \end{aligned} \tag{2.6}$$

where $L$ is a characteristic length properly chosen for each case, $c_\infty$ is the ambient speed of sound and $\rho_\infty$ is the ambient density.

Unlike the Navier-Stokes equations for which the non-dimensional form brings forth the Reynolds number, the non-dimensional Euler equations keep the same form given by (2.1). Therefore, the asterisk superscripts will be henceforth omitted, with the understanding that all the variables are nondimensional.

## 2.3 Curvilinear coordinates

In view of a multidomain approach, the computational domain $\mathcal{D}$ is divided into $E$ non-overlapping simply connected domains (called elements for convenience) $D_e, e = 1, \ldots, E$ such that $\mathcal{D} = \cup_{e=1}^{E} D_e$. A representative element denoted by $D$ is a general quadrilateral for $d = 2$ and a general hexahedron for $d = 3$. The IBVP is solved on each element separately, with continuity of the state vector $Q$ (patching) enforced explicitly at element boundaries as part of the solution process.

Chebyshev polynomial approximations of functions are defined naturally on the interval $[-1, 1]$, with efficient extension to multiple dimensions through the use of tensor products on what will be called the 'master element', $\mathbb{D} = [-1, 1]^d$.

On the generic element $D$ the approximation can be obtained by the use of a one-to-one transformation $\vec{\xi}(\vec{x})$ with $\xi_i \in [-1, 1], i = 1, \ldots, d$ which maps $D$ onto $\mathbb{D}$, and of its inverse $\vec{x}(\vec{\xi})$. A way to obtain such a mapping is described in chapter 5. The correspondence between $\mathbb{D}$ and $D$ is shown schematically in figure 2.1, from which it can be seen that constant $\xi_i$ planes become curvilinear surfaces in physical space. As for $\vec{x}$, the components of $\xi$ will be considered to be $(\xi_1, \xi_2, \xi_3)$ or $(\xi, \eta, \zeta)$ interchangeably, while $\xi$ will be used generically for any of the components.

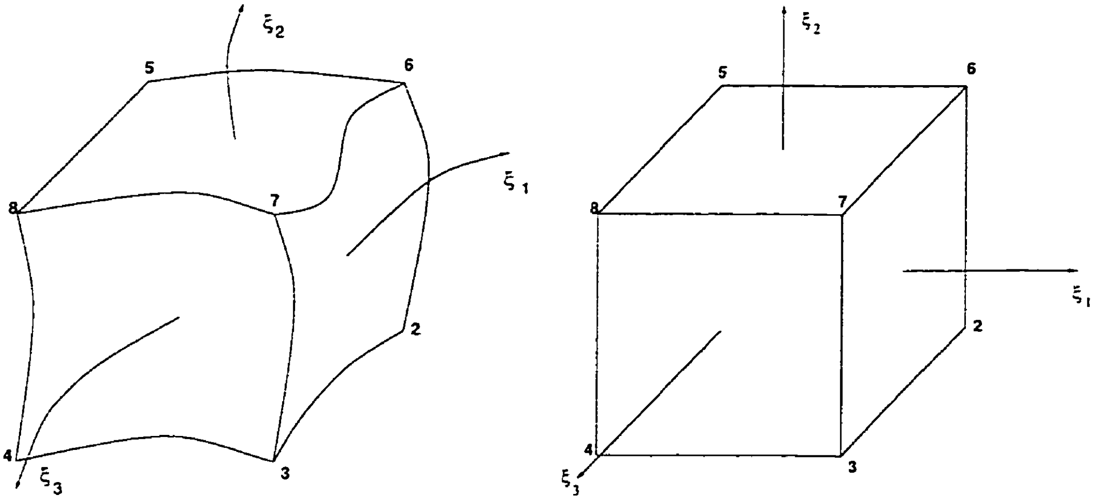


Figure 2.1: Mapping of a generic element $D$ onto the master element $\mathbb{D}$.

Using the component form of the transformation, $\xi = \xi(x, y, z), \eta = \eta(x, y, z)$, $\zeta = \zeta(x, y, z)$, infinitesimally small displacements in the two coordinate systems can be related by:

$$\begin{bmatrix} d\xi \\ d\eta \\ d\zeta \end{bmatrix} = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} . \tag{2.7}$$

Denoting by $\mathcal{J}$ the first Jacobian of the transformation,

$$\mathcal{J} = \det \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} = \xi_x\eta_y\zeta_z + \eta_x\zeta_y\xi_z + \zeta_x\xi_y\eta_z - \xi_z\eta_y\zeta_x - \eta_z\zeta_y\xi_x - \zeta_z\xi_y\eta_x \tag{2.8}$$

and using the chain rule of differentiation, the governing equations on the master element can be shown [38] to become:

$$\frac{\partial \tilde{Q}}{\partial t} + \sum_{i=1}^{d} \frac{\partial \tilde{F}_i}{\partial \xi_i} + vH\delta_{2d} = 0. \tag{2.9}$$

where the transformed components of the state and flux vectors are obtained using the Jacobian and the components of the metric tensor:

$$\tilde{Q} = \frac{Q}{\mathcal{J}}, \quad \tilde{F}_i = \frac{1}{\mathcal{J}} \sum_{k=1}^{d} \frac{\partial \xi_i}{\partial x_k} F_k \tag{2.10}$$

The spatial derivatives that appear in equation (2.9) can now be conveniently calculated within the master element.

## 2.4 Staggered grid approximation

It is advantageous to enforce the continuity of $Q$ required by the multidomain approximation as part of the evaluation of the derivatives by the use of a set of staggered grids first introduced by Kopriva and Kolias [33]. One of the advantages of the staggered grid approximation is that it is fully conservative, as demonstrated in [33], hence shock capturing techniques can be used. The most important advantage however resides in the fact that element corners are not included in the approximation, such that no matter how many elements meet together at a corner, continuity of solution needs only be enforced between any two neighboring elements at their common points along a face. This greatly simplifies the upwinding process which is necessary for the stability of the method. Standard upwinding techniques such as flux-vector or flux-difference splitting [37] can be used to this end.

The staggered grids are constructed using two discrete sets of points over which the Chebyshev polynomials are orthogonal, the Gauss-Chebyshev points $\mathbb{G}$ and the Gauss-Chebyshev-Lobatto points $\mathbb{L}$. Denoting by $N$ the maximum polynomial degree present in the approximation, they are defined as follows:

$$\mathbb{G} = \left\{ \bar{\xi}^j | \bar{\xi}^j = -\cos\left(\frac{(2j-1)\pi}{2N}\right), j = 1, \ldots, N \right\}, \tag{2.11}$$

$$\mathbb{L} = \left\{ \xi^j | \xi^j = -\cos\left(\frac{j\pi}{N}\right), j = 0, \ldots, N \right\} \tag{2.12}$$

Notice that only the set $\mathbb{L}$ includes the endpoints of the interval, i.e. $\xi^0 = -1$, $\xi^N = 1$.

The state vector $\tilde{Q}$ and the source vector $H$, if present, are defined only at points that belong to $\mathbb{G}^d = \mathbb{G} \times \mathbb{G} \times \mathbb{G}^{d-2}$, the tensor product of the one-dimensional sets of Gauss-Chebyshev points. It is at the same points that the partial differential equations (2.9) are satisfied in a collocation sense since values of the spatial derivatives of the fluxes are also computed there. The components of the flux are defined however on different grids obtained by replacing the Gauss-Chebyshev points with the Gauss-Chebyshev-Lobatto points for the corresponding flux direction in the above tensor product, i.e. $\tilde{F}_1$ is collocated at $\mathbb{L} \times \mathbb{G}^{d-1}$, $\tilde{F}_2$ at $\mathbb{G} \times \mathbb{L} \times \mathbb{G}^{d-2}$, and $\tilde{F}_3$ at $\mathbb{G}^{d-1} \times \mathbb{L}$. Figure 2.2(a) shows the distribution of the flux collocation points on the surface of the master element for the case $N = 3$, and figure 2.2(b) shows a cut in a $\xi_3$=cst plane that goes through the state vector definition points.

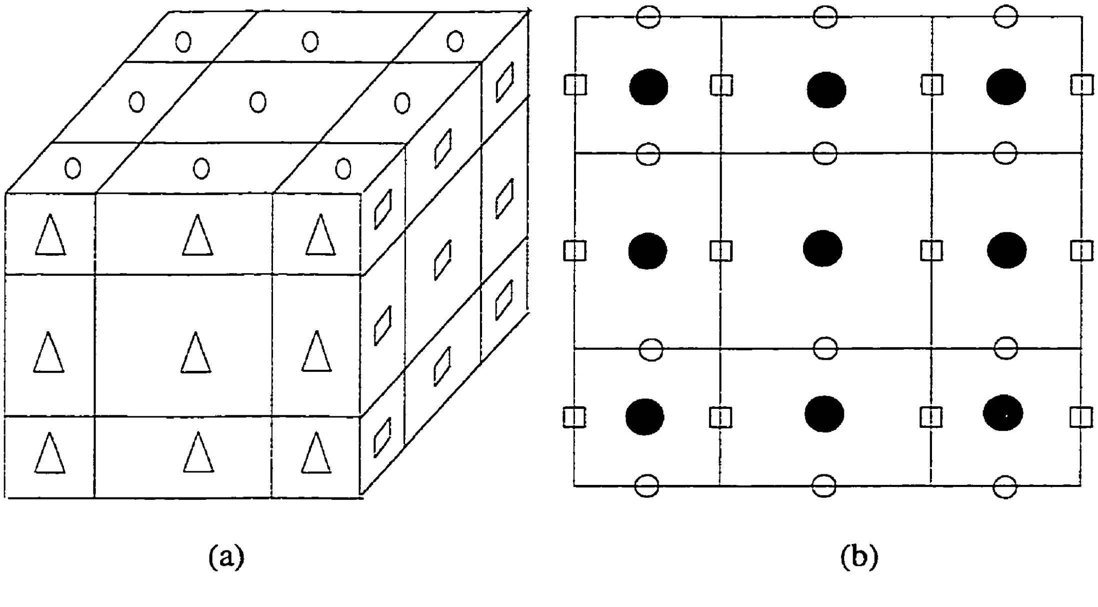


Figure 2.2: Distribution of flux and state vector collocation points (□ : $\tilde{F}_1$, ∘ : $\tilde{F}_2$, △ : $\tilde{F}_3$, • : $\tilde{Q}$). (a) Surface of the master element (b) $\xi_3$=cst plane.

## 2.5 Spatial derivatives and patching

At the beginning of a time step, the initial data are the state vector values $\tilde{Q}$ at the points $\mathbb{G}^d$. From these values one can create a multi-dimensional interpolant of the form:

$$\tilde{Q}(\xi_1,\xi_2,\xi_3) = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\tilde{Q}(\bar{\xi}_1^i,\bar{\xi}_2^j,\bar{\xi}_3^k)\bar{h}_i(\xi_1)\bar{h}_j(\xi_2)\bar{h}_k(\xi_3) \tag{2.13}$$

where the interpolating Lagrange polynomials $\bar{h}(\xi)$ corresponding to the set of Gauss-Chebyshev points $\mathbb{G}$,

$$\bar{h}_i(\xi) = \prod_{l=1,l\neq i}^{N}\left(\frac{\xi-\bar{\xi}^l}{\bar{\xi}^i-\bar{\xi}^l}\right) \tag{2.14}$$

are introduced. From this interpolant the values of the state vector at the flux definition points can be readily obtained. For example, at the points where the $\tilde{F}_1$ flux is needed, $\tilde{Q}$ can be computed as:

$$\tilde{Q}(\xi_1^n,\bar{\xi}_2^j,\bar{\xi}_3^k) = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\tilde{Q}(\bar{\xi}_1^i,\bar{\xi}_2^j,\bar{\xi}_3^k)\bar{h}_i(\xi_1^n)\bar{h}_j(\bar{\xi}_2^j)\bar{h}_k(\bar{\xi}_3^k), \quad n=0,\ldots,N. \tag{2.15}$$

Due to the properties of the Lagrange interpolants that $\bar{h}_i(\bar{\xi}^j) = \delta_{ij}$, this becomes in fact a one-dimensional operation,

$$\tilde{Q}(\xi_1^n,\bar{\xi}_2^j,\bar{\xi}_3^k) = \sum_{i=1}^{N}\tilde{Q}(\bar{\xi}_1^i,\bar{\xi}_2^j,\bar{\xi}_3^k)\bar{h}_i(\xi_1^n), \quad n=0,\ldots,N. \tag{2.16}$$

along a $\xi_1$-line defined by $\xi_2 = \bar{\xi}_2^j$, $\xi_3 = \bar{\xi}_3^k$. A sketch of this interpolation, within a $\xi_3$=constant plane say, is presented in figure 2.3. To define the solution at all the flux points, the interpolation is performed along all $\xi_1$-, $\xi_2$-, and $\xi_3$-lines within the master element.

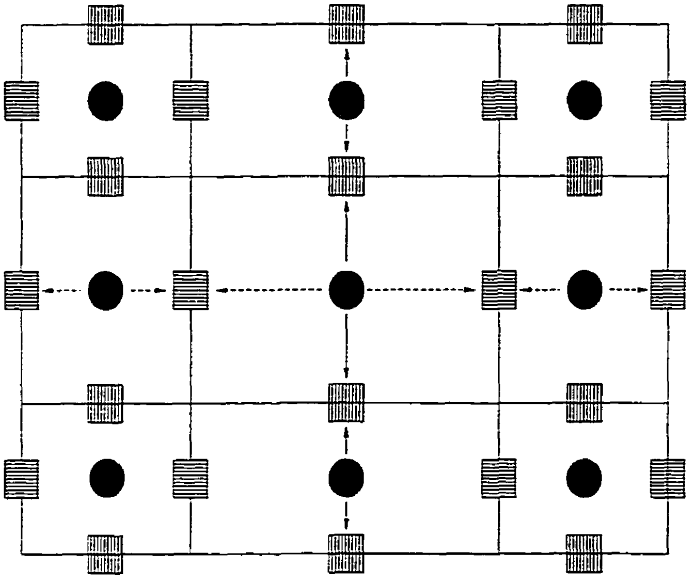


Figure 2.3: Interpolation: solution at $\mathbb{G}^d$ points is interpolated to Lobatto points. Only two lines are explicitly shown.

Upon performing the interpolation, $\tilde{Q}$ is known at all the flux definition points. As can be noticed from figure 2.3, element corners do not have flux definition points, but at element interfaces there will be two values for the state vector, from the two neighboring elements that share the face. These values are not necessarily equal. Patching together of the elements, or correspondingly imposition of the physical boundary conditions of the problem when element boundaries coincide with $\Gamma$, is done at this stage. A continuous state vector across element boundaries is created by the solution of a Riemann problem projected on the normals to the interface at each interface point. The solution is assigned to both neighboring elements. The process is represented in figure 2.4 and discussed in more detail in section 2.7. Considering the initial interpolated solution along a $\xi$-line to be $(\tilde{Q}^0, \tilde{Q}^1, \ldots, \tilde{Q}^N)$, corresponding to the points $(\xi^0, \xi^1, \ldots, \xi^N)$, the solution of the Riemann problem (or the physical boundary conditions) leads to different values for $\tilde{Q}$ at the end points, such that the line values now become $(\tilde{Q'}^0, \tilde{Q}^1, \ldots, \tilde{Q}^{N-1}, \tilde{Q'}^N)$.

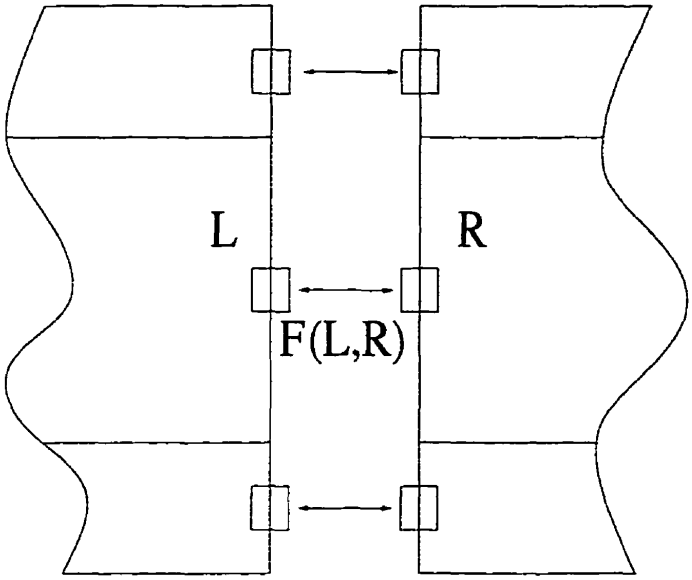


Figure 2.4: Patching: a continuous solution is created by solving a Riemann problem between the left and right states interpolated from elements on each side of an interface and assigning it to both of them.

From the solution $\tilde{Q}$ at the flux definition points, the fluxes along a $\xi$-line $(\tilde{F}^0, \tilde{F}^1, \ldots, \tilde{F}^N)$ can be obtained through the functional relation $\vec{F} = \vec{F}(Q)$. These values can be used to compute the derivatives at the initial set of points $\mathbb{G}^d$. Indeed, a multidimensional interpolant for $\tilde{F}_1$ for example can be constructed in a similar way as for $\tilde{Q}$

$$\tilde{F}_1(\xi_1, \xi_2, \xi_3) = \sum_{n=0}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} \tilde{F}_1(\xi_1^n, \bar{\xi}_2^j, \bar{\xi}_3^k) h_n(\xi_1) \bar{h}_j(\xi_2) \bar{h}_k(\xi_3) \tag{2.17}$$

where $h(\xi)$ is the Lagrange interpolant based on the Lobatto set of points, $\mathbb{L}$:

$$h_n(\xi) = \prod_{l=0, l \neq n}^{N} \left( \frac{\xi - \xi^l}{\xi^n - \xi^l} \right). \tag{2.18}$$

The needed differential is obtained by differentiating the Lagrange interpolating polynomials:

$$\frac{\partial \tilde{F}_1}{\partial \xi_1}(\bar{\xi}_1^i, \bar{\xi}_2^j, \bar{\xi}_3^k) = \sum_{n=0}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} \tilde{F}_1(\xi_1^n, \bar{\xi}_2^j, \bar{\xi}_3^k) \frac{\partial h_n}{\partial \xi_1}(\bar{\xi}_1^i) \bar{h}_j(\bar{\xi}_2^j) \bar{h}_k(\bar{\xi}_3^k), \quad i = 1, \ldots, N. \tag{2.19}$$

which can be seen to be again a one-dimensional operation along a ($\xi_2 = \bar{\xi}_2^j$, $\xi_3 = \bar{\xi}_3^k$) line:

$$\frac{\partial \tilde{F}_1}{\partial \xi_1}(\bar{\xi}_1^i, \bar{\xi}_2^j, \bar{\xi}_3^k) = \sum_{n=0}^{N} \tilde{F}_1(\xi_1^n, \bar{\xi}_2^j, \bar{\xi}_3^k) \frac{\partial h_n}{\partial \xi_1}(\bar{\xi}_1^i), \quad i = 1, \ldots, N. \tag{2.20}$$

The sketch in figure 2.5 represents the differentiation operation. The fact that both interpolation and differentiation are performed only along one $\xi$-line at a time makes the method at least as efficient as other spectral discretizations [32, 34] for $d \geq 2$.

To summarize, spatial derivatives are computed in three main steps:

**Interpolation** The solution vector $\tilde{Q}$ is interpolated to flux vector definition points.

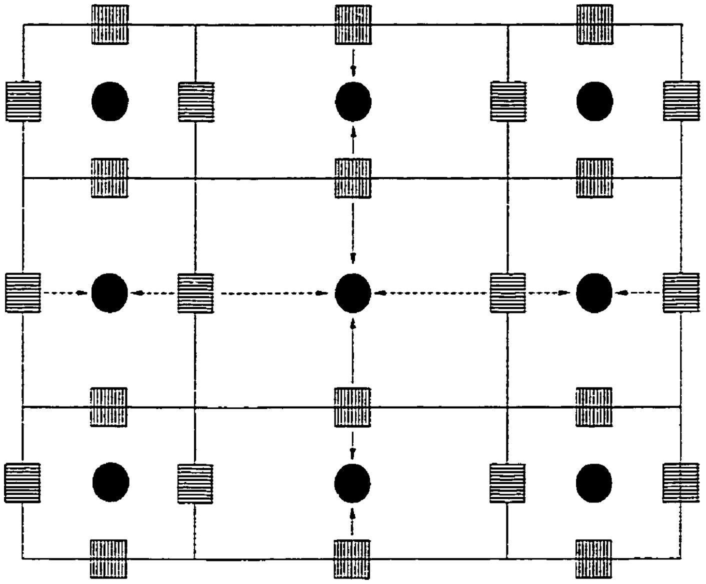


Figure 2.5: Differentiation: values of the derivatives at the set of points $\mathbb{G}$ are created from flux values at the Lobatto points.

**Patching** Continuity of the solution between elements, as well as physical boundary conditions on $\Gamma$, are enforced on the interpolated $\tilde{Q}$ values.

**Differentiation** Flux values are computed at the flux vector definition points through the use of the functional relation $\vec{F} = \vec{F}(Q)$, and differentiated to obtain the $\xi$-derivatives at the collocation points $\mathbb{G}^d$.

## 2.6 Matrix multiplication implementation

Due to the particular choice of the grid points $\mathbb{G}$ and $\mathbb{L}$, the Lagrange interpolants and their derivatives can be easily expressed in terms of Chebyshev polynomials. Using, for example, the discrete orthogonality relation between these polynomials over the set of points $\mathbb{G}$, the interpolant $\bar{h}_i(\xi)$ can be found to be:

$$\bar{h}_i(\xi) = \frac{2}{N}\sum_{l=0}^{N-1} \frac{1}{C_l}\, T_l(\bar{\xi}^i)\, T_l(\xi), \quad i = 1, \ldots, N \tag{2.21}$$

where $C_l = 2$ for $l = 0, N$ and $C_l = 1$ otherwise. On a generic $\xi$-line such as $\xi_1$ in equation (2.16), omitting for ease of presentation the two coordinates that are constant, the interpolation

$$\tilde{Q}(\xi^n) = \sum_{i=1}^{N} \tilde{Q}(\bar{\xi}^i)\bar{h}_i(\xi^n), \quad n = 0, \ldots, N \tag{2.22}$$

can then be written in matrix form:

$$\mathbf{Q}_{\mathbb{L}} = \mathbf{I}\mathbf{Q}_{\mathbb{G}} \tag{2.23}$$

where $\mathbf{Q}_{\mathbb{G}} = \left[\tilde{Q}(\bar{\xi}^1), \ldots, \tilde{Q}(\bar{\xi}^N)\right]^T$ is the vector of known solution values at the Gauss-Chebyshev points, $\mathbf{Q}_{\mathbb{L}} = \left[\tilde{Q}(\bar{\xi}^0), \ldots, \tilde{Q}(\bar{\xi}^N)\right]^T$ is the solution interpolated to the Lobatto points, and $\mathbf{I}$ is a $(N+1) \times N$ matrix, the elements of which can be found to be:

$$\begin{aligned}
\mathbf{I}_{ni} \;&=\; \bar{h}_i(\xi^n) = \frac{2}{N}\sum_{l=0}^{N-1}\frac{1}{C_l}\, T_l(\bar{\xi}^i)\, T_l(\xi^n) && (2.24)\\
&= \frac{2}{N}\sum_{l=0}^{N-1}\frac{1}{C_l}\cos\left[l\cos^{-1}\left(-\cos\left(\frac{(2i-1)\pi}{2N}\right)\right)\right]\cos\left[l\cos^{-1}\left(-\cos\left(\frac{n\pi}{N}\right)\right)\right]\\
&= \frac{2}{N}\sum_{l=0}^{N-1}\frac{1}{C_l}\cos\left(l\pi\left(1-\frac{2i-1}{2N}\right)\right)\cos\left(l\pi\left(1-\frac{n}{N}\right)\right).
\end{aligned}$$

Differentiation of the flux values on a $\xi$-line can also be put in a matrix-vector multiplication form using either the direct differential of the Lagrange interpolants or in the same way as above by expressing them through the use of Chebyshev polynomials. In the first alternative, for example, writing the differentiated flux interpolant equation (2.20) on a generic $\xi$-line on which the other two components of $\xi$ are constant as

$$\frac{\partial \tilde{F}}{\partial \xi}(\bar{\xi}^i) = \sum_{n=0}^{N} \tilde{F}(\xi^n) \frac{\partial h_n}{\partial \xi}(\bar{\xi}^i), \quad i = 1, \ldots, N. \tag{2.25}$$

the corresponding matrix form can be found to be

$$\mathbf{F}'_{\mathbb{G}} = \mathbf{D}\mathbf{F}_{\mathbb{L}} \tag{2.26}$$

where, taking into consideration equation (2.18), the elements of the $N \times (N+1)$ differentiation matrix are:

$$\mathbf{D}_{in} = \left.\frac{\partial h_n}{\partial \xi}\right|_{\bar{\xi}^i} = \frac{1}{\prod\limits_{l=0,l\neq n}^{N} (\xi^n - \xi^l)} \sum_{k=0,k\neq n}^{N} \left( \prod_{l=0,l\neq k,n}^{N} (\bar{\xi}^i - \xi^l) \right). \tag{2.27}$$

A stable and efficient way to compute differentiation matrices based on the Lagrange interpolants as above was developed by Fornberg [28].

## 2.7 The Riemann problem

The interaction of the two states interpolated from the neighboring elements at an interface can be thought of as a Riemann problem if the time step is sufficiently small such that wave propagation effects in a non-uniform flow can be neglected.

A 1D Riemann problem is the initial value problem defined by the hyperbolic system of equations

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial \xi} = 0 \tag{2.28}$$

subject to the particular initial conditions

$$Q(\xi, t = 0) = \begin{cases} Q_L, \xi < 0 \\ Q_R, \xi \geq 0 \end{cases} \tag{2.29}$$

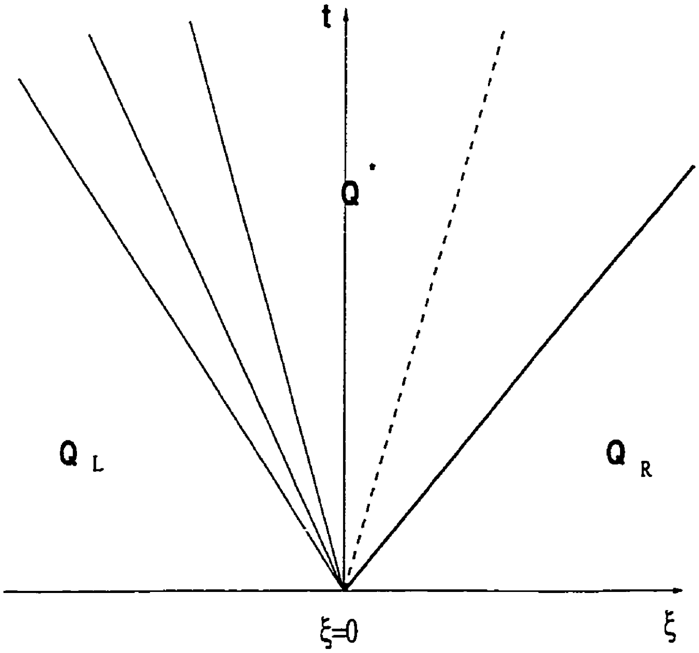


Figure 2.6: Riemann problem solution. The state $Q^*$ at the interface $\xi = 0$ is computed from given values of $Q$ left ($Q_L$) and right ($Q_R$) of it. The shock wave (thick line), the contact discontinuity (dashed line) and the rarefaction fan are also shown.

In the case of the 1D Euler equations, the Riemann problem (2.28) describes the flow in a shock tube (tildes on $Q$ and $F$ have been omitted). The resulting flow is self-similar, the flow variables being constant along any $\xi/t$ =constant line going through $\xi = 0$ in the $(\xi, t)$ plane. To obtain the resulting flow that constitutes the solution to the Riemann problem, an iterative process is necessary due to the nonlinear nature of the equations. Such a solution has first been presented by Godunov [39]. The solution consists of three waves, a shock wave, a contact discontinuity and a rarefaction fan, moving from the initial discontinuity at $\xi = 0$ with different speeds as represented in figure 2.6. Given the initial data $\tilde{Q}_L$ and $\tilde{Q}_R$ obtained by interpolation, the state $Q^*$ at the interface $\xi = 0$ can be obtained for any $t \geq 0$. This is the common state that is assigned to both neighboring elements at the interface. Approximate solvers which keep desired properties of the solutions have been introduced first by Engquist and Osher [40, 41] and Roe [42], and numerous others have been developed afterwards. An algorithm based on Godunov's method was developed in the present author's master thesis [43] and has been used for some of the computations presented herein. The remaining computations were done using an implementation of Roe's approximate solver [42] suggested by LeVeque [44].

For multidimensional flows, the Riemann problem is complicated by the fact that there can now be an infinite number of possible directions for the resulting waves. Although some work has been done towards the solution of multidimensional Riemann problems [45, 46], such techniques are still in their development stage and are not feasible enough to be used with spectral discretizations. To provide proper upwinding at interfaces, an ad-hoc solution consists then to solve a 1D Riemann problem in a direction that is normal to the interface. Consider for example an interface that corresponds to a $\xi_1 = \xi_1^N$ surface for one of the neighboring ('left') elements and a $\xi_1 = \xi_1^0$ for the other ('right') one. Let the interpolated primitive variables be $(\rho_L, \vec{V}_L, p_L)$ and $(\rho_R, \vec{V}_R, p_R)$, respectively, and let the normal to the surface, $\vec{n} = \nabla\xi_1$, be defined such that it points out of the left element and into the right element. The initial conditions for the 1D Riemann problem in equation (2.29) are then considered to be $(\rho_L, \vec{V}_L \cdot \vec{n}, p_L)$ and $(\rho_R, \vec{V}_R \cdot \vec{n}, p_R)$. The tangential velocity components are defined using the solution for the contact discontinuity, such that the values at the interface are the same with the values of the left state if the speed of the contact discontinuity is positive or equal to the right values in the contrary case. This completely determines the interface state, from which the flux values can be computed.

This way of solving the interface problem, while being the basic dissipative mechanism of the method, necessarily introduces an error in the approximation. As has been however shown by Powell *et al.* [47], this error decreases when the interface states are accurately computed, as can be expected with the spectral approximation.

## 2.8 Parallel processing

For the algorithm to take advantage of parallel computers, a convenient data structure has to be defined. It can be seen that both interpolation and differentiation can be performed element by element, since no interaction between elements is necessary. The involved variables are the state vector values at the $\mathbb{G}^d$ points and the flux values at the $d$ sets of flux definition points, hence for easy parallel processing they should be stored in a data structure with its outmost index the element number. On the other hand, the resolution of the Riemann problem is done along faces, such that the interpolated interface values should also be written in a data structure with the last varying index indicating the face.

Since for generality the element mesh can be unstructured and is given only as an array of nodal coordinates and a table of connectivity for the nodes of each element, a proper connection between the element-indexed structure and the face-indexed structure must be built within the algorithm in order to correctly transfer the values between them. For ease of presentation such a connection is considered to be known in the form of a mapping between the face-indexed structure and the element-indexed one, as represented in figure 2.7 for $d = 2$. The parallel algorithm for one time step can then be summarized as follows:

```
for e=1 to number_of_elements do_parallel
    interpolation(Q,e)
endfor;
for f=1 to number_of_faces do_parallel
    if( interface) then
       (L,R,f) := inverse_map(Q,e) ;
       Riemann_solve (L,R,f);
       (Q,e) := map(L,R,f)
    else
       impose_boundary_conditions(f)
    endif
endfor;
for e=1 to number_of_elements do_parallel
    compute_fluxes(Q,e);
    differentiation(F,e)
endfor;
```

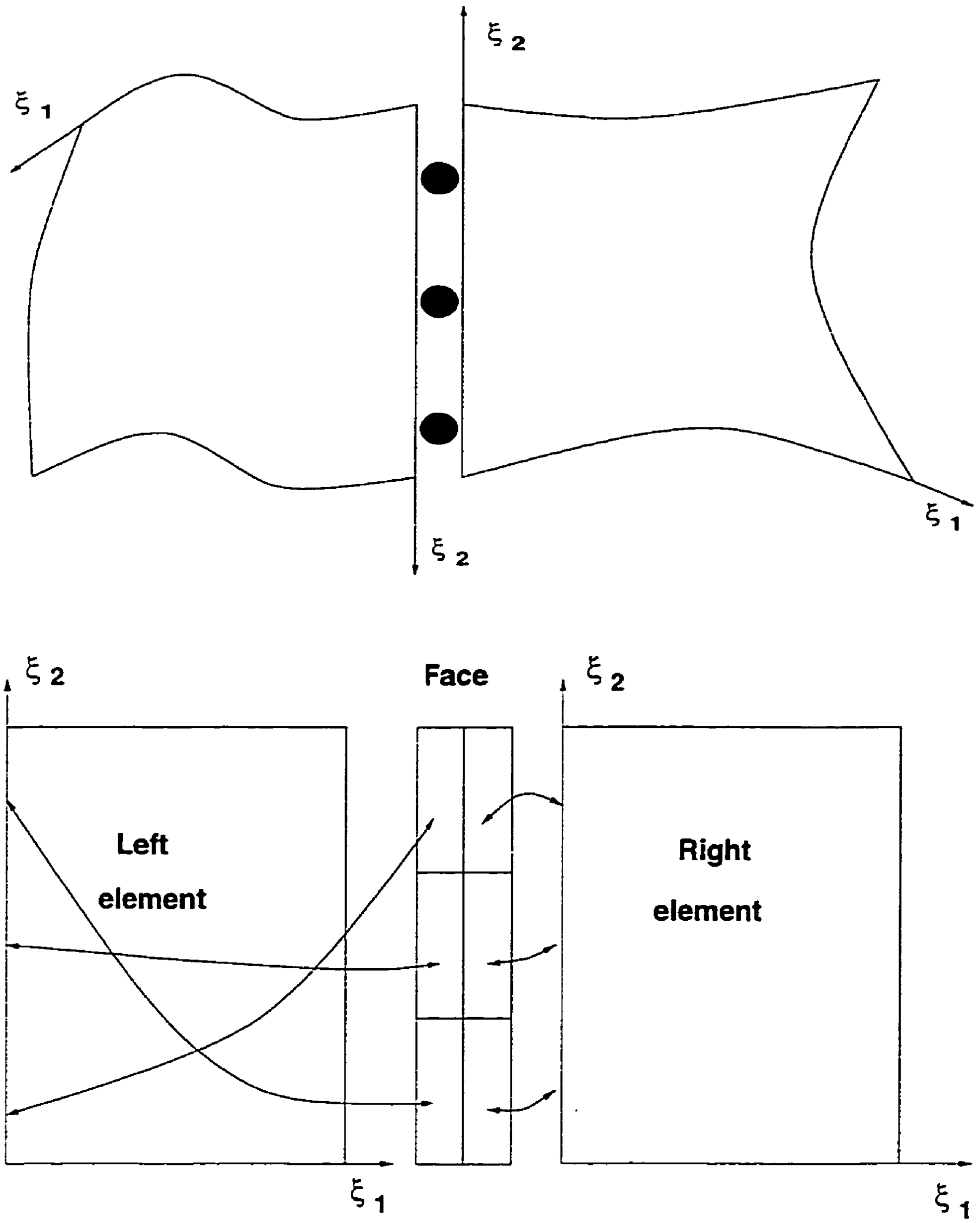


Figure 2.7: Mapping from element to face data structure. Upper side of the figure shows the neighboring elements in physical space. Bottom part shows correspondence of indices between the two data structures.

# Chapter 3

# Time discretization

## 3.1 Introduction

For physical problems that require accurate time-dependent numerical wave propagation in their simulation, as is the case in acoustics, the usual requirement of a low truncation error is not enough to guarantee that a numerical method yields accurate results. Indeed, as has been pointed out mainly in [21], the dissipation and dispersion properties of the numerical method are very important for computing wave solutions of systems of partial differential equations. This is valid for both the spatial and the time discretization methods. The explicit Runge-Kutta (RK) methods are widely used to discretize the time derivative because of their advantages that include flexibility, large stability limits and ease of programming. The properties of the RK methods have been studied extensively, in particular by Butcher [48]. He found the general form of the relations that the coefficients have to obey such that a given RK method has a certain order of accuracy, usually called the 'order conditions'. He further demonstrated that more stages than the order of accuracy are needed for fifth or higher order RK methods. Recently, Hu and coworkers [49] showed that the dissipation and dispersion properties of the RK methods depend on their coefficients and optimized them for the convective wave equation, obtaining so-called low dissipation and dispersion Runge-Kutta (LDDRK) methods. These methods are more efficient for wave propagation problems than classical ones, when the work required for a given accuracy is considered.

Solutions of three-dimensional aeroacoustics problems may easily need millions of grid points, in particular when high frequencies are involved. For such large size problems, in-core memory requirements may become exhaustive. They can be decreased using special RK methods that can be written such that only $2\mathcal{N}$-storage is required, where $\mathcal{N}$ is the number of degrees of freedom of the system (i.e. number of grid points × number of variables). To design such RK methods, enough free coefficients must exist such that additional constraints hold between them, together with the order conditions. Williamson [50] first showed that all second-order and some third-order methods can be written in $2\mathcal{N}$-storage form. He also showed that the fourth-order four-stage methods cannot be written in this way. By allowing additional stages and using the resulting new free coefficients to impose the $2\mathcal{N}$-storage constraints, Carpenter and Kennedy [51] devised a fourth order, five-stages RK method that is competitive with the classical fourth order method which however requires at least $3\mathcal{N}$ storage.

Hu *et al.* [49] provide $3\mathcal{N}$-storage implementations of the LDDRK schemes. These are valid for linear problems only, in the sense that they turn to second order accuracy when applied to nonlinear problems. A fourth order $2\mathcal{N}$-storage method has also been developed by Zingg and Chisholm [52], under the same linearity restriction. Since most LDDRK schemes have a number of stages that exceeds their order of accuracy, they can however be written in 2N-storage format even for nonlinear problems. The following sections present the necessary theory and develop the 2N-storage conditions that the coefficients of the RK method have to obey in such a case. The explicit imposition of the order conditions, the 2N-storage constraints, and the low-dissipation/dispersion conditions lead to nonlinear systems of equations that are solved numerically. Such systems generally have multiple solutions. For each case a full set of coefficients is given, chosen among the multiple solutions available. The chapter ends with numerical results that show that the methods developed herein keep their order of accuracy for nonlinear problems as well, and have clear advantages over other methods for wave-propagation problems.

## 3.2 General theory

The general case of a non-autonomous system of ordinary differential equations of the form:

$$\frac{dQ}{dt} = \mathbf{F}(t, Q(t)); \; Q(t_0) = Q_0 \tag{3.1}$$

is considered here. Such a system can be seen to arise from the Euler equations (2.1) upon the discretization of the space derivatives, which will form the right-hand side of equation (3.1). The function $\mathbf{F}$ is supposed to be a general, nonlinear function of its arguments. These arguments also contain the time because in the general case the boundary conditions (2.5) can depend on time.

The general form of an explicit, $o$-th order of accuracy $s$-stage RK method for computing the numerical approximation $q^n$ to $Q(t^n = t^{n-1} + \Delta t)$ is:

$$\begin{aligned} q^n &= q^{n-1} + \Delta t \sum_{i=1}^{s} b_i K_i \\ K_i &= \mathbf{F}\left(t^{n-1} + \Delta t c_i,\ q^{n-1} + \Delta t \sum_{j=1}^{i-1} a_{ij} K_j\right), \end{aligned} \tag{3.2}$$

where $c_i = \sum_{j=1}^{i-1} a_{ij},\ i = 1 \ldots s.$

For the method in equation (3.2) to have the required order of accuracy, the coefficients must obey certain order conditions [48]. These are obtained by equating coefficients of the Taylor series developments of $Q$. The explicit form of these conditions, up to fourth order of accuracy, is:

$$\begin{array}{ll} (o=1)\ \sum b_i = 1\ ; & (o=4)\ \sum b_i c_i^3 = \frac{1}{4} \\ (o=2)\ \sum b_i c_i = \frac{1}{2}\ ; & (o=4)\ \sum b_i c_i a_{ij} c_j = \frac{1}{8} \\ (o=3)\ \sum b_i c_i^2 = \frac{1}{3}\ ; & (o=4)\ \sum b_i a_{ij} c_j^2 = \frac{1}{12} \\ (o=3)\ \sum b_i a_{ij} c_j = \frac{1}{6}\ ; & (o=4)\ \sum b_i a_{ij} a_{jk} c_k = \frac{1}{24} \end{array} \tag{3.3}$$

In these relations, sums on all indices extend from 1 to $s$. On the left side of each condition the order of accuracy that the condition governs is indicated between parentheses, with the understanding that for a certain order all the conditions up to that order must be obeyed. For example, the coefficients of a second order method must obey only the two conditions marked with $(o = 1)$ and $(o = 2)$, while those for a fourth order method must obey all the eight conditions in equation (3.3).

To obtain low-storage methods, the principle is to leave useful information in the storage locations, by writing each successive stage on the same register without zeroing the previously held values. Instead of computing the $s$ coefficients $K_i$, the algorithm then becomes (with $w/\Delta t$ an approximation to $dQ/dt$):

$$\left.\begin{aligned} w_i &= \alpha_i w_{i-1} + \Delta t \mathbf{F}(t_{i-1}, q_{i-1}) \\ q_i &= q_{i-1} + \beta_i w_i \end{aligned}\right\}, \quad i = 1, \ldots, s \tag{3.4}$$

with $\alpha_1 = 0$ for the algorithm to be self-starting. Here $q_0 = q^{n-1}$, $q^n = q_s$ and $t_i = t^{n-1} + c_i \Delta t$. Since only the $w$ and $q$ values must be stored for each degree of freedom, this results in a $2\mathcal{N}$-storage algorithm.

Using equations (3.2) and (3.4), one can express the usual RK coefficients $a_{ij}, b_i$ in terms of $\alpha_i, \beta_i$. Since these relations depend on the number of stages $s$, they are presented in detail in the next sections. One can then, in principle, solve the order conditions in terms of the $2\mathcal{N}$-storage coefficients $\alpha_i$ and $\beta_i$, and obtain valid low-storage RK schemes. For $o = 3, s = 3$ (classical 3rd order RK schemes), it can be seen that there are 5 such constants $(\alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3)$ and four order conditions to be satisfied. It seems hence plausible, and this has been confirmed by Williamson [50], that such schemes can be constructed. For $o = 4, s = 4$ there are eight order conditions and only seven free coefficients. Therefore, as has also been shown in [50], such schemes do not exist.

The dissipation and dispersion properties of the RK methods are closely related to their stability. They are usually studied using the model linear equation $dQ/dt = \phi Q$, $Q(0) = 1$, with $\phi$, an eigenvalue of the linearized operator of $\mathbf{F}$, being possibly complex [48]. Using equation (3.2), it can be found that the amplification factor of a RK method is given by:

$$r(\mathbf{z}) = \frac{q^n}{q^{n-1}} = 1 + \mathbf{z}b^T \left(I + \mathbf{z}A + \mathbf{z}^2A^2 + \ldots + \mathbf{z}^{s-1}A^{s-1}\right) \mathbf{1}^T \qquad (3.5)$$

where $\mathbf{z} = \phi\Delta t$. In this equation, $A$ is the matrix having the RK coefficients $a_{ij}$ as its entries, with $a_{ij} = 0$ for $j \geq i$ for an explicit method. Also, $b = [b_1, b_2, \ldots, b_s]^T$, $I$ is the $s \times s$ unit matrix, and $\mathbf{1}^T = [1, 1, \ldots, 1]^T$. In order to use the expression for $r(\mathbf{z})$ to build LDDRK schemes, it is convenient to write it explicitly as:

$$r(\mathbf{z}) = 1 + \psi_1\mathbf{z} + \ldots + \psi_s\mathbf{z}^s \qquad (3.6)$$

where the constants $\psi_i$ are function of the coefficients of the method and are given for $s \leq 6$ by:

$$\begin{aligned} \psi_1 &= \sum b_i \ ; & \psi_4 &= \sum b_i a_{ij} a_{jk} c_k \\ \psi_2 &= \sum b_i c_i \ ; & \psi_5 &= \sum b_i a_{ij} a_{jk} a_{kl} c_l \\ \psi_3 &= \sum b_i a_{ij} c_j \ ; & \psi_6 &= \sum b_i a_{ij} a_{jk} a_{kl} a_{lm} c_m \end{aligned} \qquad (3.7)$$

It is important to note that some of the sums in (3.7) also appear in the order conditions, equations (3.3), and hence are specified by the order of the method. The method will be stable for all values of $\mathbf{z}$ that satisfy $|r(\mathbf{z})| \leq 1$. This defines the stability region in the complex $\mathbf{z} = \phi\Delta t$ plane. If $s = o$, which is possible [48] only for $o \leq 4$, all coefficients $\psi_i$ are determined by the order conditions. This is the reason why all classical RK methods, which have the number of stages $s$ equal to the order $o$ of the method, have the same stability region.

Since the exact amplification factor is $r_e(\mathbf{z}) = e^{\mathbf{z}}$, expressing the ratio:

$$\frac{r(\mathbf{z})}{r_e(\mathbf{z})} = \Upsilon e^{-\mathbf{i}\Psi} \tag{3.8}$$

gives the dissipation error $1 - \Upsilon$ and the phase (dispersion) error $\Psi$. One can then minimize the dissipation and dispersion errors using different criteria depending on the problem, since they become functions of only the RK coefficients and $\mathbf{z}$. For the linear wave equation $\partial Q/\partial t + a\partial Q/\partial x = 0$, for example, it has been shown by Hu *et al.* [49] that the eigenvalue is $\phi = -\mathbf{i}ak^*$, with $k^*$ the modified wavenumber of the spatial discretization scheme. Requiring that the coefficients $\psi_i$ be such that the integral $\int_0^{\mathcal{L}} |r(\mathbf{z}) - r_e(\mathbf{z})|^2 d(ak^*\Delta t)$ has a minimal value (where $\mathcal{L}$ specifies the limit of the optimization range), while still maintaining a certain order of accuracy, leads to optimal values for those coefficients $\psi_i$ that are not determined by the order conditions. The LDDRK methods obtained in this way by Hu *et al.* [49], and that have enough free coefficients to be put in the $2\mathcal{N}$-storage format, are the following:

1. Second-order, five-stage scheme (LDD25) with the following parameters: $\psi_3 = 0.166558$, $\psi_4 = 0.0395041$ and $\psi_5 = 0.00781071$, stable up to $\Im(\phi\Delta t) = 3.54$.

2. Fourth-order, six-stage scheme (LDD46) with $\psi_5 = 0.0078105$ and $\psi_6 = 0.00132141$, stable up to $\Im(\phi\Delta t) = 1.65$.

3. Two-step fourth-order scheme (LDD56) for which the first step is a five-stage scheme with $\psi_5 = 0.0036105$, and the second step is a six-stage scheme with $\psi_5 = 0.0121101$ and $\psi_6 = 0.00285919$, stable up to $\Im(\phi\Delta t) = 2.85$.

A clear picture of the effect of the optimization can be obtained using the plots of the phase error $\Psi$ (which should be as close to zero as possible) and the amplitude ratio $\Upsilon$ (which should be equal to one in the ideal case) versus $ak^*\Delta t$. For the classical fourth order RK methods, these plots are given in figure 3.1. It may be noticed that both $\Psi$ and $\Upsilon$ depart from their desirable values for very low values of $ak^*\Delta t$. Between the optimized methods, LDD46 is chosen here for exemplification, and the corresponding plots are shown in figure 3.2. Notice that the plots have another scale than those in figure 3.1. Both the phase error and amplitude ratio are much closer in this case to their optimal value over a wider range of $ak^*\Delta t$. In particular, it can be found that for values of $ak^*\Delta t$ for which the methods are all stable ($\Upsilon \leq 1$), the optimized LDDRK methods have lower error levels than usual RK methods. This also means that, under the same spatial discretization (hence with a fixed $k^*$), larger time steps can be taken for the same error with the LDDRK methods, so that they may become very efficient.

To obtain the 2N-storage form of the optimized methods, the usual RK coefficients $a_{ij}$, $b_i$, in terms of which the order conditions are given, have to be expressed in terms of the 2N-storage coefficients $\alpha_i$, $\beta_i$. These relations depend on the number of stages of the method and are given in the subsequent sections.

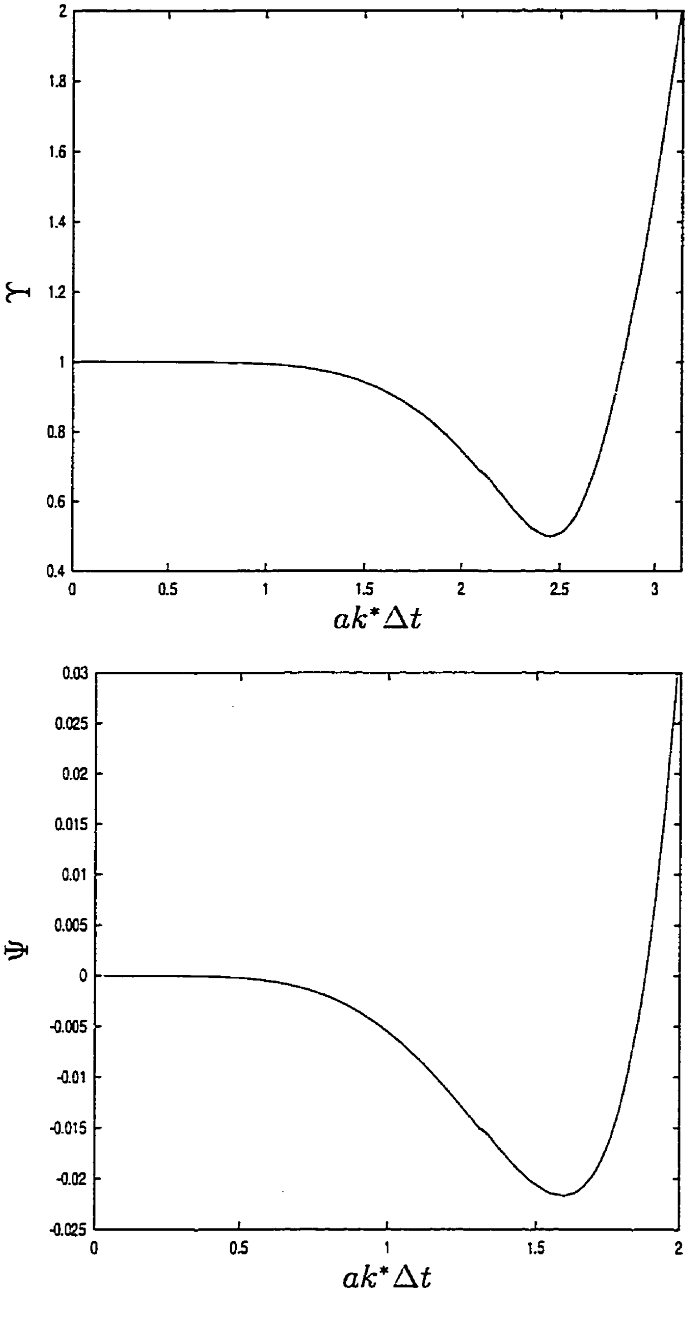


Figure 3.1: Amplitude ratio $\Upsilon$ and phase error $\Psi$ for the classical fourth order Runge-Kutta methods.

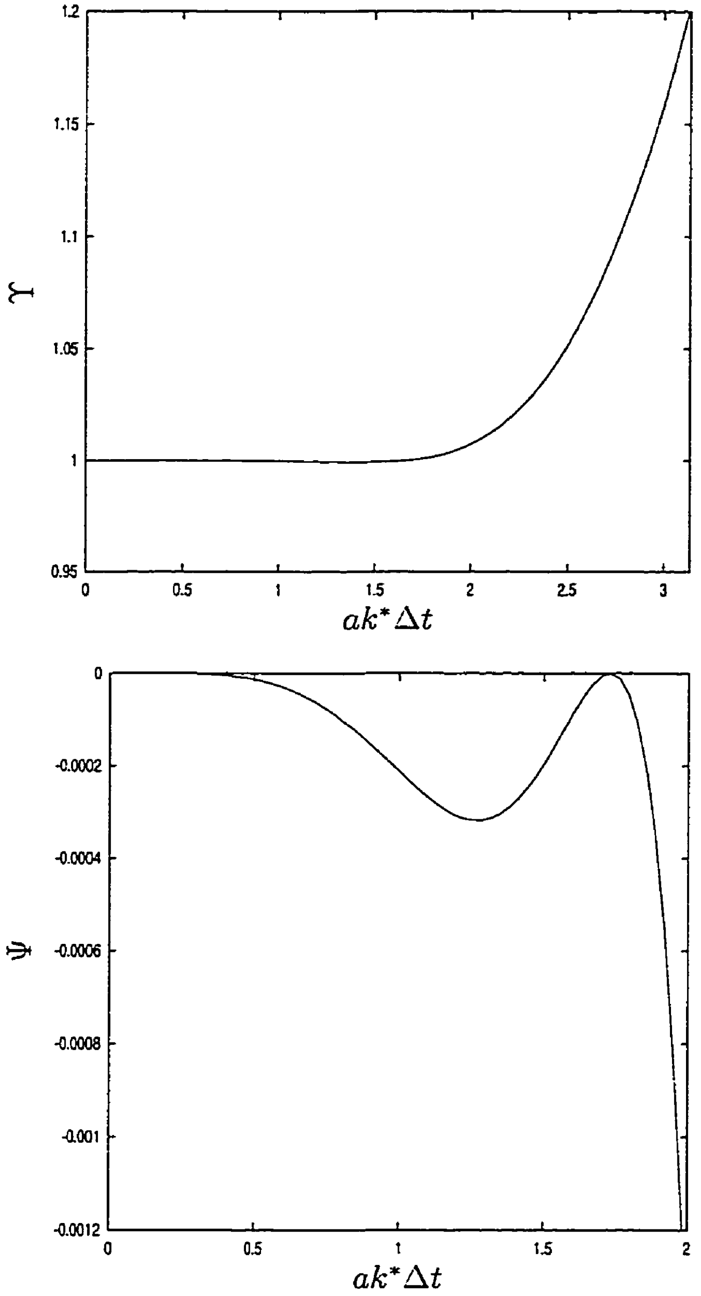


Figure 3.2: Amplitude ratio $\Upsilon$ and phase error $\Psi$ for the optimized LDD46 method.

## 3.3 Five-stage methods

In this case, using equations (3.2) and (3.4), the relationships between the usual and the 2N-storage RK coefficients are found to be:

$$
\begin{array}{ll}
a_{21} = \beta_1 & a_{53} = \alpha_4 a_{54} + \beta_3 \\
a_{32} = \beta_2 & a_{52} = \alpha_3 a_{53} + \beta_2 \\
a_{31} = \alpha_2 a_{32} + \beta_1 & a_{51} = \alpha_2 a_{52} + \beta_1 \\
a_{43} = \beta_3 & b_5 = \beta_5 \\
a_{42} = \alpha_3 a_{43} + \beta_2 & b_4 = \alpha_5 b_5 + \beta_4 \\
a_{41} = \alpha_2 a_{42} + \beta_1 & b_3 = \alpha_4 b_4 + \beta_3 \\
a_{54} = \beta_4 & b_2 = \alpha_3 b_3 + \beta_2 \\
\multicolumn{2}{c}{b_1 = \alpha_2 b_2 + \beta_1}
\end{array}
\tag{3.9}
$$

To obtain the second-order five-stage LDDRK scheme, the above relations are used to express the two order conditions that must be obeyed, denoted by $(o = 1)$ and $(o = 2)$ in equation (3.3), and the three additional constraints obtained by specifying $\psi_3$, $\psi_4$ and $\psi_5$ in terms of $\alpha_2, \ldots, \alpha_5$ and $\beta_1, \ldots, \beta_5$. This leads to a nonlinear system of five equations with nine unknowns, hence a four-parameter family of solutions may exist. To choose a solution one may, for example, impose the values for several variables and/or use additional equations. A solution thus obtained is presented in table 3.1.

| i | $\alpha_i$ | $\beta_i$ | $c_i$ |
|---|---|---|---|
| 1 | 0.0 | 0.1 | 0.0 |
| 2 | -0.691306507590891 | 0.75 | 0.1 |
| 3 | -2.65515560104995 | 0.7 | 0.331520119306831 |
| 4 | -0.814768857645745 | 0.479313317701 31 | 0.45777964054243 |
| 5 | -0.668658730443832 | 0.31039285385376 | 0.866652849930714 |

Table 3.1: $2\mathcal{N}$-storage second order five-stage (LDD25) scheme

## 3.4 Six-stage methods

For six-stage methods, the coefficients $a_{21}, \ldots, a_{54}$, $b_1, \ldots, b_4$ are still given by the relations in equation (3.9). The remaining coefficients are:

$$\begin{aligned} a_{65} &= \beta_5 & a_{62} &= \alpha_3 a_{63} + \beta_2 \\ a_{64} &= \alpha_5 a_{65} + \beta_4 & a_{61} &= \alpha_2 a_{62} + \beta_1 \\ a_{63} &= \alpha_4 a_{64} + \beta_3 & b_6 &= \beta_6 \\ & & b_5 = \alpha_6 b_6 + \beta_5 & \end{aligned} \tag{3.10}$$

There are now eleven free coefficients $\alpha_2, \ldots, \alpha_6$, $\beta_1, \ldots, \beta_6$. To obtain the fourth order six-stage LDDRK scheme the coefficients must obey the eight order conditions in equation (3.3) and the additional optimization constraints resulting from the specification of $\psi_5, \psi_6$. The resulting nonlinear system has been solved upon imposing the value of one of the coefficients. One such solution is given in table 3.2.

| i | $\alpha_i$ | $\beta_i$ | $c_i$ |
|---|---|---|---|
| 1 | 0.0 | 0.145309585177875 | 0.0 |
| 2 | -0.491957542000342 | 0.465379788883625 | 0.145309585177875 |
| 3 | -0.894626417580752 | 0.467539741872758 | 0.381742277025673 |
| 4 | -1.552667803218557 | 0.779527988100590 | 0.636781370437459 |
| 5 | -3.407797355404573 | 0.357432717815297 | 0.756074449632355 |
| 6 | -1.074264041075980 | 0.15 | 0.927104723987567 |

Table 3.2: $2\mathcal{N}$-storage fourth order six-stage (LDD46) scheme

## 3.5 Two-step methods

The two-step LDD56 scheme devised in [49] can be also put in $2\mathcal{N}$-storage format. The scheme has fourth order accuracy in both steps, and five/six stages in the first/second step. For the first step, the five stages imply nine free coefficients which are completely determined by the eight order conditions and the additional constraint $\psi_5 = 0.0036105$. The second step is completely similar to LDD46, except for the values of the constants $\psi_5$ and $\psi_6$. Tables 3.3 and 3.4 list one set of coefficients for the first and second step, respectively.

| i | $\alpha_i$ | $\beta_i$ | $c_i$ |
|---|---|---|---|
| 1 | 0.0 | 0.268745438887134 | 0.0 |
| 2 | -0.605122643328622 | 0.801470697322080 | 0.268745438887134 |
| 3 | -2.043756402347613 | 0.505157042694227 | 0.585228069295243 |
| 4 | -0.740699906375441 | 0.562356803790002 | 0.682706644784246 |
| 5 | -4.423176513029681 | 0.059006551277588 | 1.164685483772926 |

Table 3.3: $2\mathcal{N}$-storage form for the first step of LDD56

| i | $\alpha_i$ | $\beta_i$ | $c_i$ |
|---|---|---|---|
| 1 | 0.0 | 0.115848881812855 | 0.0 |
| 2 | -0.441273771538773 | 0.372876990516528 | 0.115848881812855 |
| 3 | -1.073982008079781 | 0.737953689214352 | 0.324185036404128 |
| 4 | -1.706357079125675 | 0.579811093663110 | 0.619320820351777 |
| 5 | -2.797929316268244 | 1.031284991300145 | 0.803447266633590 |
| 6 | -4.091353712091916 | 0.15 | 0.918416644520659 |

Table 3.4: $2\mathcal{N}$-storage form for the second step of LDD56

## 3.6 Numerical validation

In order to test the accuracy of the proposed methods for nonlinear problems, a system of nonlinear and non-autonomous first order differential equations,

$$Q_1' = \frac{1}{Q_1} - \frac{Q_2 e^{t^2}}{t^2} - t; \; Q_2' = \frac{1}{Q_2} - e^{t^2} - 2te^{-t^2} \tag{3.11}$$

is solved here, with the initial conditions specified as $Q_1(1) = 1$, $Q_2(1) = e^{-1}$. The exact solution of this system is $Q_1(t) = 1/t$, $Q_2(t) = e^{-t^2}$. The system is solved numerically in double precision over the range $t \in [1, 1.4]$ using several step sizes $\Delta t$. In the limit $\Delta t \to \infty$ a decrease of $\Delta t$ by a factor of two should decrease the error by a factor $O = 16$ for a fourth order method, and $O = 4$ for a second order method. The global error norms (computed as $|Q_1 - q_1| + |Q_2 - q_2|$) at $t = 1.4$ and their ratios, listed in table 3.5, clearly show that LDD46 and LDD56 are fourth order accurate for nonlinear systems, while LDD25 is second order accurate.

The performance of the methods for wave propagation problems is tested us-

| $\Delta t$ | LDD46 | $O$ | LDD56 | $O$ | LDD25 | $O$ |
|---|---|---|---|---|---|---|
| 1e-2 | 5.16e-8 | 17.8 | 4.31e-8 | 15.7 | 6.29e-7 | 5.9 |
| 5e-3 | 2.89e-9 | 16.9 | 2.74e-9 | 15.9 | 1.05e-7 | 5.2 |
| 2.5e-3 | 1.7e-10 | 16.5 | 1.73e-10 | 16.0 | 2.01e-8 | 4.7 |
| 1.25e-3 | 1.04e-11 | | 1.08e-11 | | 4.29e-9 | |

Table 3.5: Error norms and their ratios for various step sizes for the nonlinear system (3.11).

ing the convective wave equation, for which they have been optimized,

$$\frac{\partial Q}{\partial t} + a\frac{\partial Q}{\partial x} = 0 \ , \quad Q(x, t = 0) = 0.5e^{-x^2/9} \tag{3.12}$$

with $a = 1$ (convection of a Gaussian pulse). The domain extends from $x = -50$ to $x = 450$, and two spatial discretizations have been used in order to show the importance of the time integration method. The first discretization is obtained using eighth order central differences with $\Delta x = 1$, while the second is the staggered grid method presented in chapter 2, using twenty elements. For the central finite difference discretization, two cases have been considered. In a first case the time step for all schemes has been chosen close to the stability limit of the fourth order six-stage scheme. For the second case, the time step is larger than the stability limit of LDD46 and close to the stability limit of LDD56. The maximum norm of the error $L_\infty = \max|q - Q|$ at time $t = 400$ for the 2$\mathcal{N}$-storage schemes devised above in the two cases is given in table 3.6. Also given is the error for the fourth order 2$\mathcal{N}$-storage scheme developed by Carpenter (CAR) [51] and the classical fourth order four-stage RK (RK4) method which needs at least 3$\mathcal{N}$ storage. It turns out that for $\Delta t < 1.54$ the error for LDD56 is governed by the spatial discretization, no further decrease of the error being possible upon decreasing $\Delta t$. Figure 3.3 presents graphically the results for the first case (RK4 not shown), with the exact solution sampled at the same data points.

| Method | $\Delta t = 1.263$ | $\Delta t = 1.543$ |
|---|---|---|
| RK4 | 1.05e-1 | 1.41e-1 |
| CAR | 8.12e-2 | 1.01e-1 |
| LDD25 | 3.77e-2 | 5.08e-2 |
| LDD46 | 2.80e-2 | - |
| LDD56 | 2.43e-2 | 2.44e-2 |

Table 3.6: $L_\infty$ error norms for the linear convection equation discretized with eighth order central differences (401 points).

| Method | $N = 12$ | $N = 13$ |
|---|---|---|
| RK4 | 1.93e-1 | 1.63e-1 |
| CAR | 1.52e-1 | 1.21e-1 |
| LDD25 | 8.62e-2 | 6.46e-2 |
| LDD46 | 7.36e-2 | 5.06e-2 |
| LDD56 | 7.31e-2 | 5.04e-2 |

Table 3.7: $L_\infty$ error norms for the linear convection equation discretized with the staggered grid method, 20 elements.

For the staggered grid spectral discretization, only the case of the highest time step allowed by LDD46 has been considered. Due to the better resolution properties of this spatial discretization, a lower number of points (240 and 260 points, corresponding to $N = 12$ and $N = 13$) is used than for the central differences discretization, such that the phase and amplitude errors can still manifest themselves. The corresponding error norms are given in table 3.7, and the results are represented graphically in figure 3.4.

Different ways can be used to compare the relative efficiency of the methods. Among them, RK4 can be considered the most efficient if accuracy is not a concern, since it needs the smallest number of function evaluations to reach $t = 400$. When one compares the work needed to obtain a certain accuracy, however, the comparison favors the optimized methods. For an error norm $L_\infty$=2.8e-2 for example, where one is limited by the stability of LDD46 when using the central differences discretization, LDD25 needs a total of 2070 function evaluations (414 steps), LDD46 1914, CAR 3530 and RK4 3600, while the error for LDD56, at its stability limit where 1430 stages are needed, is lower than the asked-for value. Considering the work required by RK4 as a reference, it follows that LDD25 is 1.74 times, LDD46 1.88 times, and LDD56 at least 2.52 times more efficient for this problem. It becomes thus clear that the optimized methods should be preferred for the computation of propagating waves.

## 3.7 Time step size for the Euler equations

For the Euler equations the stability of the time integration method has to be studied on their matrix form

$$\frac{dQ}{dt} = \Phi Q \tag{3.13}$$

where $\Phi$ is supposed to incorporate the effects of the spatial operator acting on $Q$ as well as any boundary conditions. Under the assumption that $\Phi$ is diagonalizable, the time step $\Delta t$ has to be chosen such that $\phi\Delta t$ is within the stability region for any eigenvalue $\phi$ of $\Phi$. The stability regions for two RK methods used in this thesis are given in figure 3.5. In general, they extend more along the imaginary axis than along the negative real axis, which is advantageous since wave dominated problems tend to have complex eigenvalues.

The matrix $\Phi$ can be constructed only in very particular cases. For example, for the case of the linear wave equation, if $Q$ is uniform and no boundary conditions are taken into account, the spectral discretization presented in chapter 2 leads to $\Phi = a\mathbf{DI}$, and $Q$ becomes the vector of the Gauss-Chebyshev point values $\mathbf{Q}_\mathbb{G}$ which are advanced in time using equation (3.4) or equivalently (3.13). It is obviously impractical to form the matrix and study its eigenvalues for computing the maximum time step for stability. It is known, however, that the eigenvalues of the matrices that appear when a spectral discretization is used for the space terms are complex numbers, with the real part negative, and satisfying $|\phi| = \mathcal{O}(N^2)$ in the limit $N \rightarrow \infty$. To maintain the stability, the time step has to decrease with an order $\mathcal{O}(1/N^2)$, as does also the minimal grid spacing $\Delta\xi_m = \xi^1 - \xi^0$ between the end Lobatto points. In practice, this grid spacing is used to compute the time step that ensures stability. Hence, for the linear wave equation, the limit on $ak^*\Delta t$ leads to an approximate formula for computing the time step,

$$\Delta t = \frac{\mathcal{C}\Delta x_m}{a} \tag{3.14}$$

Here, $\mathcal{C}$ is a constant (a Courant-Friedrichs-Lewy number) that is chosen for a particular method to be smaller than the extension of its stability region on the imaginary axis, and $\Delta x_m$ is the distance in the physical plane corresponding to $\Delta\xi_m$.

For the Euler equations, the wave speed must be replaced with the maximum wave propagation speed which the equations support, which is $|\vec{V}| + c$, and the computation has to be done at each point since it varies within the flow field. A formula which has been found to give good results in this case is:

$$\Delta t = \frac{\mathcal{C}}{\max\left(\frac{V_\xi}{\Delta\xi} + \frac{V_\eta}{\Delta\eta} + \frac{V_\zeta}{\Delta\zeta}\right)} \tag{3.15}$$

where the wave speeds along each $\xi$-axis are the contravariant components

$$V_{\xi_i} = \frac{\partial \xi_i}{\partial x_j}(c + |v_j|) \tag{3.16}$$

and the maximum is searched across all $\mathbb{G}^d$ points in the flow field.

## 3.8 Implementation details

It can be seen from equation (3.4) that the $2\mathcal{N}$-storage Runge-Kutta methods are in fact easier to implement than methods that use the form (3.2). The algorithm can use an integer switch, that is input as data at the start of the program, and selects the number of stages $s$ and the coefficients $\alpha_i, \beta_i$ of the desired scheme. This is very useful since a different scheme can be used for steady state computations than for acoustic computations. Indeed, the higher dissipation properties of non-optimized schemes, such as the scheme of Carpenter and Kennedy [51] or Williamson [50] are desirable for steady state computations, since they tend to reduce the time needed for elimination of the transients.

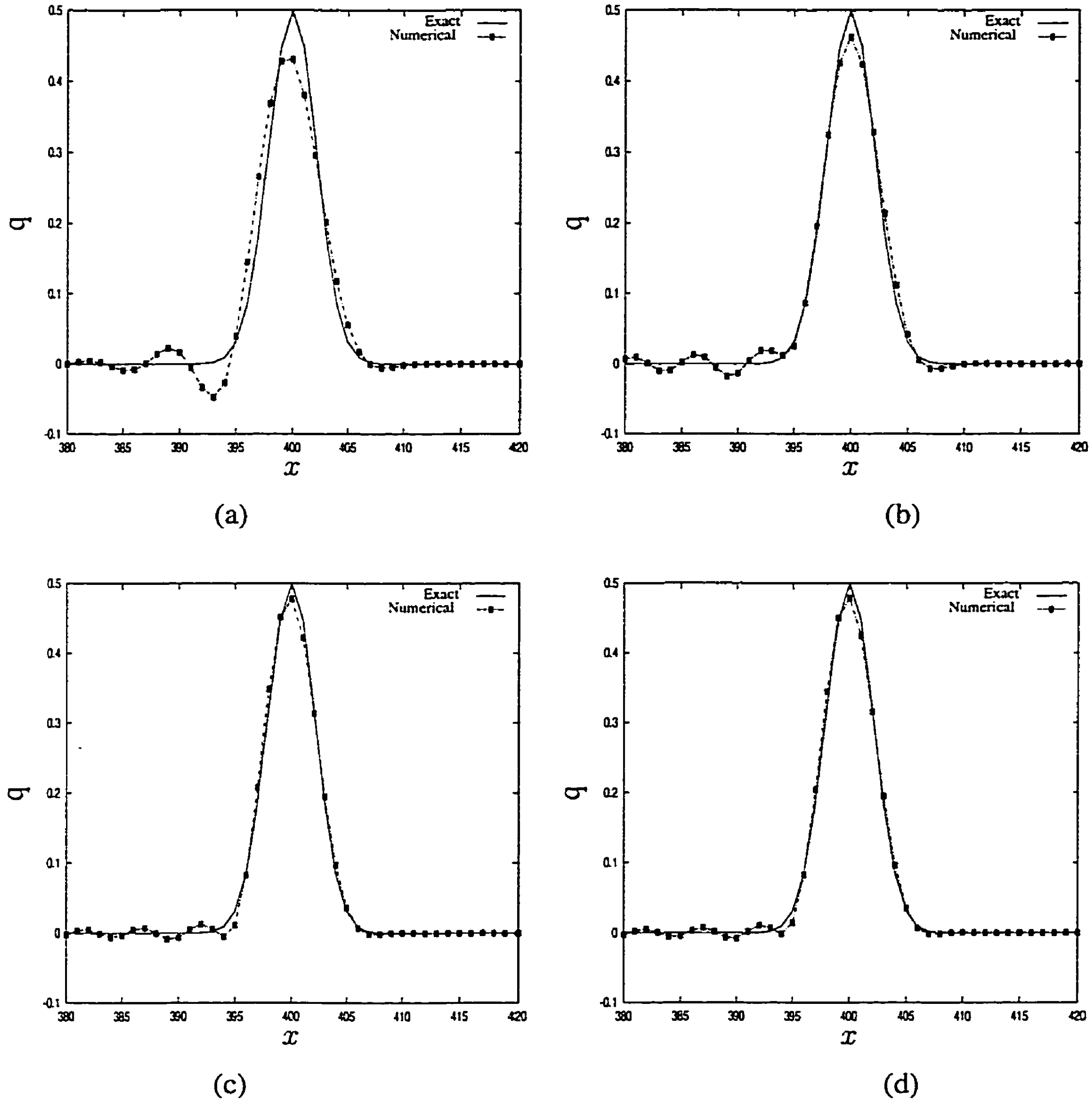
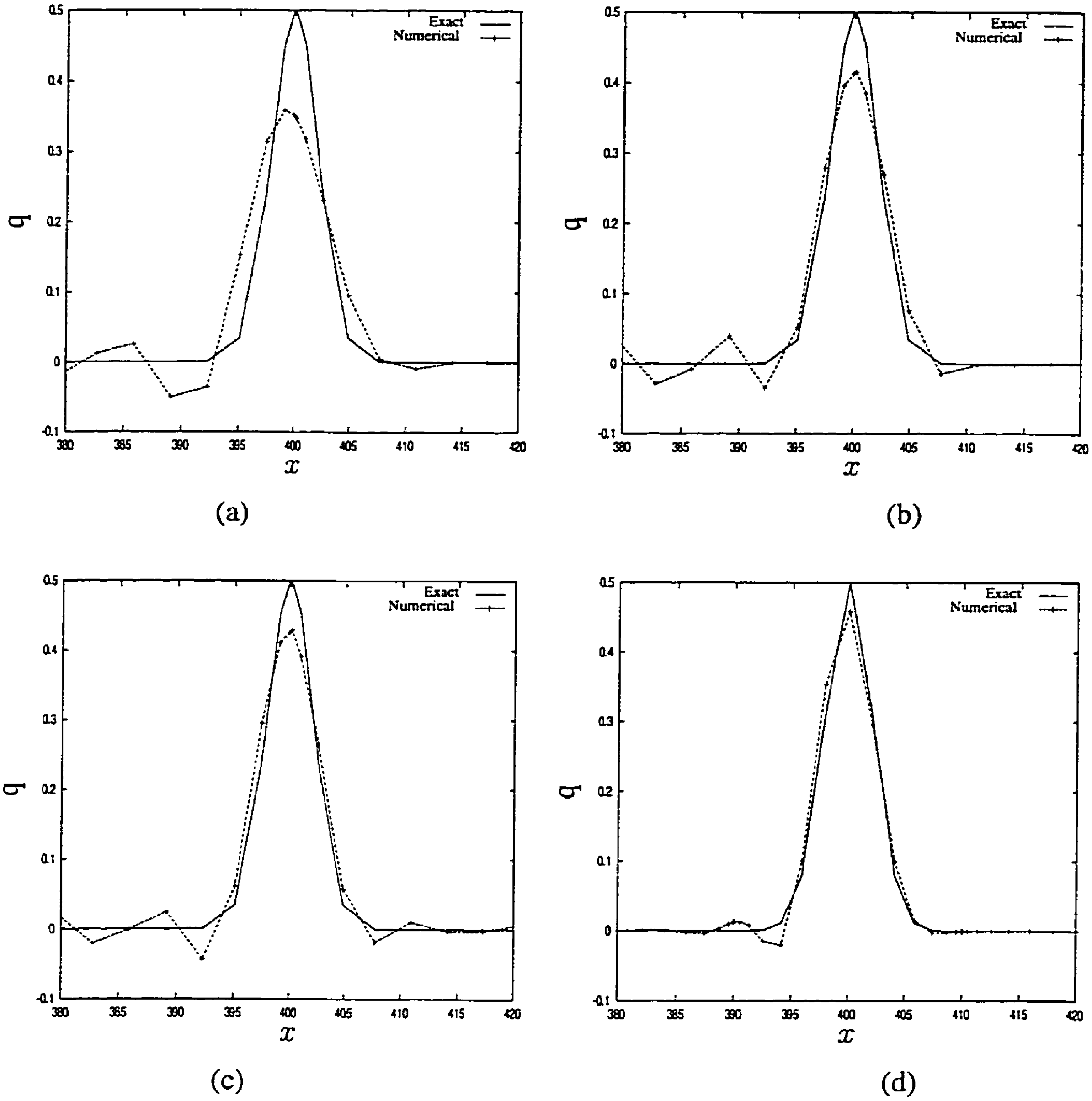


Figure 3.3: Results obtained with 2N-storage Runge-Kutta schemes for the linear convection equation, central difference discretization: (a) Carpenter's 5-stage scheme (b) LDD25 2nd order 5-stage scheme (c) LDD46 4th order 6-stage scheme (d) LDD56 two-step scheme.



(a)

(b)

(c)

(d)

Figure 3.4: Results obtained with 2N-storage Runge-Kutta schemes for the linear convection equation, staggered grid spectral discretization, $N = 12$: (a) Carpenter's 5-stage scheme (b) LDD25 2nd order 5-stage scheme (c) LDD46 4th order 6-stage scheme (d) LDD56 two-step scheme.

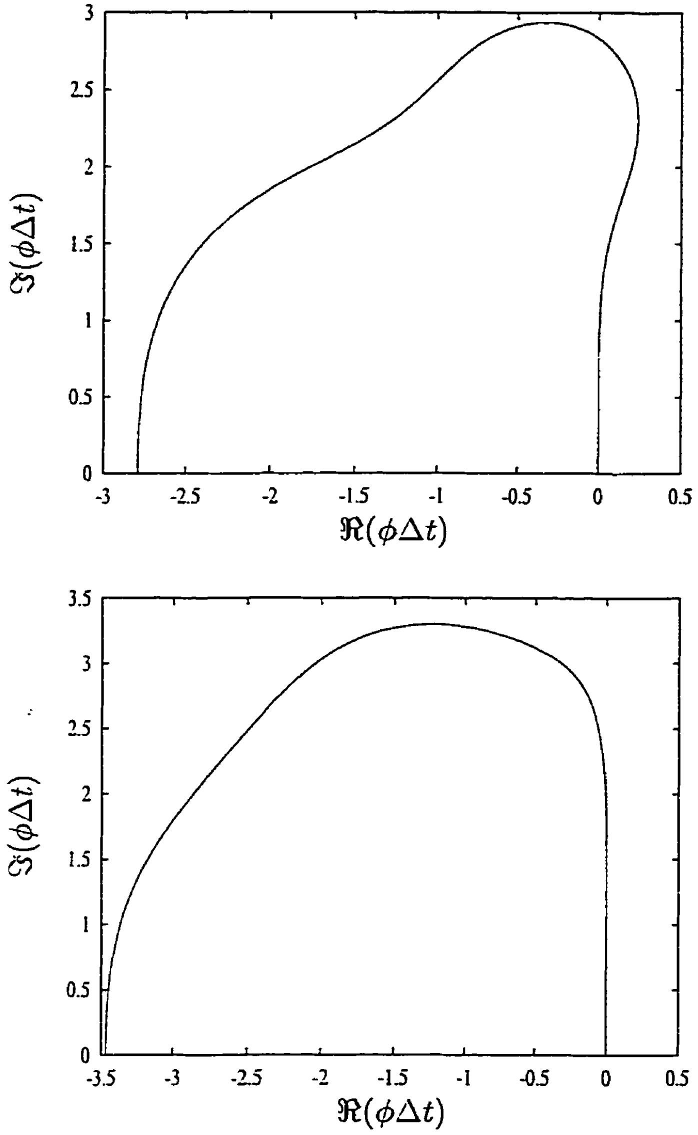


Figure 3.5: Stability regions; RK methods are stable for $\phi h$ inside the region closed by the curved contour. Top: classical fourth order RK method, stability limit along imaginary axis 2.82. Bottom: LDD46, stability limit along imaginary axis 1.65.

# Chapter 4

# Boundary conditions

## 4.1 Solid surfaces

Since the Euler equations describe the flow of inviscid fluids, the proper boundary condition at the solid surfaces present in the flow field is the flow tangency condition. This boundary condition is valid for both steady state and unsteady acoustic computations. It is straightforward to enforce this boundary condition using the Riemann solver. In this case, one set of pointwise values for the Riemann problem is obtained through interpolation from the element that has a solid boundary face, while the other set is obtained by 'reflecting' the first set such that the resulting normal velocity on the solid surface is zero. Let interpolated variables from inside the domain be $(\rho_d, \vec{V}_d, p_d)$, and let the local unit vector normal to the solid surface at the flux definition point under consideration be $\vec{n}$. The states that define the Riemann problem are then $(\rho_d, \vec{V}_d \cdot \vec{n}, p_d)$ and $(\rho_d, -\vec{V}_d \cdot \vec{n}, p_d)$, as this results in a solution with $\vec{V} \cdot \vec{n} = 0$. This solution is then used to obtain the boundary fluxes. The velocity components that are tangent to the surface are left unchanged.

## 4.2 Steady-state boundary conditions

### 4.2.1 Fan face outlet boundary

The treatment of the boundary condition at the fan face is based on an analysis of the flow in a frame normal to the boundary using the theory of characteristics [37]. To describe it, suppose the boundary is a $x$=constant plane, and the flow comes from right to left (the computational domain is to the right of the fan), hence $u \leq 0$. The case of supersonic flow, $|u| > c$, is trivial, the boundary fluxes being computed in this case from the state variables interpolated from the element next to the boundary. For subsonic flow, the analysis shows that one of the characteristics ($\mathrm{C}_+$) has a positive slope ($dx/dt = u + c$) in the $(x, t)$ plane, hence the information pertaining to this characteristic has to be provided from the exterior of the domain. The remaining characteristics for $d = 1$ ($\mathrm{C}_0$ and $\mathrm{C}_-$) have the negative slopes $dx/dt = u$ and $dx/dt = u - c$, respectively, and the corresponding information must be provided from the interior of the domain by interpolation, as well as for the $d - 1$ additional characteristics that appear for $d > 1$, and have the slope $dx/dt = u$.

The flow being locally isentropic, the characteristic variables can be integrated to give the Riemann invariants that should be left unchanged along these characteristics. The corresponding Riemann invariants are $\mathrm{R}_+ = u + 2c/(\gamma - 1)$, $\mathrm{R}_0 = p/\rho^\gamma$ (the entropy) and $\mathrm{R}_- = u - 2c/(\gamma - 1)$, plus the tangential velocity components for the additional characteristics. While the invariants can be readily obtained from the interior, it is not common to specify the boundary condition at the outlet by the value of $\mathrm{R}_+$. Instead, the value of the outlet pressure $p_o$ is usually known, and this information has to replace the one from $\mathrm{R}_+$. Letting values interpolated from the interior domain be denoted by the subscript '*d*' and the searched-for boundary values be denoted by the subscript '*b*', the following relations can then be found:

$$
\begin{aligned}
p_b &= p_o\,; & \rho_b &= \rho_d \left(\frac{p_b}{p_d}\right)^{\frac{1}{\gamma}} \\
c_b &= \sqrt{\gamma\frac{p_b}{\rho_b}}\,; & u_b &= u_d + \frac{2}{\gamma-1}(c_b - c_d) \\
v_b &= v_d\,; & w_b &= w_d
\end{aligned}
\tag{4.1}
$$

The computed boundary values can then be used directly to compute the boundary fluxes. The alternative used in this thesis is however to compute the boundary fluxes using the solution of the Riemann problem between the boundary variables (subscript *b*) and the interior variables (subscript *d*). Thus, only minor modifications are needed to use the same part of the code for specifying the incoming acoustic modes at the fan face.

### 4.2.2 Far-field boundaries

Boundary conditions at far-field boundaries are imposed by computing boundary fluxes using the solution of the Riemann problem between the infinity state, denoted by subscript '$\infty$' and the state interpolated from the interior of the domain. This has the advantage that there is no need to distinguish between outlet boundaries (where the fluid leaves the domain) and inlet boundaries (where the fluid enters the domain), since the correct direction of wave propagation is ensured by the Riemann solution. Taking into account the reference quantities used to obtain the non-dimensional equations, and considering the flow from right to left as previously discussed and aligned with the $x$-axis, the infinity state variables are

$\rho_\infty = 1$, $p_\infty = 1/\gamma$, $u_\infty = -M_\infty$ and $v_\infty = w_\infty = 0$, where $M_\infty$ is the infinity Mach number that must be specified. The boundary fluxes are then defined by the solution of the Riemann problem between $(\rho_d, \vec{V}_d \cdot \vec{n}, p_d)$ and $(\rho_\infty, \vec{V}_\infty \cdot \vec{n}, p_\infty)$, where $\vec{n}$ is the unit vector normal to the boundary surface.

## 4.3 Boundary conditions for acoustic modeling

### 4.3.1 Incoming duct modes

At the source plane a general function of time and space can be specified as the acoustic perturbation in the present time domain approach. However, the present work is restricted to specifying a single duct acoustic mode at the source plane. For the case without mean flow, this is done by solving a Riemann problem with flow variables extrapolated from the interior on one side of the interface, and the mean state perturbed with the incoming acoustic mode on the other side. The incoming modes that have been used correspond to rectangular and to axisymmetric and 3D circular duct geometries, for which the duct eigenfunctions are cosine functions [36] and combinations of Bessel functions [6], respectively.

For the first case, when the medium is quiescent, a 2D duct acoustic mode has the form

$$\begin{bmatrix} p - \bar{p} \\ \rho - \bar{\rho} \\ v_1 - \bar{v}_1 \\ v_2 - \bar{v}_2 \end{bmatrix} = \mathcal{A} \begin{bmatrix} \cos(k_y y)\cos(\omega t - k_x x) \\ \frac{1}{\bar{c}^2}\cos(k_y y)\cos(\omega t - k_x x) \\ \frac{k_x}{\omega\bar{\rho}}\cos(k_y y)\cos(\omega t - k_x x) \\ \frac{k_y}{\omega\bar{\rho}}\sin(k_y y)\sin(\omega t - k_x x) \end{bmatrix} \tag{4.2}$$

In this equation the overbar denotes the mean flow variables, $y$ is the cross section coordinate non-dimensionalized by the duct cross dimension, $\omega$ is the driving frequency, and the axial wave number is $k_x = \sqrt{(\omega/c)^2 - k_y^2}$, with $k_y$ determined such that the boundary condition $v_2 = 0$ is satisfied at the duct walls.

For circular geometries, when there is no flow, a spinning mode with azimuthal order $m$ and radial order $\mu$, usually denoted by $(m, \mu)$, is specified by

$$\begin{bmatrix} p - \bar{p} \\ \rho - \bar{\rho} \\ v_x - \bar{v}_x \\ v_r - \bar{v}_r \\ v_\theta - \bar{v}_\theta \end{bmatrix} = A \begin{bmatrix} E_m(k_{m\mu}r)\cos(k_x x + m\theta - \omega t) \\ \frac{1}{\bar{c}^2} E_m(k_{m\mu}r)\cos(k_x x + m\theta - \omega t) \\ \frac{k_x}{\omega\bar{\rho}} E_m(k_{m\mu}r)\cos(k_x x + m\theta - \omega t) \\ \frac{k_{m\mu}}{\omega\bar{\rho}} E_m'(k_{m\mu}r)\sin(k_x x + m\theta - \omega t) \\ \frac{m}{r\omega} E_m(k_{m\mu}r)\cos(k_x x + m\theta - \omega t) \end{bmatrix} \qquad (4.3)$$

Here $r$ is the radius non-dimensionalized by the duct radius $R_0$, and $E_m(k_{m\mu}r) = J_m(k_{m\mu}r) + \vartheta Y_m(k_{m\mu}r)$ is the duct eigenfunction, with $J_m$ and $Y_m$ the Bessel functions of first and second kind, respectively. The axial wave number is $k_x = \sqrt{(\omega/c)^2 - k_{m\mu}^2}$, and $v_x$, $v_r$ and $v_\theta$ are the axial, radial and circumferential components of the velocity, respectively, from which the Cartesian components $v_i$ can be determined easily. The waviness of the Bessel functions, $k_{m\mu}$, and the factor $\vartheta$ are determined by the boundary conditions at the casing and hub, with $\vartheta = 0$ in the case there is no center body.

For non-uniform flow cases, the source plane is treated as a subsonic outlet boundary where the pressure $p_o$ is obtained as above, while the other variables are computed using the solution from inside the domain as in equation (4.1).

### 4.3.2 Radiation boundary conditions

At the far-field boundaries where the computational domain $\mathcal{D}$ is truncated from infinity, the acoustic perturbations coming from the interior of the domain must propagate as if this boundary did not exist and the computational domain extended to infinity. Waves propagating from the boundary towards the interior should be excluded because they violate the causality principle [35]. However, in a numerical computation, such waves are usually generated when acoustic perturbations impinge on the boundary, either because of the boundary condition itself or because of discretization errors inherent to the numerical method. The amplitude of these reflected waves must be kept of the order of the truncation error, otherwise the computed acoustic field can become meaningless. The term 'radiation boundary conditions' is used hereafter for the boundary conditions used to this end.

Several radiation boundary conditions have been used in this thesis. They can be considered to belong to three distinct types, namely characteristic, asymptotic form and buffer region boundary conditions. In the first group, a method that is equivalent to the characteristic boundary conditions developed by Thompson [53], with the difference that the equations are not linearized, is to treat the far-field boundary as in section 4.2.2. This is the first method that has been implemented in the algorithm. It works reasonably well for waves impinging with normal incidence on the boundary, as has been also reported by Hixon *et al.* [54]. However, large reflections appear if the boundary is not far enough from the source, in particular from the corners of the domain.

Excellent results can, however, be obtained with this type of method if the Riemann problem is projected on the ray path, instead of on the normal at the exterior boundary. The direction of the ray path is computed using the position $(x^s, y^s, z^s)$ of the source, which can be considered at the point where the fan axis of symmetry intersects the leading edge plane of the inlet duct. Then one can construct, at the point $(x^b, y^b, z^b)$ on the boundary, the unit vector $\vec{b}$ with components

$$b_i = \frac{x_i^b - x_i^s}{r_b} \;\; ; \quad r_b = \sqrt{\sum_{i=1}^{3}(x_i^b - x_i^s)^2} \tag{4.4}$$

The boundary flux is then defined by the solution of the Riemann problem between the boundary state interpolated from the interior, $(\rho_d, \vec{V}_d \cdot \vec{b}, p_d)$ and the infinity state $(\rho_\infty, \vec{V}_\infty \cdot \vec{b}, p_\infty)$. This boundary condition is denoted here as the 'finite wave model' (FWM), and is similar to the one proposed by Atkins and Casper [55]. In their study, these authors replaced however the full Riemann problem by a simplified problem which considers only isentropic waves, thus excluding the case when shock waves impinge on the boundary. While this is a reasonable assumption, since the radiation boundaries should always be placed far enough from the region with high hydrodynamic disturbances, it was found that the savings in computer time it brought about were minimal. Therefore, for the sake of simplicity, the same Riemann solver was used at the radiation boundary as within the domain.

The FWM has been used to obtain most of the results in the case of a quiescent fluid. As can be noticed, the same method should also allow, in principle, steady-state computations. Indeed, the only difference from the method in section 4.2.2 is to use the vectors $\vec{b}$ in place of the normal vectors $\vec{n}$, which were chosen somewhat arbitrarily in order to make the Riemann problem unidimensional, as discussed in section 2.7. However, the application of the FWM to steady-state problems proved in some cases to be unstable. For this reason, it was not used for acoustic problems with non-uniform mean flow.

The second type of boundary conditions uses the asymptotic form of the acoustic wave, valid at large distances from the source, to obtain a partial differential equation that the boundary values must obey. To discretize this partial differential equation, which is different from the Euler equations, the solution and its spatial derivatives must be computed from the interior field. Such boundary conditions have been developed by Bayliss and Turkel [56, 57] to arbitrary order of accuracy and, from a different perspective, by Tam and Webb [21]. The partial differential equation proposed by Tam and Webb to update the boundary values has been implemented in the $d = 2$ version of the algorithm. It has the form:

$$\left(\frac{1}{V(\theta)}\frac{\partial}{\partial t} + \frac{\partial}{\partial r_b} + \frac{1}{\kappa r_b}\right)\begin{bmatrix} \rho - \bar{\rho} \\ v_1 - \bar{v}_1 \\ v_2 - \bar{v}_2 \\ p - \bar{p} \end{bmatrix} = 0 \tag{4.5}$$

with $\kappa = 2$ for slab symmetry and $\kappa = 1$ for axisymmetry. In this equation, $V(\theta) = \bar{v}_1 \cos\theta + \bar{v}_2 \sin\theta + \sqrt{c_\infty^2 - (\bar{v}_1 \sin\theta - \bar{v}_2 \cos\theta)^2}$ is the group velocity of the acoustic waves in the mean flow of velocity $(\bar{v}_1, \bar{v}_2)$, and $r_b$ and $\theta$ are the radial distance from the source and the angle made by the direction of the main flow with the radius from the source, respectively. The spatial derivative

$$\frac{\partial}{\partial r_b} = \cos\theta\frac{\partial}{\partial x} + \sin\theta\frac{\partial}{\partial y} \tag{4.6}$$

is computed by spectral differentiation in the first layer of elements next to the boundary.

Taking into account the theory of characteristics, imposition of all the boundary values computed from equation (4.5) would not correspond to the physics. Therefore, to obtain the correct direction of wave propagation, boundary fluxes are again defined using the Riemann solution between these boundary values and the state interpolated from the interior. Without any form of dissipation, this boundary condition showed instability for large values of $N$ for several of the test cases that involved long time integration. Stability was achieved in all cases by filtering the flux values before computing the derivatives. The filter, described in section 4.4, was applied only to the elements in the layer next to the boundary for efficiency reasons. However, even with filtering, the results for the FWM were slightly better than those obtained with this method, as will be shown in chapter 6. The method is denoted hereafter as Tam and Webb (TW).

The third category of boundary conditions uses a buffer region near the boundary, where the equations are modified in such a way as to either damp the waves, or force them to be convected towards the boundary, or both. Such boundary conditions have been developed by Ta'asan and Nark [58], who add an additional convection velocity, Colonius *at al.* [59], who use characteristic variables and a buffer region with temporal damping, grid stretching and filtering, and Freund [60], who also uses a form of temporal damping applied to the conservative variables in the buffer region. In the same category can be included the perfectly matched layer (PML) method first developed for the linearized Euler equations by Hu [61]. It uses a directional splitting of the Euler equations for temporal damping of the waves. While it seems to offer excellent results [62], the splitting increases the cost of the solution, and the method has been shown to be unstable [63, 64] unless a form of dissipation is added to the solver. This makes its use for spectral solvers less desirable. It is, however, useful to modify the equations with lower order damping terms that do not change the stability properties, and without splitting. This temporal damping tries to make the solution of the system of Euler equations mimic the behavior of the solution $Q = e^{-At}$ of the ordinary differential equation $dQ/dt = -AQ$. With $A$ a positive constant, $Q$ decreases exponentially in time.

To this end, a modified system of equations of the form:

$$\frac{\partial Q}{\partial t} + \sum_{i=1}^{d} \frac{\partial F_i}{\partial x_i} + vH\delta_{2d} = -\sigma(\vec{x})(Q - \bar{Q}) \tag{4.7}$$

is considered in this thesis in a layer of elements next to the boundary. Obviously, since the governing equations are modified, the solution in this layer has no more physical meaning. The positive value of $\sigma$ varies from zero to $\sigma_M$ according to a power law

$$\sigma(\vec{x}) = \sigma_M \sum_i \left( \frac{x_i - x_i^{int}}{x_i^{ext} - x_i^{int}} \right)^{\beta} \tag{4.8}$$

within these elements ($x_i^{int}$ and $x_i^{ext}$ are the coordinates of the interior and exterior limits of the absorbing layer, limits that lie along planes on which one coordinate is constant). The mean flow $\bar{Q}$ is used on the right hand side such that only the perturbations, which are the quantities that actually appear in the time derivative, are damped, and the solution $Q$ tends towards $\bar{Q}$ for large $t$. Hence, acoustic, entropy and vorticity waves will decrease exponentially with time upon their penetration in this layer as they propagate towards the boundaries. The most important advantage of this method, hereafter denoted as 'damping layer' (DL), is that it can handle uniform flow fields as well as non-uniform flows (after steady computation of the mean flow, the mean values $\bar{Q}$ are known) without any differences in the implementation. The implementation is also very simple, and steady state computations for $\tilde{Q}$ can be made with the same method, using $\sigma_M = 0$. In both cases, boundary fluxes can be computed as in the FWM or using the grid normals as in section 4.2.2.

## 4.4 Spurious waves and filtering

The replacement of the values of the solution $\tilde{Q}$ at the end of a $\xi$-line in an element as shown in figure 2.4 has the global aim of enforcing the continuity of the solution across elements. Locally however, supposing the interpolated data were smooth within an element before the replacement, this smoothness may disappear afterwards, even if the function to be approximated is globally smooth, due to truncation and round-off errors of the interpolation process. Although these errors decrease exponentially fast, they can lead to undesirable oscillations near element boundaries, as can be seen from figure 4.1 for the case of an approximation of the linear wave equation. These oscillations manifest themselves as high-frequency (above the resolved frequencies of the discretization) spurious waves which travel at their own speed [65] and are obviously undesirable.

These waves are one of the causes of the numerical instability that has been observed when discretizing equation (4.5), since they lead to wrong values of the time derivative when long time integrations are performed. In the context of spectral methods, such waves are controlled using filters. The filter modifies the coefficients of the higher degree Chebyshev polynomials in the spectral sum, thus reducing the amplitude of the high frequency modes. The filter is applied here

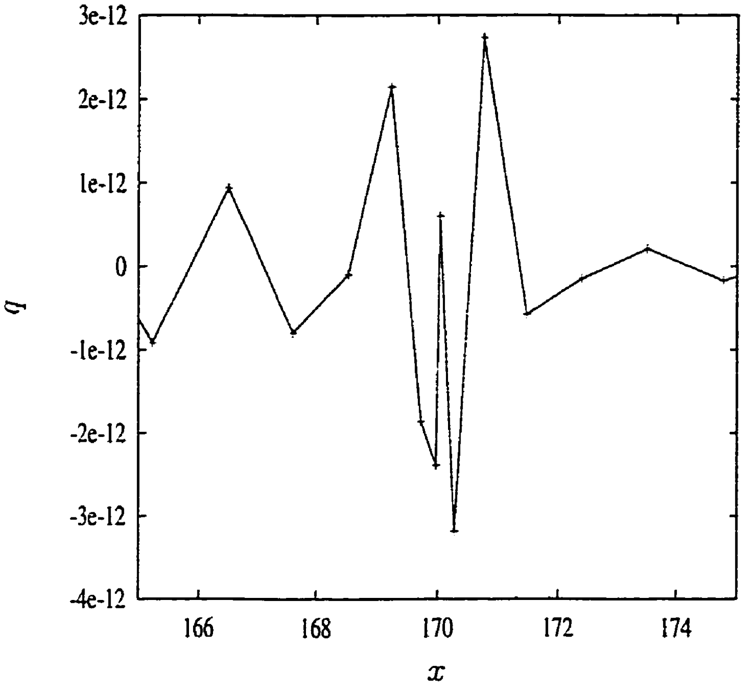


Figure 4.1: Spurious oscillations generated near element boundaries. The exact solution is virtually zero in this region. The inter-element boundary is at $x = 170$.

before differentiation, such that equation (2.26) is replaced by

$$\mathbf{F'}_{\mathbb{G}} = \mathbf{D} \cdot (\mathbf{S}\mathbf{F}_{\mathbb{L}}) = \mathbf{D} \cdot \mathbf{F}^{s}_{\mathbb{L}} \tag{4.9}$$

where the smoothing matrix $\mathbf{S}$ represents the action of the filter, and $\mathbf{F}^{s}_{\mathbb{L}}$ the vector of filtered flux values at the Lobatto points.

The matrix $\mathbf{S}$ is constructed using the Chebyshev spectral sum for the flux values, based on the set of Lobatto points along a $\xi$-line. Omitting constant $\xi$

components, the interpolant of the flux can be written in two alternative ways,

$$\tilde{F}(\xi) = \sum_{n=0}^{N} \tilde{F}(\xi^n) h_n(\xi) = \sum_{n=0}^{N} \hat{f}_n T_n(\xi) \tag{4.10}$$

where the Chebyshev coefficients are given by

$$\hat{f}_n = \frac{2}{NC_n} \sum_{l=0}^{N} \tilde{F}(\xi^l) \frac{T_n(\xi^l)}{C_l} \tag{4.11}$$

the value of the constants $C_l$ being the same as in section 2.6. A $o > 1$ order filter function $\varrho(x)$ can be defined following Don and Gottlieb [66] as a $C^{\infty}[0,1]$ function satisfying

$$\begin{aligned} &\varrho(0) = 1, && \varrho(1) = 0 && \\ &\varrho^{(j)}(0) = 0, && \varrho^{(j)}(1) = 0, && j \leq o \end{aligned} \tag{4.12}$$

$\varrho^{(j)}$ denoting here the $j$-th derivative. The interpolant in equation (4.10) is replaced by the filtered interpolant

$$\tilde{F}_s(\xi) = \sum_{n=0}^{N} \varrho\left(\frac{n}{N}\right) \hat{f}_n T_n(\xi) \tag{4.13}$$

which, replacing the Chebyshev coefficients by their values in equation (4.11), becomes:

$$\tilde{F}_s(\xi) = \sum_{n=0}^{N} \frac{2}{NC_n} \varrho\left(\frac{n}{N}\right) T_n(\xi) \sum_{l=0}^{N} \tilde{F}(\xi^l) \frac{T_n(\xi^l)}{C_l}. \tag{4.14}$$

Rearranging and interchanging the order of summation, the filtered interpolant can be written as

$$\tilde{F}_s(\xi) = \sum_{l=0}^{N} \left( \frac{2}{NC_l} \sum_{n=0}^{N} \varrho\left(\frac{n}{N}\right) \frac{T_n(\xi^l)T_n(\xi)}{C_n} \right) \tilde{F}(\xi^l) \tag{4.15}$$

or, in matrix form

$$\mathbf{F}^{s}_{\mathbb{L}} = \mathbf{S}\mathbf{F}_{\mathbb{L}} \tag{4.16}$$

where the entries of the smoothing matrix $\mathbf{S}$ are given by

$$\mathbf{S}_{il} = \frac{2}{NC_l} \sum_{n=0}^{N} \varrho\left(\frac{n}{N}\right) \frac{T_n(\xi^l)T_n(\xi^i)}{C_n}, \qquad i,l = 0,\ldots,N\,. \tag{4.17}$$

Although several types of filters have been implemented in the algorithm, the results have been obtained using either the sharpened raised cosine filter or the exponential filter. The sharpened raised cosine filter, which is eighth order accurate, is given by [29]:

$$\varrho(x) = \varrho_0^4(35 - 84\varrho_0 + 70\varrho_0^2 - 20\varrho_0^3); \;\; \varrho_0 = 0.5(1 + \cos(\pi x)) \tag{4.18}$$

The exponential filter offers the advantage that it can keep the spectral accuracy of the method. Such a filter can be defined as

$$\varrho\left(\frac{n}{N}\right) = \begin{cases} 1, & n \leq n_c \\ \exp\left( -\mathrm{In}(\varepsilon) \left( \frac{n - n_c}{N - n_c} \right)^{o_f} \right), & n > n_c \end{cases} \tag{4.19}$$

where $o_f$ is the order of the filter, $\varepsilon$ is the machine precision (around le-16 on the computers that have been used) and $n_c$ is the filter cut-off frequency. Increasing $o_f$ with $N$ keeps the method exponentially accurate. In practice, both $n_c$ and $o_f$

have been chosen around $N - 3$.

The filtering process does not necessarily increase the cost of the computation. Indeed, once the type of the filter is chosen, the smoothing matrix S can be constructed in the same time with the differentiation matrix $\mathbf{D}$, and equation (4.9) can actually be implemented as

$$\mathbf{F}'_{\mathbb{G}} = (\mathbf{DS}) \cdot \mathbf{F}_{\mathbb{L}} = \mathbf{D}^{s} \cdot \mathbf{F}_{\mathbb{L}} \tag{4.20}$$

where the product matrix $\mathbf{D}^{s}$ incorporates the effect of the filter and can be stored in place of, and used in the same way as, the differentiation matrix $\mathbf{D}$.

# Chapter 5

# Mapping techniques

## 5.1 Grid generation

### 5.1.1 Introduction

To define the spectral approximation on the generic element $D$, the mapping $\vec{\xi}(\vec{x})$ and/or its inverse must be defined. There are several criteria that this mapping must satisfy. First, it must keep the spectral convergence properties of the spatial discretization. In other words, the geometry of the domain must also be approximated with spectral accuracy when the boundaries are curved surfaces. This is particularly important for the present method, since the elements can be much larger then those used for a finite element method say, and boundary approximation errors can become in this case important, as has been demonstrated numerically by Atkins [67]. Second, the map must be simple enough and cost effective, such that it can be constructed in only a fraction of the time needed by the solver itself. Third, it is useful if the mapping for any number of Gauss points $N$ can be automatically generated by the algorithm itself using only the underlying element mesh. The element mesh can then be read in typical finite element format, which specifies the total number of 'nodes' (in this case the element corners), the total number of elements, and the connectivity table that associates to each element its respective corners (four for $d = 2$, eight for $d = 3$) in a prescribed order. The element mesh does not need to have any structure, any number of elements being allowed to join together at a corner. This allows the use of a single element mesh to be used to study several different frequencies, since the proper number of grid points per wavelength can be obtained by changing only $N$ in the data file.

To obtain such a mapping, it is reasonable to consider that the element mesh is such that element edges are smooth curves, and element faces are smooth surfaces. Then, the Chebyshev spectral approximation developed in chapter 2, which satisfies the above criteria, can be used to obtain interpolants of these manifolds as will be described hereafter. A complete isoparametric map within the element can then be obtained from these interpolants using Boolean sums [68]. It specifies the position of the state vector and flux definition points inside and on the boundaries of the element, as well as the metrics terms at these points. The main aspects of this process are described below.

### 5.1.2 Edge grids

To keep the spectral accuracy of the method, the geometry of the edge is supposed to be known exactly. In this case, it can be specified in the usual parametric form

$$\mathcal{E} = \left\{ \vec{x} \in \mathbb{R}^d \middle| \vec{x} = \vec{x}_{\mathcal{E}}(\tau), \tau \in [0,1] \right\} \tag{5.1}$$

and its length is given by

$$\mathcal{L} = \int_0^1 \sqrt{\sum_{i=1}^{d} x_{i,\tau}^2}\, d\tau. \tag{5.2}$$

Noting that the interval $[-1, 1]$ can be transformed into the interval $[0, \mathcal{L}]$ by the linear application $\xi = 2\ell/\mathcal{L} - 1$, where $\ell$ is the arc length along the edge, a one-to-one mapping from the unit interval onto the edge can be defined as

$$\mathcal{M}_{\mathcal{E}} : [-1, 1] \to \mathcal{E}, \qquad \mathcal{M}_{\mathcal{E}}(\xi) = \vec{x}_{\mathcal{E}}\left(\frac{\xi + 1}{2}\right) \tag{5.3}$$

and the image of the Lobatto points set $\mathbb{L}$ on the edge can be found as

$$\mathbb{L}_{\mathcal{E}} = \left\{ \vec{x} \in \mathbb{R}^d | \vec{x} = \vec{x}_{\mathcal{E}}\left(\frac{\xi^i + 1}{2}\right), i = 0, \ldots, N \right\}. \tag{5.4}$$

To have a sufficiently general algorithm, these points are used to define the $N$-th degree Chebyshev interpolant of the edge as

$$\vec{x}_{\mathcal{E}}^{(N)} : [-1, 1] \to \mathcal{E}, \qquad \vec{x}_{\mathcal{E}}^{(N)}(\xi) = \sum_{n=0}^{N} \vec{x}_{\mathcal{E}}\left(\frac{\xi^n + 1}{2}\right) h_n(\xi) \tag{5.5}$$

It is this interpolant that is actually used in the computations, and not the mapping in equation (5.3). The reason for this is twofold. First, there may be instances when the edge is not known in the parametric form (5.1). Instead, and this is the case in usual engineering applications, a set of points on the edge are usually given in the form $(x_1, x_2(x_1), x_3(x_1))$. Here $x_1$ can be, by extension, any of the three space coordinates, and it is supposed that $x_1 \in [-1, 1]$ which is always possible by a linear transformation as above. Then, the parametric mapping in equation (5.3) is replaced by a spline interpolant $\vec{x}_{\mathcal{E}}^{(s)}$, from which the Chebyshev interpolant follows from equation (5.5). Second, the Chebyshev interpolant can be differentiated easily to obtain the needed metrics, using the formulas developed in chapter 2.

Several parametric forms have been implemented in the algorithm to provide edge descriptions. For both $d = 2$ and $d = 3$, straight lines and circles can be defined, in the first case by specifying the end points (which are also the edge corners) and in the second by specifying also the position of the center. A general parametric form can also be specified for the case $d = 2$, as a functional expression of the parameter $\tau$, which is parsed using a parser developed by Prof. D.A. Kopriva at Florida State University and translated into the corresponding computer instructions. When only discrete point sets are known on the edges, splines under tension are the last alternative provided for $d = 2$. Obviously, in this case, the mapping is not truly isoparametric, and spectral accuracy can no longer be obtained due to the lack of physical information. It should be also noticed that for $d = 1$ the edge is actually the element itself.

### 5.1.3 Face grids

Following Gordon and Hall [68], the edge Chebyshev interpolants are used to build a multidimensional surface interpolant for the face. This face interpolant is required to coincide with the edge interpolants along face boundaries. Consider that the two coordinates that vary along the face are $\xi$ and $\eta$, such that its bounding edges are $\mathcal{E}_1(\xi = \xi^0 = -1)$, $\mathcal{E}_2(\xi = \xi^N = 1)$, $\mathcal{E}_3(\eta = \eta^0 = -1)$ and $\mathcal{E}_4(\eta = \eta^N = 1)$. The edge interpolants $\vec{x}_{\mathcal{E}_i}^{(N)}$, $i = 1, \ldots, 4$, are then obtained as described in the previous section.

A continuous mapping $\vec{x}_\xi(\vec{\xi})$ that coincides with the end edges $\mathcal{E}_1$ and $\mathcal{E}_2$ for $\xi = \xi^0$ and $\xi = \xi^N$, respectively, can be defined by interpolating the edge mappings in the $\xi$ direction

$$\vec{x}_\xi(\xi,\eta) = \Lambda_0(\xi)\vec{x}^{(N)}_{\mathcal{E}_1}(\eta) + \Lambda_N(\xi)\vec{x}^{(N)}_{\mathcal{E}_2}(\eta) \tag{5.6}$$

where the $\Lambda$'s are two univariate functions such that $\Lambda_i(\xi^j) = \delta_{ij}$. In this thesis, they have been chosen such that the interpolant is linear,

$$\Lambda_0(\xi) = \frac{1-\xi}{2}\ ;\ \ \Lambda_N(\xi) = \frac{1+\xi}{2}\ . \tag{5.7}$$

This mapping will not coincide, however, with the edge mappings on $\mathcal{E}_3$ and $\mathcal{E}_4$. A similar interpolant in the $\eta$-direction can be used to match these two edges,

$$\vec{x}_\eta(\xi,\eta) = \Theta_0(\eta)\vec{x}^{(N)}_{\mathcal{E}_3}(\xi) + \Theta_N(\eta)\vec{x}^{(N)}_{\mathcal{E}_4}(\xi) \tag{5.8}$$

where

$$\Theta_0(\eta) = \frac{1-\eta}{2}\ ;\ \ \Theta_N(\eta) = \frac{1+\eta}{2}\ . \tag{5.9}$$

A product of the two mappings can be constructed that matches the edge mappings only at the four corners of the face $(\xi^0,\eta^0)$, $(\xi^0,\eta^N)$, $(\xi^N,\eta^0)$ and $(\xi^N,\eta^N)$ by:

$$\vec{x}_{\xi\eta}(\xi,\eta) = \vec{x}_\xi(\xi,\eta)\cdot\vec{x}_\eta(\xi,\eta) \tag{5.10}$$

To match the edge mappings along all the face boundary, the face interpolant is then defined as the Boolean sum

$$\vec{x}_{\mathcal{F}}(\xi,\eta) = \vec{x}_{\xi}(\xi,\eta) \oplus \vec{x}_{\eta}(\xi,\eta) = \vec{x}_{\xi}(\xi,\eta) + \vec{x}_{\eta}(\xi,\eta) - \vec{x}_{\xi\eta}(\xi,\eta) \tag{5.11}$$

This is the basic construction for transfinite interpolation [68]. Following Fletcher, equation (5.11) is implemented in two stages. In the first stage, the $\xi$ interpolation in equation (5.6) is performed. In the second stage, the face interpolant is constructed as

$$\begin{aligned} \vec{x}_{\mathcal{F}}(\xi,\eta) = \vec{x}_{\xi}(\xi,\eta) \quad &+ \quad \Theta_0(\eta)\left[\vec{x}_{\mathcal{E}_3}^{(N)}(\xi) - \vec{x}_{\xi}(\xi,\eta^0)\right] \\ &+ \quad \Theta_N(\eta)\left[\vec{x}_{\mathcal{E}_4}^{(N)}(\xi) - \vec{x}_{\xi}(\xi,\eta^N)\right]. \end{aligned} \tag{5.12}$$

To compute grid metrics, or equivalently the face normals, it is again useful to express $\vec{x}_{\mathcal{F}}(\xi,\eta)$ using the Lagrange interpolants. Based upon the tensor product of Lobatto points $\vec{x}_{\mathcal{F}}^{ij} = \vec{x}_{\mathcal{F}}(\xi^i,\eta^j)$, the numerical form of the interpolant actually used to define the face geometry is:

$$\vec{x}_{\mathcal{F}}^{(N)}(\xi,\eta) = \sum_{i=0}^{N}\sum_{j=0}^{N}\vec{x}_{\mathcal{F}}^{ij} h_i(\xi) h_j(\eta). \tag{5.13}$$

For most situations encountered in practice, computing the position of the face Lobatto points from the edge mappings using transfinite interpolation as above is appropriate. In situations when this is not the case (as for example when the face, although smooth, contains local humps well within its boundaries) the algorithm provides the possibility that the complete set of Lobatto points $\vec{x}_{\mathcal{F}}^{ij}$ be specified in a data file, from which the face interpolant (5.13) is obtained directly.

### 5.1.4 Domain grids

Domain mappings can be constructed from face mappings using three dimensional transfinite interpolation. Consider the generic element $D$ with faces $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_6$ corresponding to $\xi = \xi^0, \xi = \xi^N, \ldots, \zeta = \zeta^N$, respectively. Taking into account the respective local coordinates, the numerical face mappings are $\vec{x}^{(N)}_{\mathcal{F}_1}(\eta, \zeta)$, $\vec{x}^{(N)}_{\mathcal{F}_2}(\eta, \zeta)$, $\vec{x}^{(N)}_{\mathcal{F}_3}(\xi, \zeta)$, $\vec{x}^{(N)}_{\mathcal{F}_4}(\xi, \zeta)$, $\vec{x}^{(N)}_{\mathcal{F}_5}(\xi, \eta)$ and $\vec{x}^{(N)}_{\mathcal{F}_6}(\xi, \eta)$. The natural extension of equation (5.11) to three dimensions is then [68] the Boolean sum

$$\begin{aligned}
\vec{x}_{\mathcal{D}}(\xi,\eta,\zeta) &= \vec{x}_\xi(\xi,\eta,\zeta) \oplus \vec{x}_\eta(\xi,\eta,\zeta) \oplus \vec{x}_\zeta(\xi,\eta,\zeta) \\
&= \vec{x}_\xi(\xi,\eta,\zeta) + \vec{x}_\eta(\xi,\eta,\zeta) + \vec{x}_\zeta(\xi,\eta,\zeta) - \vec{x}_{\xi\eta}(\xi,\eta,\zeta) \qquad (5.14) \\
&- \vec{x}_{\eta\zeta}(\xi,\eta,\zeta) - \vec{x}_{\zeta\xi}(\xi,\eta,\zeta) + \vec{x}_{\xi\eta\zeta}(\xi,\eta,\zeta)
\end{aligned}$$

The corresponding uni-dimensional interpolants for the faces (face projectors) are

$$\begin{aligned}
\vec{x}_\xi(\xi,\eta,\zeta) &= \Lambda_0(\xi)\vec{x}^{(N)}_{\mathcal{F}_1}(\eta,\zeta) + \Lambda_N(\xi)\vec{x}^{(N)}_{\mathcal{F}_2}(\eta,\zeta) \\
\vec{x}_\eta(\xi,\eta,\zeta) &= \Theta_0(\eta)\vec{x}^{(N)}_{\mathcal{F}_3}(\xi,\zeta) + \Theta_N(\eta)\vec{x}^{(N)}_{\mathcal{F}_4}(\xi,\zeta) \qquad (5.15) \\
\vec{x}_\zeta(\xi,\eta,\zeta) &= \Xi_0(\zeta)\vec{x}^{(N)}_{\mathcal{F}_5}(\xi,\eta) + \Xi_N(\zeta)\vec{x}^{(N)}_{\mathcal{F}_6}(\xi,\eta)
\end{aligned}$$

where, similar to $\Lambda$ and $\Theta$,

$$\Xi_0(\zeta) = \frac{1-\zeta}{2}\,; \quad \Xi_N(\zeta) = \frac{1+\zeta}{2}\,. \qquad (5.16)$$

The reason for subtracting the edge interpolants $\vec{x}_{\xi\eta}(\xi,\eta,\zeta)$, etc., from the Boolean sum is that each face interpolant will bring its own image of its bounding edges. Therefore, two edge images will otherwise appear in equation (5.14) from the two faces that have the edge in common. A similar reasoning applies for $\vec{x}_{\xi\eta\zeta}(\xi,\eta,\zeta)$

which interpolates the domain corners.

Equation (5.14) is implemented in three steps. In the first step, the interpolant $\vec{x}_1(\xi,\eta,\zeta)=\vec{x}_\xi(\xi,\eta,\zeta)$ is constructed. In the second step, this interpolant is used to define

$$\begin{aligned}\vec{x}_2(\xi,\eta,\zeta)=\vec{x}_1(\xi,\eta,\zeta) \;&+\; \Theta_0(\eta)\left[\vec{x}^{(N)}_{\mathcal{F}_3}(\xi,\zeta)-\vec{x}_1(\xi,\eta^0,\zeta)\right]\\ &+\; \Theta_N(\eta)\left[\vec{x}^{(N)}_{\mathcal{F}_4}(\xi,\zeta)-\vec{x}_1(\xi,\eta^N,\zeta)\right]\end{aligned} \tag{5.17}$$

in a similar way to equation (5.12). The last step is to obtain the domain mapping as

$$\begin{aligned}\vec{x}_{\mathcal{D}}(\xi,\eta,\zeta)=\vec{x}_2(\xi,\eta,\zeta) \;&+\; \Xi_0(\eta)\left[\vec{x}^{(N)}_{\mathcal{F}_5}(\xi,\eta)-\vec{x}_2(\xi,\eta,\zeta^0)\right]\\ &+\; \Xi_N(\eta)\left[\vec{x}^{(N)}_{\mathcal{F}_6}(\xi,\eta)-\vec{x}_2(\xi,\eta,\zeta^N)\right]\;.\end{aligned} \tag{5.18}$$

From this mapping, the Lobatto point locations within the element $D$ can be computed as the image $\vec{x}^{ijn}_{\mathcal{D}}=\vec{x}_{\mathcal{D}}(\xi^i,\eta^j,\zeta^n)$. The corresponding numerical mapping constructed with this set of points,

$$\vec{x}^{(N)}_{\mathcal{D}}(\xi,\eta,\zeta)=\sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{n=0}^{N}\vec{x}^{ijn}_{\mathcal{D}}h_i(\xi)h_j(\eta)h_n(\zeta) \tag{5.19}$$

is used to compute the derivatives needed in the computation of the first Jacobian (2.8), as well as to obtain the position of the set of state vector collocation points within an element.

## 5.2 Increasing the time step by mapping

### 5.2.1 New interpolation and differentiation operators

The differentiation matrix **D** has eigenvalues that grow like $\mathcal{O}(N^2)$, being proportional to the reciprocal of the minimum grid spacing, as discussed in section 3.7. The quadratic clustering of the grid points near the end of the interval is required to avoid the oscillations that occur when high degree polynomials are used to approximate smooth functions (the Runge phenomenon). However, the time step size that also decreases quadratically with $N$ might be smaller than the one dictated by the physics of the problem. To increase the minimum spacing, and hence the time step, one possibility is to work with non-polynomial bases. Such a method has been developed by Kosloff and Tal-Ezer [70]. They introduced a mapping of the interval $[-1, 1]$ on itself that tends to equi-distribute the Lobatto points within the interval. Derivatives are computed in transformed space using standard Chebyshev polynomials, which is equivalent to using a non-polynomial basis in the physical space. The grid size in physical space, and hence the time step, decrease however as $\mathcal{O}(N)$. The Kosloff–Tal-Ezer mapping has been used in the context of the staggered grid approximation by Bismuti and Kopriva [71]. They perform the transformation on the governing equations themselves. A distinct possibility, easier to implement in the algorithm since it only requires the modification of the differentiation matrix **D** and the interpolation matrix **I**, which are computed only once, is described below.

The Kosloff–Tal-Ezer transformation from $\xi \in [-1, 1]$ to $\chi$-space ($\chi \in [-1, 1]$), where the derivatives are actually computed in this case, is given by

$$\xi = \frac{\arcsin(\varpi\chi)}{\arcsin(\varpi)} = g(\chi;\varpi) \tag{5.20}$$

with $\varpi$ a parameter that will be discussed subsequently. For a given function $f$, the chain rule of differentiation gives

$$\frac{df}{d\xi} = \frac{d\chi}{d\xi}\frac{df}{d\chi} = \frac{1}{g'}\frac{df}{d\chi} \tag{5.21}$$

If the Lobatto points are now chosen in $\chi$-space as

$$\chi^j = -\cos\left(\frac{\pi j}{N}\right) \ , \ \ j = 0,\ldots,N \tag{5.22}$$

the position of the corresponding images $\xi^j$ will depend on the value of the parameter $\varpi$. The two limit cases are $\varpi = 1$, which corresponds to equi-distant nodes as for Fourier methods, and $\varpi = 0$, in which case $\xi^j$ will be the Lobatto points. Hence, a value of $\varpi$ larger than zero will bring about an increase of the minimal spacing in $\xi$-space, and consequently a larger time step. In this section, $\xi^j$ will be used henceforth to denote the position of the points obtained from $\chi^j$ for a fixed value of $\varpi$.

For the staggered grid approximation, the initial data are the values of the state vector at the set of Gauss-Chebyshev points $\bar{\xi}^j$. The first step of the spatial discretization is to interpolate the data to the points $\xi^j$. To accomplish this, the interpolation matrix **I** in equation (2.23) must be modified to reflect the new position of these points, its entries being given now by

$$\begin{aligned}\mathbf{I}_{ni} &= \bar{h}_i(\xi^n) = \frac{2}{N}\sum_{l=0}^{N-1}\frac{1}{C_l}\,T_l(\bar{\xi}^i)\,T_l(\xi^n) \\ &= \frac{2}{N}\sum_{l=0}^{N-1}\frac{1}{C_l}\cos\left(l\pi\left(1-\frac{2i-1}{2N}\right)\right)\cos\left(l\arccos(\xi^n)\right)\end{aligned} \tag{5.23}$$

where $n = 0,\ldots,N$ and $i = 1,\ldots,N$.

For differentiation, values of the derivatives at the points $\bar{\xi}^j$ are needed. They must be computed using the function values at the points $\xi^j$. When considered in the $\chi$-space, derivatives based on function values at the points $\chi^j$ are needed at the image of the Gauss-Chebyshev points $\bar{\xi}^j$. These images are given by

$$\bar{\chi}^j = \frac{\sin\left(\bar{\xi}^j\arcsin(\varpi)\right)}{\varpi} \tag{5.24}$$

Supposing now that the differentiation matrix in $\chi$-space is $\mathbf{D}^\chi$, such that the derivative of a generic function can be written as

$$\left.\frac{df}{d\chi}\right|_{\bar{\chi}^i} = \sum_{n=0}^{N}\mathbf{D}^\chi_{in}f(\chi^n)\,, \tag{5.25}$$

then the needed derivative, taking into consideration equation (5.21), is given by

$$\left.\frac{df}{d\xi}\right|_{\bar{\xi}^i} = \frac{\arcsin(\varpi)\sqrt{1-(\varpi\bar{\xi}^i)^2}}{\varpi}\sum_{n=0}^{N}\mathbf{D}^\chi_{in}f\left(\xi(\chi^n)\right)\,, \qquad i = 1,\ldots,N \tag{5.26}$$

The differentiation matrix $\mathbf{D}^\chi$ can be constructed similar to matrix $\mathbf{D}$ in chapter 2. The above relation then shows how it must be modified to act directly on the values at $\xi^n$ and thus replace matrix $\mathbf{D}$ in equation (2.26).

Working with a non-polynomial basis in $\xi$ brings along a decrease in accuracy. Kosloff and Tal-Ezer [70] show that this is however compensated by an increase in the resolution of the high modes. This may be beneficial for wave propagation problems because it reduces the requirements of number of points per wavelength, which decrease from $\pi$ points per wavelength in the case $\varpi = 0$ (Chebyshev approximation) to two points per wavelength when $\varpi = 1$ (Fourier approximation). A way to choose the value of $\varpi$ such that the loss in accuracy inherent in the transformation is of the order of the machine precision is also presented in [70]. This corresponds to setting

$$\varpi = \frac{2}{\varepsilon^{1/N} + \varepsilon^{-1/N}} \tag{5.27}$$

and is the form that has been implemented in the algorithm. In this formula, $\varepsilon$ can be chosen to be the machine precision when there are sufficient points per wavelength to resolve the highest modes with the Chebyshev approximation. This can still lead to an increase in the time step for large enough values of $N$. As an alternative, when this proves too expensive, $\varepsilon$ can be given a slightly higher value, such that the transformation reduces the number of necessary grid points.

### 5.2.2 Numerical experiments

To test the performance of the staggered grid spectral method with the use of the Kosloff–Tal-Ezer transformation, the linear convection equation (3.12) is solved in the domain $x \in [-50, 110]$ with several values of $\varepsilon$ in (5.27). Two separate discretizations have been considered. In the first case, the domain is divided into $E = 8$ elements, with $N = 20$ Gauss-Chebyshev points per element. In the second case, $E = 16$ and $N = 10$. Integration in time is performed up to $t = 60$. Results for the accuracy and the relative computer time (with the Chebyshev case taken as reference) in the two cases are given in tables 5.1 and 5.2.

| $\varepsilon$ | $\varpi$ | $L_\infty$ | CPU time |
|---|---|---|---|
| Chebyshev | 0.000 | 3.46e-4 | 1.00 |
| 1.e-16 | 0.309 | 4.48e-4 | 0.97 |
| 1.e-12 | 0.472 | 4.29e-4 | 0.92 |
| 1.e-10 | 0.575 | 5.10e-4 | 0.87 |
| 1.e-8 | 0.687 | 1.34e-3 | 0.81 |

Table 5.1: $L_\infty$ error norms and relative computer time for various mappings. Discretization: $E = 8$ and $N = 20$.

| $\varepsilon$ | $\varpi$ | $L_\infty$ | CPU time |
|---|---|---|---|
| Chebyshev | 0.000 | 2.14e-3 | 1.000 |
| 1.e-16 | 0.050 | 2.15e-3 | 0.999 |
| 1.e-12 | 0.126 | 2.16e-3 | 0.995 |
| 1.e-10 | 0.198 | 2.19e-3 | 0.987 |
| 1.e-8 | 0.309 | 2.28e-3 | 0.969 |

Table 5.2: $L_\infty$ error norms and relative computer time for various mappings. Discretization: $E = 16$ and $N = 10$.

It follows from these results that the method becomes more advantageous as the value of $N$ increases. For three dimensional computations, however, the savings in computer time can become important even for values around $N = 10$, which are expected to be commonly used.

# Chapter 6

# Numerical results

## 6.1 Subsonic flow in a 2D channel

The first computation performed with the algorithm was aimed at testing the grid generation routines and the convergence of the method when computing steady flows. To this end, the flow over a circular bump in a channel that has been also used in the paper that introduced the staggered grid approximation [33] was computed on two different grids. For both grids, the circular bump is described using a parameterization of the circle that is obtained within the algorithm from the given position of the center and the arc endpoints. For the first grid, all other edges are straight lines, while for the second grid several internal edges are specified using splines under tension. The inflow Mach number is $M_\infty = 0.3$, and the number of Gauss points per element has been varied between $N = 7$ and $N = 9$. Spectral accuracy with increasing $N$ has been reported for this case in [33]. In general, less accurate Euler solutions on this geometry fail to provide a symmetric solution over the bump [43], and a horseshoe vortex appears at the trailing edge due to the numerically generated entropy. The grids obtained with the transfinite interpolation routines for $N = 8$, together with the corresponding solutions, are shown in figures 6.1 and 6.2. The convergence history for $C = 2$ on the second grid is given in figure 6.3 for $N = 7$. As can be seen, the residual could be decreased by fourteen orders of magnitude, although the required number of time steps is relatively high. No technique for increasing the time step has been used in this case. No horseshoe vortex appears at the trailing edge, and to the resolution allowed by the plotting program (which considers variables varying only linearly between the grid points) the solution can be considered perfectly symmetric.

## 6.2 Acoustic pulse in a 2D quiescent medium

This type of problem has been proposed by Tam and Webb [21] as a test of the isotropic properties of numerical algorithms for computational aeroacoustics and of the boundary conditions. The pulse is initialized at $t = 0$ at the point $(x = 0, y = 0)$ in the computational domain $(x, y) \in [-50, 50]^2$, the coordinates being non-dimensionalized such that the width of the pulse is $\delta_p = 3$. The acoustic variables have a Gaussian distribution of the form:

$$\begin{bmatrix} p - \bar{p} \\ \rho - \bar{\rho} \\ v_1 - \bar{v}_1 \\ v_2 - \bar{v}_2 \end{bmatrix} = 0.01 \exp\left( - \ln(2) \frac{x^2 + y^2}{\delta_p^2} \right) \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{6.1}$$

There is no mean flow in the computational domain, such that $\bar{v}_1 = \bar{v}_2 = M_\infty = 0$. The pulse spreads like a cylindrical wave, propagating with the non-dimensional sound speed $c_\infty = 1$, such that at time $t = 47$ it starts impinging on the domain boundaries, and leaves the domain completely at approximately $t = 75$. The computation is accomplished on a grid made of 100 elements, with $N = 7$ Gauss points per element. Both the FWM and TW have been used as radiation boundary conditions. In this case, no filter was necessary for stability for the TW method.

Figure 6.4 shows the pulse immediately after its initialization, at time $t = 5$. A comparison of the computed acoustic pressure along the line $y = 7.169$ at time $t = 30$ with the exact solution obtained from the linearized Euler equations is given in figure 6.5. Figures 6.6 and 6.7 show the acoustic pulse at time $t = 55$, computed with the FWM and TW, respectively. In both these figures, 10 contours of the acoustic pressure in the interval $[-4 \cdot 10^{-4}, 8 \cdot 10^{-4}]$ are plotted. The circular form of the contours and the lack of oscillations near the boundaries are an indication that the boundary conditions allow the pulse to cross the boundaries without reflections. It can be noticed however that there are small deviations from the circular shape of the contours near the corners of the domain in the results obtained with TW. Although the reflections are very small, this is an evidence of the fact that the FWM is more adequate for use with the staggered grid method than TW.

## 6.3 Two-dimensional flat ducts

One of the most important requirements for a nonlinear acoustic solver is that it must reproduce correctly results obtained from the linearized version of the equations when the amplitude of the perturbations is sufficiently small. A very useful test problem, featuring all the difficulties encountered in the computation of fan noise radiation, is therefore the flat duct radiation problem. The flat duct is a semi-infinite duct with zero thickness, completely rigid, walls. For the cases considered here, there is no flow in the computational domain, $M_\infty = 0$. A duct acoustic mode propagates in the positive $x$ direction (to the right) from $x = -\infty$. Upon reaching the duct aperture, situated at $x = 0$, a part of the acoustic energy is reflected back within the duct, the rest radiating to free space in a way that depends on the incident mode and its driving frequency. The problem is represented schematically in figure 6.8.

Analytical results for the linearized problem have been obtained by Mani [14], who solved the Helmholtz equation using a Wiener-Hopf technique. To match the analytical results, the numerical algorithm must correctly reproduce the acoustic scattering from the aperture. Due to the finite size of the computational domain, the infinite extension of the duct to the left is modeled by the boundary condition at the source plane. The left-propagating waves scattered from the aperture must exit the domain at this location without reflection. Since this is only ensured by the Riemann problem solution, without other special treatment, the results also validate this approach. The radiation boundary conditions are also tested by comparing the directivity on the radiation boundary with the far-field analytical results.

The computational domain for this case is $(x, y) \in [-3, 5] \times [-5, 6]$, the reference length $L$ being the duct cross dimension. The duct walls are situated at $y = 0$ and $y = 1$. The amplitude of the incoming waves in equation (4.2) was specified as $\mathcal{A} = 10^{-4}$, and the non-dimensional driving frequency $\omega = 15$ is the same that has been considered by Dong *et* al. [72]. The computational domain is subdivided into 88 quadrilateral elements of size $1 \times 1$, with 13 Gauss-Chebyshev points within each element. This domain discretization gives an average of 5.4 points per wavelength, with a total of 14872 points within the domain.

Figure 6.9(a) shows the computational domain and the Gauss-Chebyshev points grids for the elements inside the duct. To achieve a periodic flow field, the solution has been marched for 2000 time steps with a four stages Runge-Kutta scheme. This required about 13 minutes on one R10000 CPU with 195 MHz clock speed of a Silicon Graphics Origin 2000 machine, which gives $24\mu s$ per grid point per time step. Since the domain boundary is not a circle, the two-dimensional spreading of the wave must be compensated for by multiplying the root mean square pressure $\mathcal{P}$ on the boundary of the domain with the factor $r^{1/2}$. Hence, the directivity pattern is obtained by normalizing $\mathcal{P} \times r^{1/2}$ such that the maximum corresponds to 100dB. The root mean square pressure is obtained as

$$\mathcal{P}^2 = \frac{1}{T}\int_t^{t+T} (p - \bar{p})^2 dt \qquad (6.2)$$

where $T = 2\pi/\omega$ is the period of the wave, the integral being approximated numerically. Figures 6.9(b)-(d) display directivity patterns thus obtained for the plane wave and the first two cross modes, compared to the analytical solutions by Mani [14]. The radiation boundary condition used to obtain these results is the FWM, with the location of the source specified at the middle of the duct aperture. As can be seen, the agreement is very good, although some small reflections can be observed at the left hand side corners of the domain. Figure 6.10 presents acoustic pressure contours for the first two cross modes.

The results obtained with characteristic boundary conditions for this problem, although meaningful, feature much larger reflections at the corners of the domain.

They are compared with those obtained with FWM in figure 6.11. The TW boundary condition was unstable unless a filter was used for the layer of elements next to the boundary. The results with filtering were in general slightly less accurate than those obtained with the FWM, as can be seen in figure 6.12. Both the sharpened raised cosine filter and the exponential filter (order ten) led to identical results. To demonstrate the efficiency of the absorbing layer when used with the FWM, directivity patterns for the first cross mode calculated with very small damping (damping parameters: $\sigma_M = 2.5$ and $\beta = 1$) and without damping are compared in figure 6.13. Practically no more reflections from the boundaries can be noticed, and the agreement with the exact solution becomes excellent. The effect of the absorption can be seen better on figure 6.14, where contour plots of the RMS pressure are drawn using the data at Gauss-Chebyshev points. The acoustic signal decreases rapidly in magnitude inside the damping layer, such that the amplitude of the waves that reach the boundary, as well as that of the eventual reflections, is negligible.

## 6.4 Axisymmetric flat ducts

The axisymmetric flat duct duct problem has been used in the second computational aeroacoustics workshop on benchmark problems [73] to test the capability of numerical algorithms to compute radiation from cylindrical ducts. The problem, similar to the 2D case, is however more difficult to solve numerically, in the sense that it requires larger computational domains to avoid boundary reflections when the FWM and TW are used as radiation boundary conditions. Analytical results also exist for this case, obtained by Savkar [15]. They have not been used for comparison in this thesis because some of the complex integrals involved are particularly difficult to compute and require special numerical treatment. Instead, analytical results for the directivity of a plane circular piston vibrating in the same radial mode as given by Tyler and Sofrin [6] are used for comparison with the computed directivity pattern, obtained in this case using $\mathcal{P} \times r$ on the boundary of the computational domain, due to the spherical spreading of the wave. This quantity is normalized to peak at 100dB. Since the results for the circular piston do not take into consideration the existence of the duct walls and the reflections from the duct aperture, agreement is expected to be less perfect, as can be noticed in figure 6.15 for the plane wave, mode (0,0), and the first radial mode (0,1). For this case, the duct radius is taken as the reference length, and the non-dimensional angular frequency is $\omega = 10.3$. The computational domain extends 9 radii from the duct exit in the axial direction and 9 radii from the axis in the radial direction, and is made up of 162 elements on which 10 Gauss-Chebyshev points have been used. Acoustic pressure contours for the two modes are shown in figures 6.16 and 6.17. The TW boundary condition has been used for these results, with the sharpened raised cosine filter in the boundary elements.

## 6.5 Fan flight inlet

To test the full capabilities of the $d = 2$ algorithm, the geometry of a turbofan inlet currently under production at Pratt & Whitney Canada has been considered. Computations have been performed only for axisymmetric modes, at the blade passage frequency (BPF), both with ($M_\infty = 0.2$) and without ($M_\infty = 0$) flow. The source plane within the inlet was located before the end of the center body, at a position where the inlet shroud radius is 13.7in (taken as the reference length for non-dimensionalization) and the hub-to-tip ratio is 0.423. The angular speed $\mathcal{N} = 12000$rpm leads to a BPF of 3800 Hz (the number of blades is $B = 19$) and a reduced frequency $\omega = 24.4$. The acoustic pressure amplitude at the source plane was set to $\mathcal{A} = 0.01$, which is 1.4 percent of the absolute pressure at infinity, $p_\infty = 1/\gamma$. The computation is therefore close to the limit of validity of linear results. The element mesh contains 358 nodes and 314 elements, and the number of Gauss points used was between $N = 10$ and $N = 13$. Based on the largest element size and taking the convection effect into account when there is a mean flow, this leads to an average number of points per wavelength varying between three and six. A view of the spectral mesh in the region of the inlet, obtained from a splines under tension description of the hub and casing, is provided in figure 6.18.

For the case without flow, the plane wave and the first three radial modes have been studied. To achieve a periodic solution, the solver has been marched in time until $t = 30$, which needed about 11,000 time steps. A snapshot of the acoustic pressure for the BPF(0,1) mode is displayed in figure 6.19. The interior limits of the damping layer are also displayed in this figure, such that the evanescent pressure amplitude can be noticed. A relatively large value for the damping parameter, $\sigma_M = 120$ has been used for the computations. The acoustic energy is mostly radiated sideways, the main lobe being directed away from the duct axis. Using a Wiener-Hopf technique, Rice [74] showed that, for cylindrical flat ducts, the angle of the far-field peak radiation, $\alpha_R$, depends mainly on the proximity of the mode to cut-off, and provided a formula to compute it. Denoting by $\varsigma = \omega/k_{m\mu}$ the cut-off ratio, the formula established by Rice is

$$\alpha_R = \arccos\left(\sqrt{1-\frac{1}{\varsigma^2}}\right). \tag{6.3}$$

Although the geometry of the inlet is significantly different from the ideal case considered by Rice, this formula was found to predict the main lateral lobe angle quite accurately. The result obtained from equation (6.3) is compared with the angle obtained from the numerical computation in table 6.1. The analytical result can be seen to constantly over-predict the numerical result by about one degree, an effect that can be anticipated because of the converging shape of the inlet towards the leading edge of the casing.

| Mode | (0,0) | (0,1) | (0,2) | (0,3) |
|---|---|---|---|---|
| Formula (6.3) | 0 | 11.9 | 23.6 | 37.0 |
| Present method | 0 | 10.7 | 22.6 | 35.8 |

Table 6.1: Main lateral lobe angles for BPF, $M_\infty = 0$.

A mesh refinement study has been performed for the BPF(0,1) mode by changing the number of Gauss-Chebyshev points, $N$. As the root mean square pressure $\mathcal{P}$ can be very small in certain regions of the computational domain, reaching values of the order of the truncation error of the computed value which is the total pressure $p$, an effect similar to that in figure 4.1 can be expected to appear at element interfaces. This is actually the case, as can be seen in figure 6.20. These discontinuities at interfaces decrease exponentially with $N$, but it can be appreciated that the results for the lowest value $N = 10$ already provide a correct picture of the radiated field. The convergence of the results with the number of mesh points can also be noticed in figure 6.21, which displays the corresponding far field directivity.

For the forward flight case, the mean flow was computed using an imposed outlet pressure at the source plane equal to $0.7p_\infty$. Only $N = 10$ points have been used for this computation. The solution was marched in time until the residual decreased fourteen orders of magnitude. The computed mean flow Mach contours in the inlet region are displayed in figure 6.22. This steady state, saved to a file, constitutes the initial condition for the computation of acoustic propagation. The same BPF(0,1) mode is used here to exemplify the effects of the non-uniform mean flow on the far-field radiation. The major radiation lobe shifts towards the axis for the forward flight case, for the BPF(0,1) mode the peak radiation taking place now exactly along the inlet axis. Acoustic pressure contours for this case are displayed in figure 6.23. The effect of the flow is clearly visible in this figure, the wavelength of the sound wave being shorter when it propagates against the flow (to the right) and larger when it propagates downstream with the flow. At infinity upstream, the ratio between the wavelength with and without flow is $1 - M_\infty = 0.8$, while at infinity downstream this ratio is $1 + M_\infty = 1.2$. The scale of wavelengths present in the flow-field is however larger, due to the higher Mach number (close to 0.4) in the region of the leading edge of the casing, such that for this number of Gauss points it can be expected that the wave-field is slightly under-resolved, and the inter-element discontinuities discussed previously are more obvious. This can be noticed in figure 6.24, where root mean square pressure contours on a logarithmic scale (proportional to the sound pressure level) are presented. Due to the logarithmic scale, the effect of the damping layer can be observed clearly. Inside this layer, largest reflections are seen to occur from the corners, but are many orders of magnitude lower in amplitude than the incident waves.

The obtained far-field directivity pattern is shown in figure 6.25 in comparison with the no flow case, both relative to the source amplitude. Qualitatively, the main effects of forward flight are the same as observed in experiments [75], namely lower sound pressure level for the flight case, and a shift of the main lobe angle towards the axis. A quantitative study implies a 3D solution because experimental results pertain to higher order modes, as for example BPF(13,0), which can not be modeled with the present method in a 2D setting.

## 6.6 Generic turboshaft engine inlet

This section presents results for the 3D propagation of spinning modes inside a simplified model of a helicopter turboshaft engine inlet, for which both experiments and computations using a boundary integral method have been carried on at ONERA [76]. The curved duct used to model the geometry is made up of three parts. Using the duct radius $R_0$ for non-dimensionalization, the source plane is located at the beginning of a straight circular duct 11.76 units long. The axis of symmetry of this cylinder is taken as the $x$ axis. The cylinder continues with a torus of internal radius 0.6 and external radius 2.6, and another straight duct 0.66 units long, from the exit plane of which sound is radiated into free space.

A frequency of 1000 Hz is considered, which corresponds to a non-dimensional angular frequency, $\omega R_0/c$, of 2.772. Both the propagation of mode (0,0) (plane waves) and mode(1,0) (first azimuthal mode) are analyzed. The mesh used for the computation is illustrated in figure 6.26. It is made up of 71 elements, with 25 elements within the pipe. The mesh within the elements is obtained using only a description of the edges in terms of circular arcs. The number of points per wavelength was varied between five and six ($N = 9$ to $N = 11$), without any noticeable changes in the results. Figure 6.27 shows snapshots of the acoustic pressure for the considered modes.

For this computation, characteristics boundary conditions have been used at radiation boundaries. The primary interest was in the RMS pressure distribution in the duct exit plane, contour plots of which are presented in figure 6.28 for the plane wave and figure 6.29 for the first azimuthal mode, respectively. For the plane wave case, the RMS pressure field is symmetric with respect to the $xz$ plane (the plane of symmetry of the duct), and there are only slight variations in sound pressure level on the duct exit plane. For the first azimuthal mode, the symmetry no longer exists, and the variations in RMS pressure on the duct exit plane are much larger. These results, as well as the shape of the RMS pressure contours, agree well qualitatively with those obtained by Malbéqui *et al* [76]. For a quantitative comparison, the root mean square pressure directivity data computed by Malbéqui *et al* using a boundary integral (BIM) method, as well as experimental data presented by the same authors have been digitized and are presented in figure 6.30 together with the results of the present computation. The latter have been computed on a sphere with radius $5R_0$, in the $x = 13.36$ plane (which goes through the center of the disc at the duct exit). As can be seen, the agreement is very good, especially when considering that the root mean square pressure is a much more sensitive quantity than the sound pressure level.

To a first approximation, the operation count is asymptotically proportional to $N^4$ in three dimensions, and varies linearly with the number of elements. However, this does not accurately account for the transmission of data between the element-indexed data structure and the face-indexed one. An estimate about the needed computational resources can be obtained using the CPU time per time step per grid point, which for example for the present grid with $N = 11$ Gauss points was 64.5 $\mu s$ on the same R10000/195MHz processor as above, when a five-stage Runge-Kutta scheme was used. The whole computation lasted about fifteen minutes on four processors on an Origin 2000 machine. The speed-up obtained by using several processors was, however, dependent on machine architecture, as can be noticed from figure 6.31. Speed-ups closer to the ideal could be obtained on bus architectures, such as the Power Challenge. On S2MP architectures, as the Origin 2000, the large multi-dimensional arrays used to store element values caused many cache misses, and the speed-up was visibly under the ideal value. When considering the results in figure 6.31, it must be, however, appreciated that the problem size is very small, and this is a primary cause why the performance deteriorates rapidly. For example, the loop over the acoustic source faces cannot be efficiently run in parallel since there are only five faces of this kind. Increasing the number of Gauss points $N$, as well as the size of the problem (number of elements and faces), should bring about a better parallel performance by allowing more work to be executed concurrently.

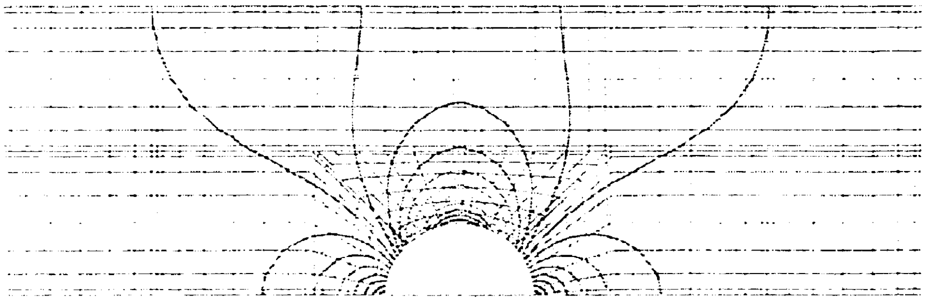


Figure 6.1: Mach number contours and first grid for the channel with a circular bump.

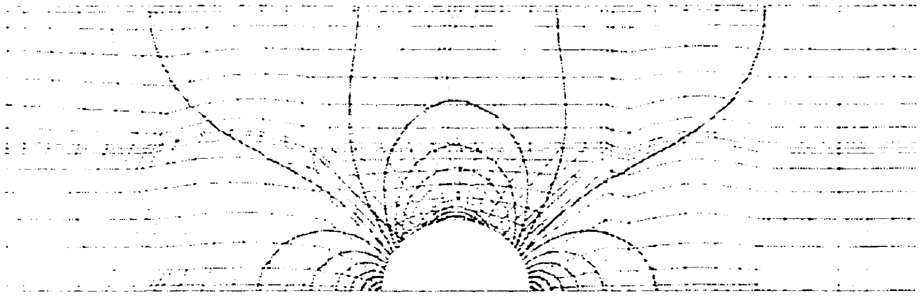


Figure 6.2: Mach number contours and second grid for the channel with a circular bump.

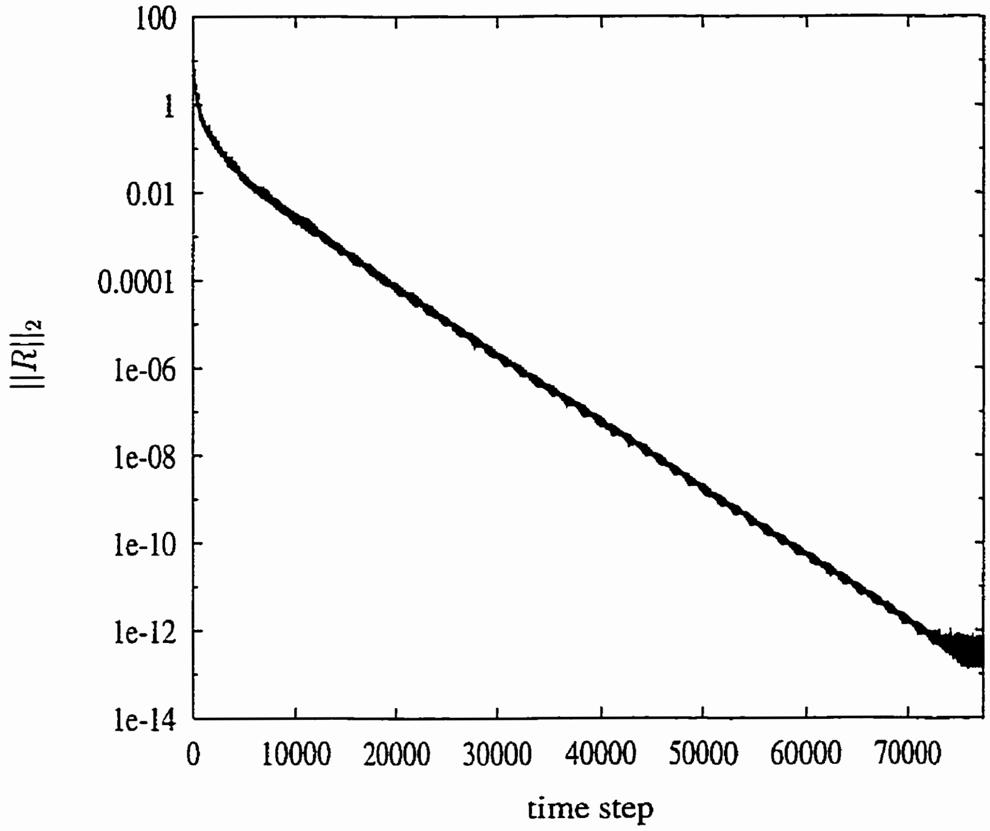


Figure 6.3: Convergence history for the second grid, $N = 7$.

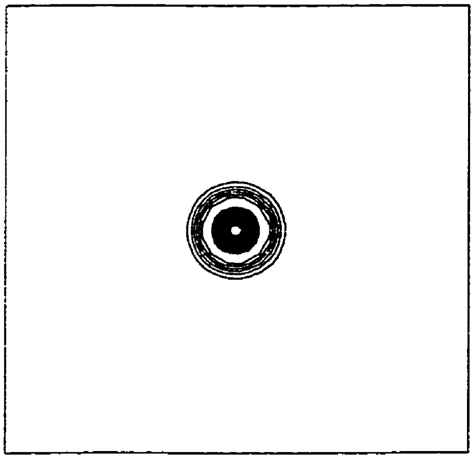


Figure 6.4: Pulse at time $t = 5$ (acoustic pressure contours).

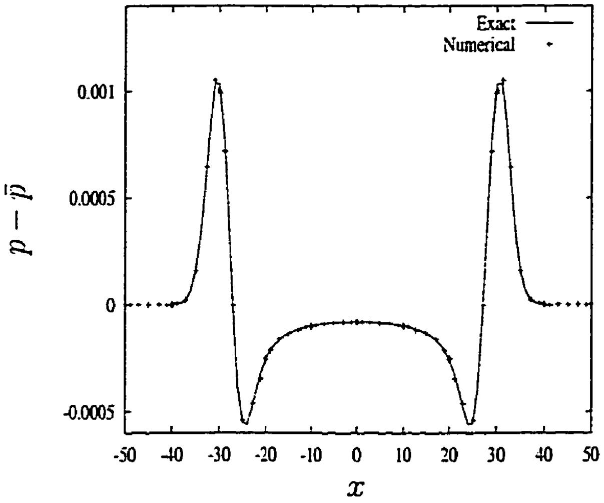


Figure 6.5: Comparison of the numerical solution for the propagation of the acoustic pulse at time $t = 30$ along the line $y = 7.169$ with the exact solution obtained from the linearized Euler equations.

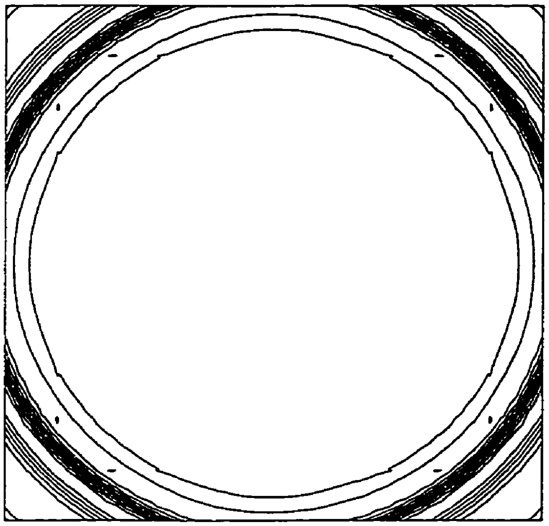


Figure 6.6: Acoustic pressure at time $t = 55$, computed with the FWM.

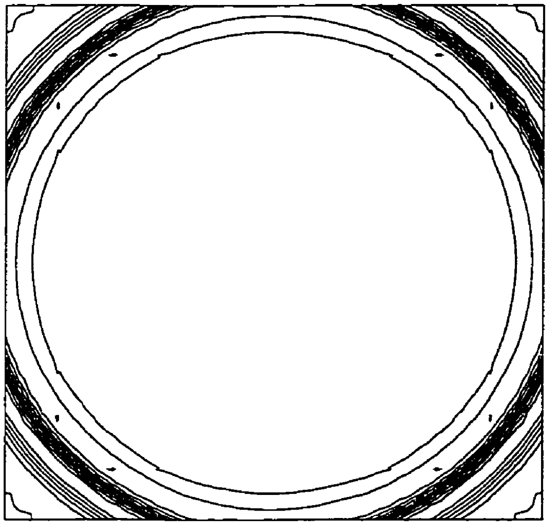


Figure 6.7: Acoustic pressure at time $t = 55$, computed with TW.

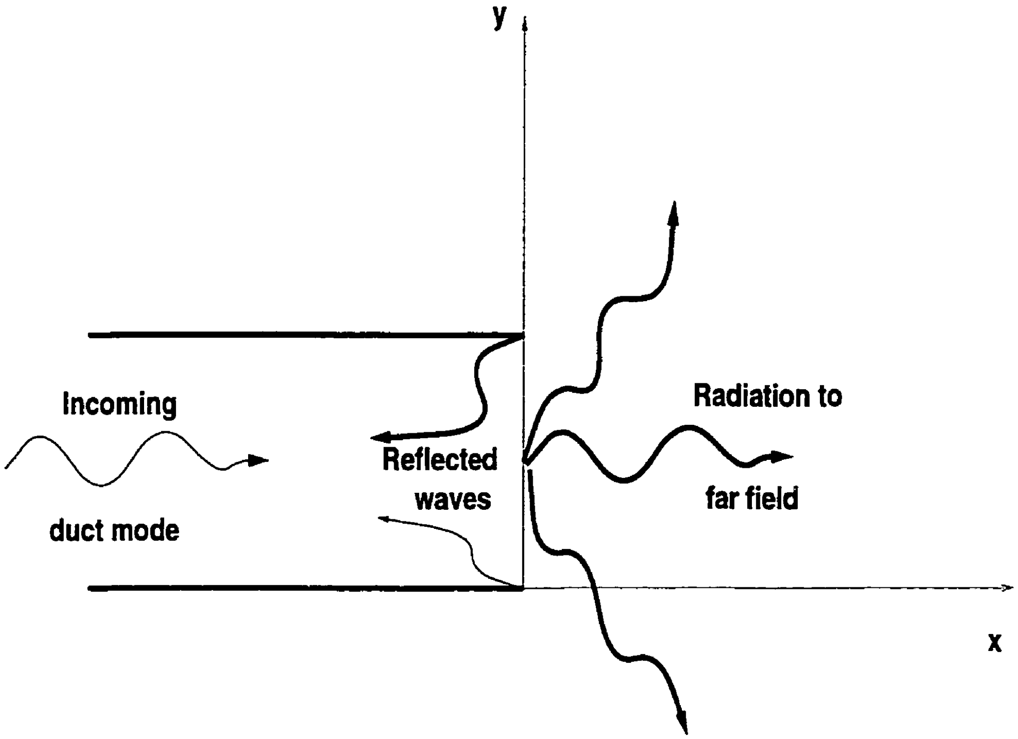
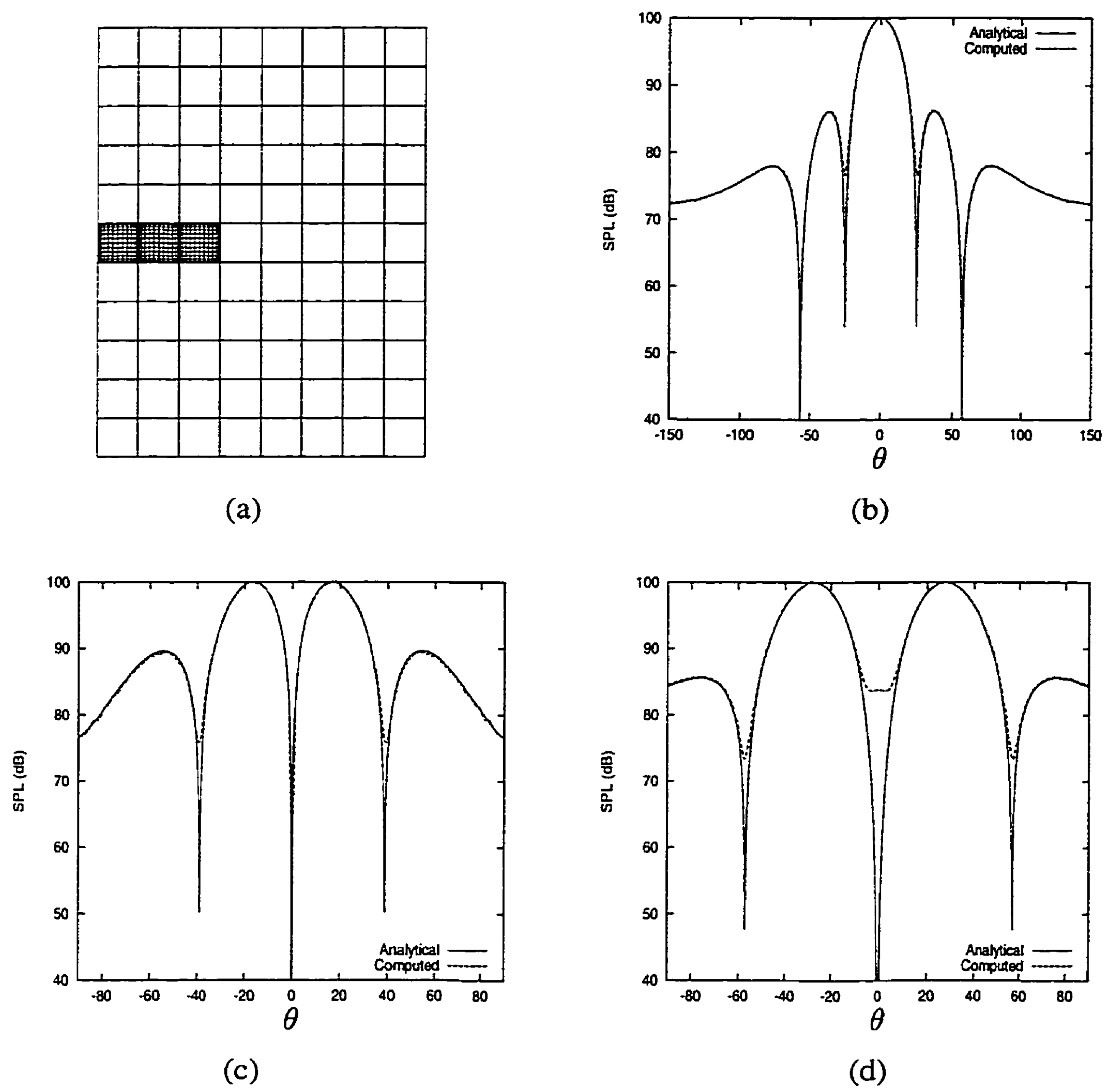


Figure 6.8: Schematic representation of the flat duct radiation problem.



(a)

(b)

(c)

(d)

Figure 6.9: Radiation from slab-symmetry flat duct, (a) Computational domain, grids shown in the domains inside the duct (b) Plane wave directivity pattern (c) First cross mode (d) Second cross mode.

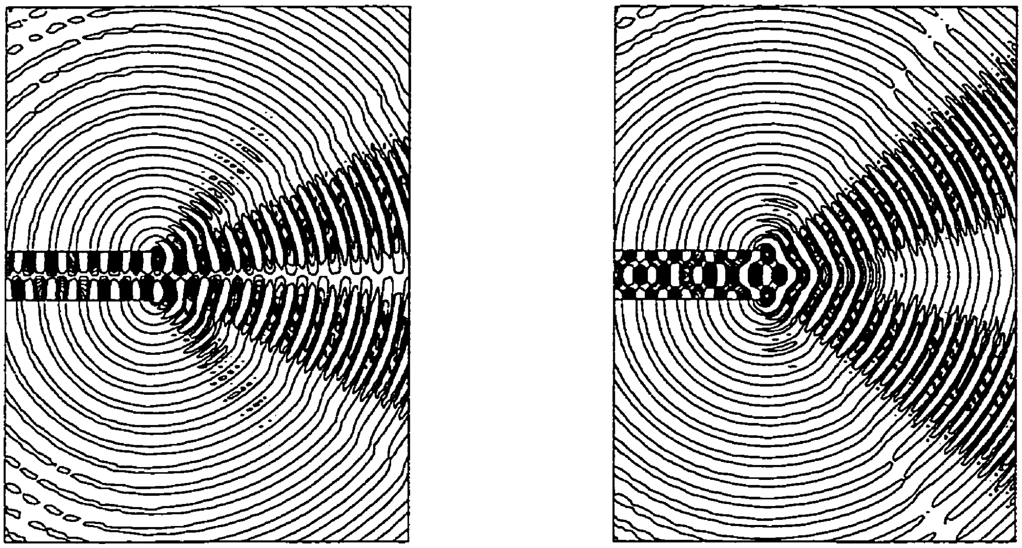


Figure 6.10: Acoustic pressure contours for the two-dimensional flat duct: first cross mode (left) and second cross mode (right).

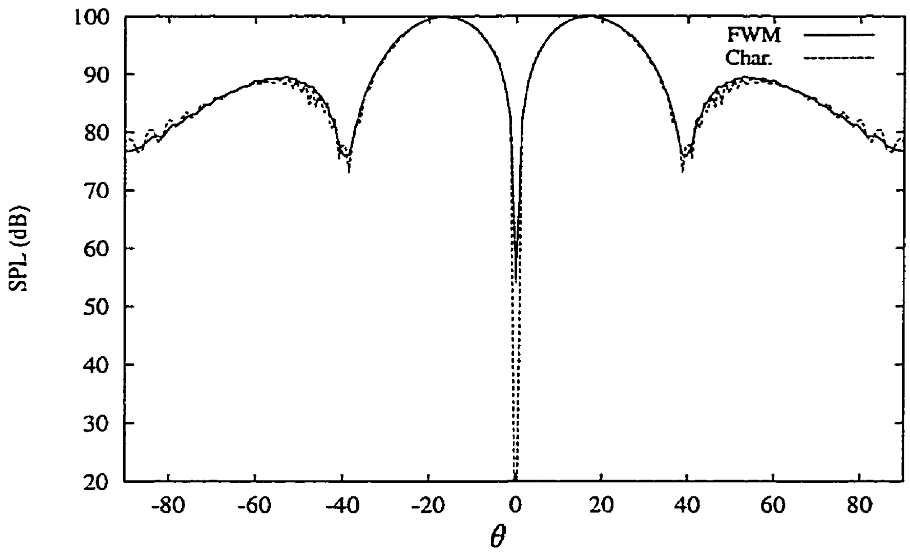


Figure 6.11: Directivity for the first cross mode computed with FWM and characteristic boundary conditions.

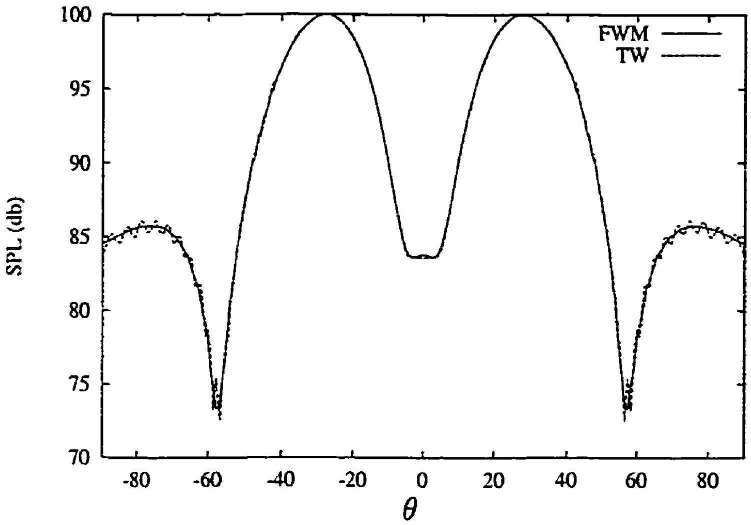


Figure 6.12: Directivity for the second cross mode computed with FWM and TW.

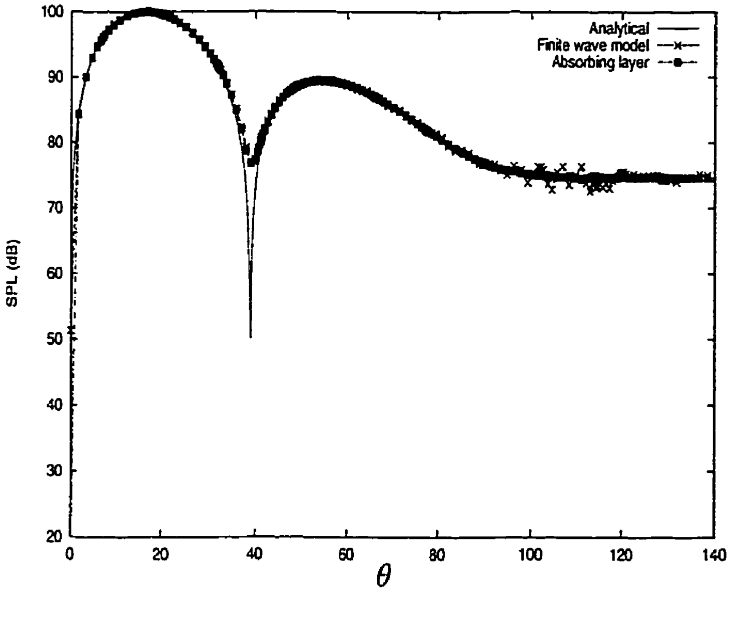


Figure 6.13: Directivity for the first cross mode computed with FWM, with and without the damping layer: comparison to the analytical solution.

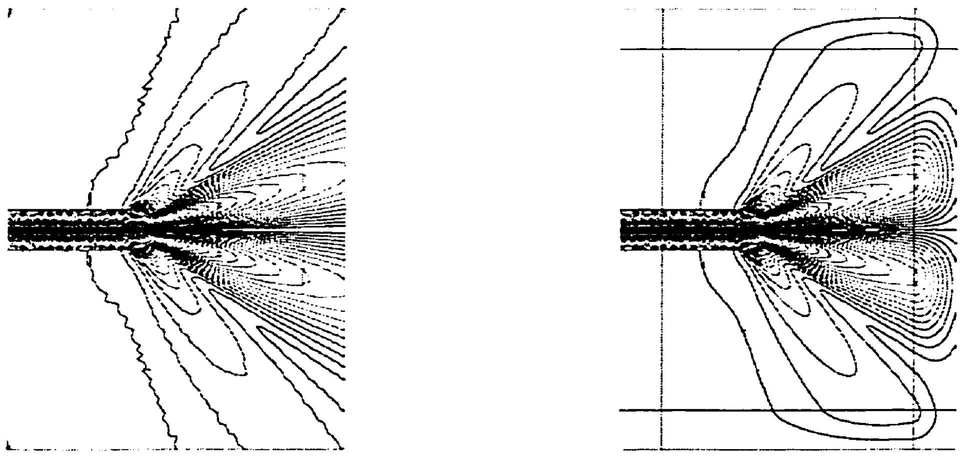


Figure 6.14: RMS pressure contours for the first cross mode: FWM (left); FWM plus an absorbing layer (shown) with $\sigma_M = 2.5$ (right).

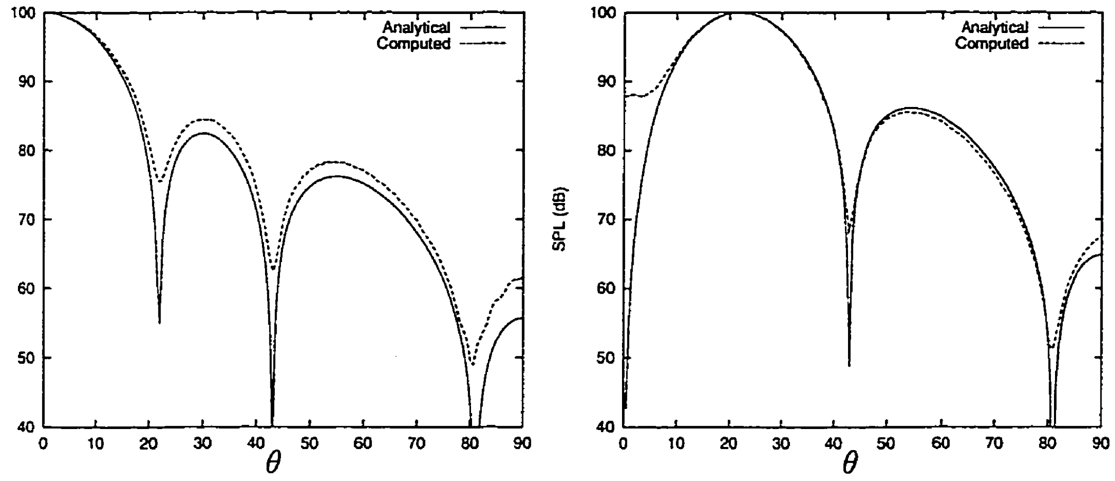


Figure 6.15: Directivity pattern for axisymmetric duct radiation, compared to the analytical solution for radiation from a plane circular piston. Left: plane wave, mode (0,0). Right: first radial mode (0,0).

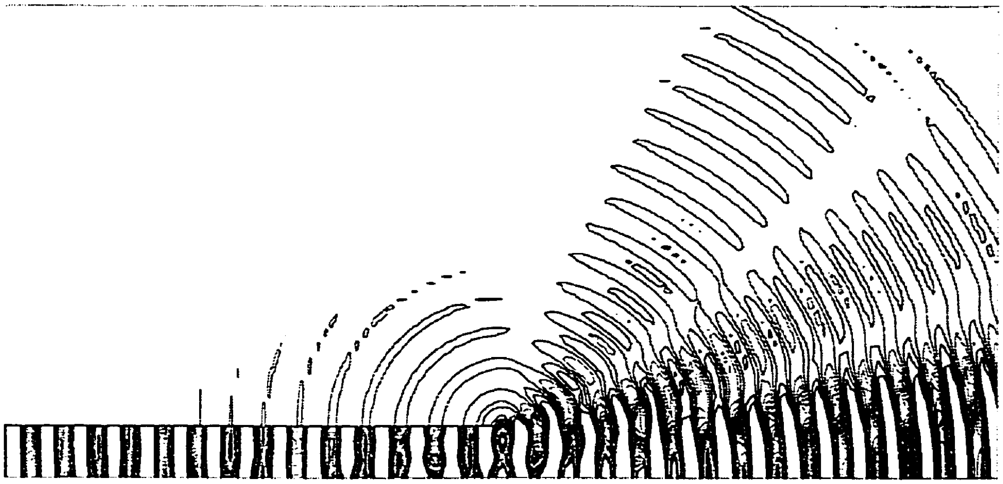


Figure 6.16: Acoustic pressure contours, axisymmetric (0,0) mode.

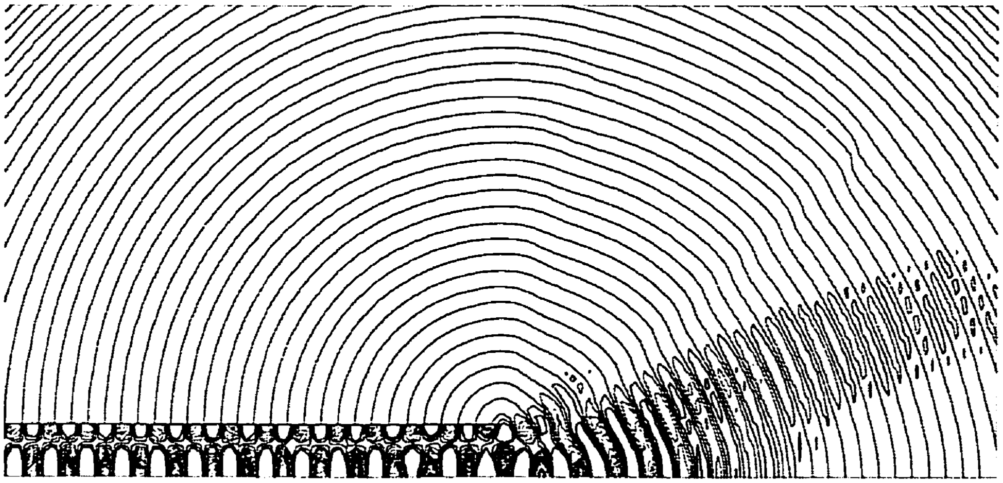


Figure 6.17: Acoustic pressure contours, axisymmetric (0,1) mode.

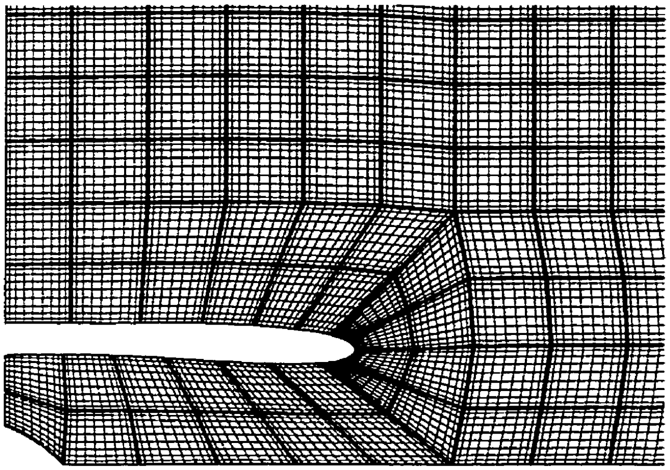


Figure 6.18: Multidomain spectral mesh for the fan flight inlet in the inlet region.

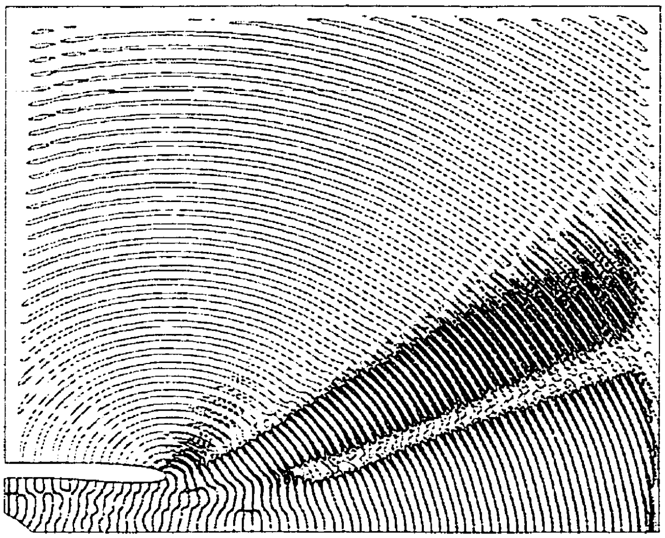


Figure 6.19: Acoustic pressure contours for the fan inlet, BPF(0,1) mode, $M_\infty = 0$.

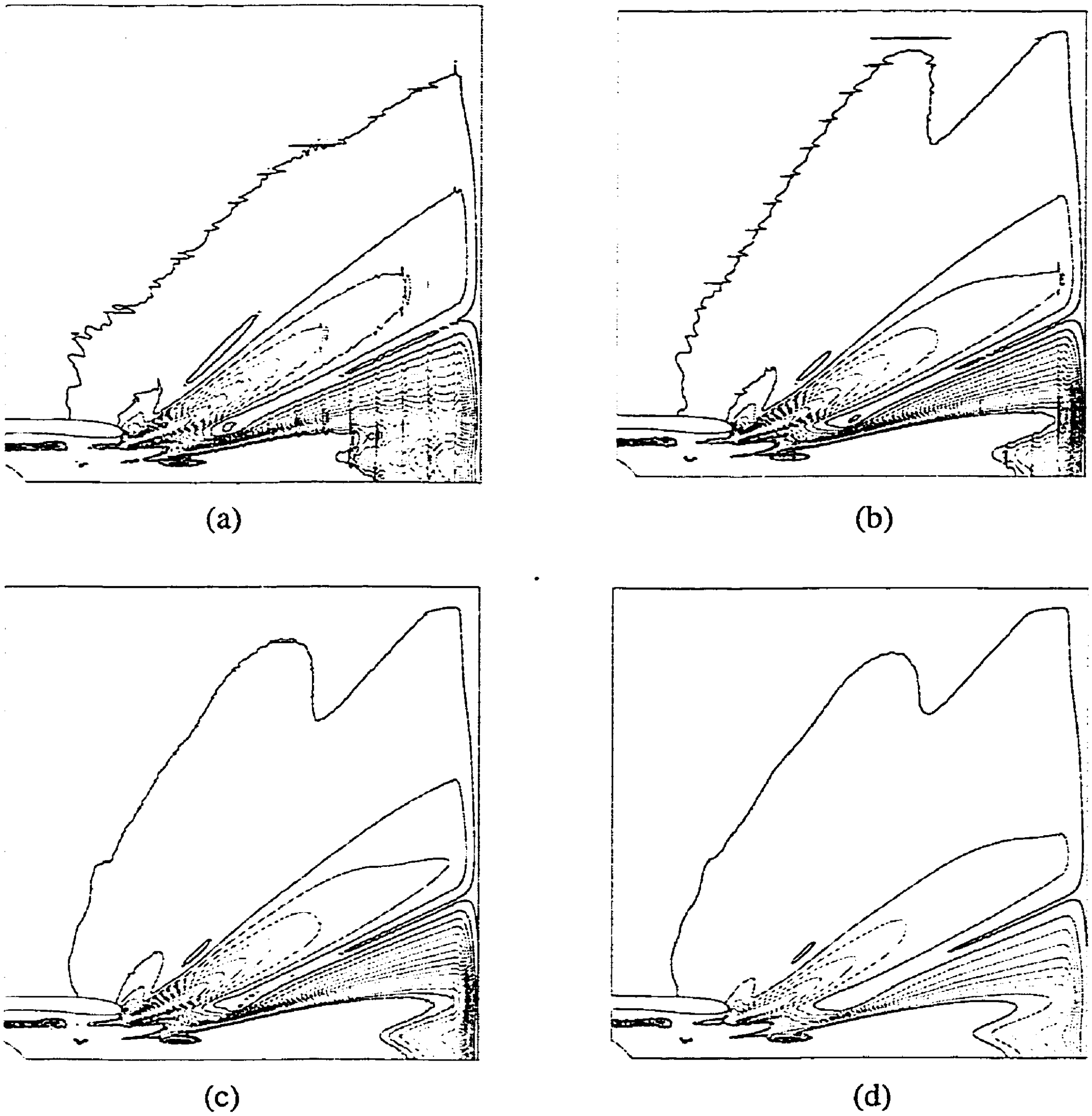


Figure 6.20: Root mean square pressure contours for the fan inlet, BPF(0,1) mode, $M_\infty = 0$: (a) $N = 10$; (b) $N = 11$; (c) $N = 12$; (d) $N = 13$.

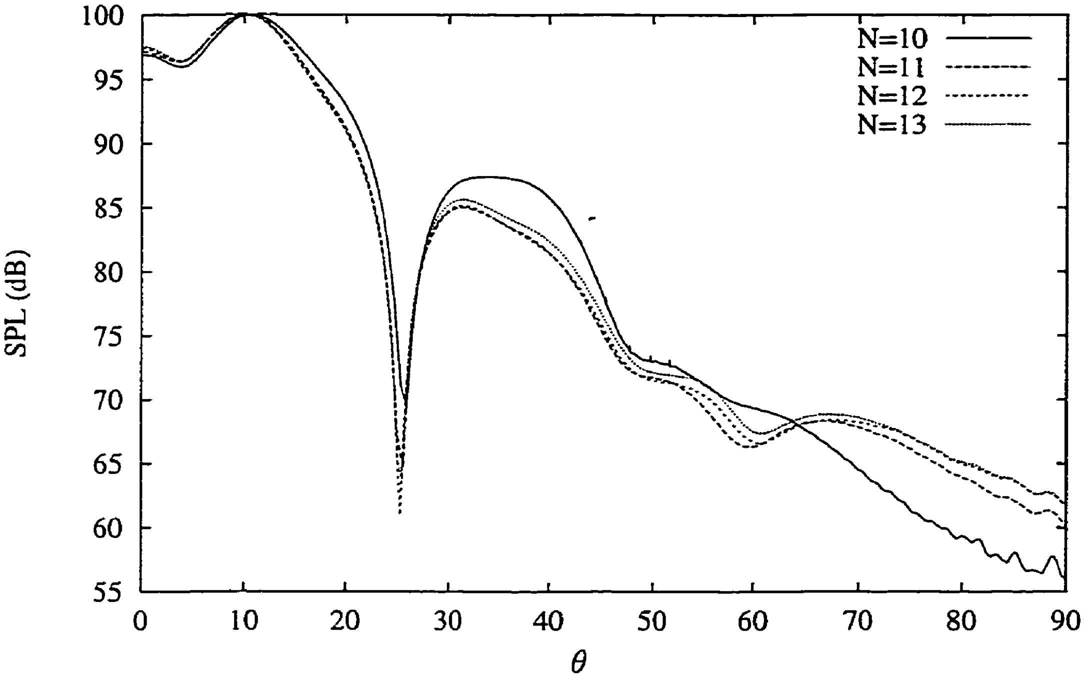


Figure 6.21: Far field directivity for the fan inlet in the BPF(0,1) mode, $M_\infty = 0$, as a function of the number of Gauss-Chebyshev points.

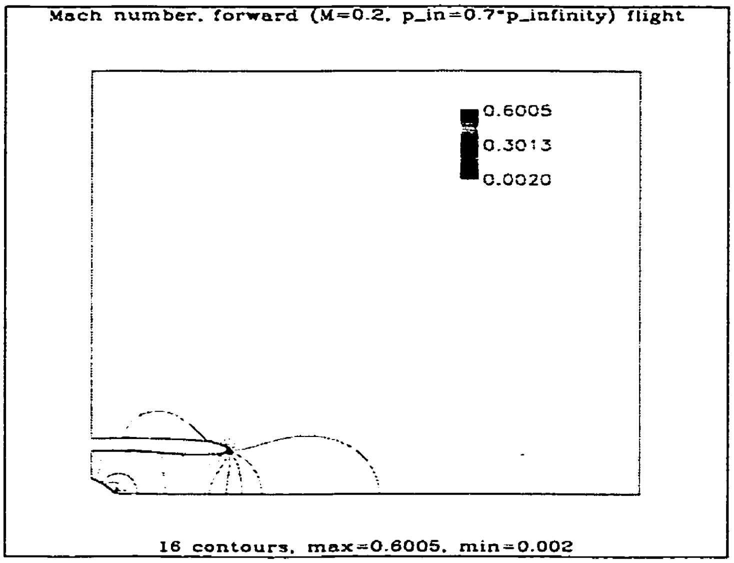


Figure 6.22: Mach number contours for the fan inlet, $M = 0.2$ forward flight case.

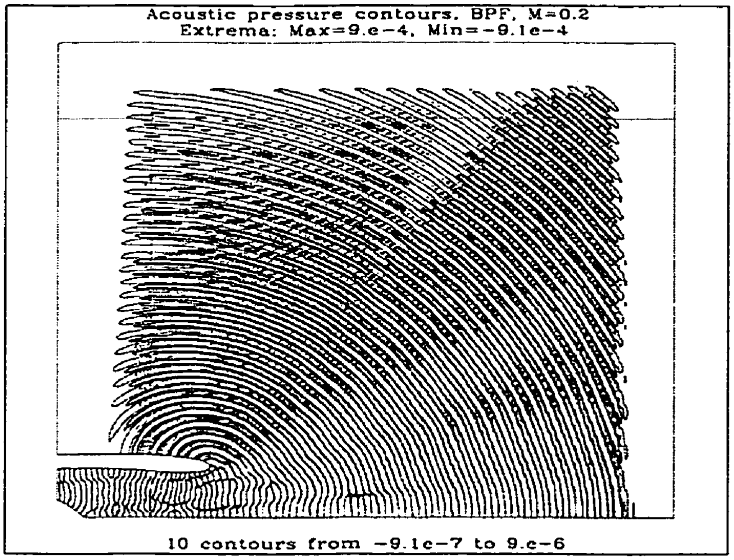


Figure 6.23: Acoustic pressure contours for the fan inlet, $M = 0.2$ forward flight case.

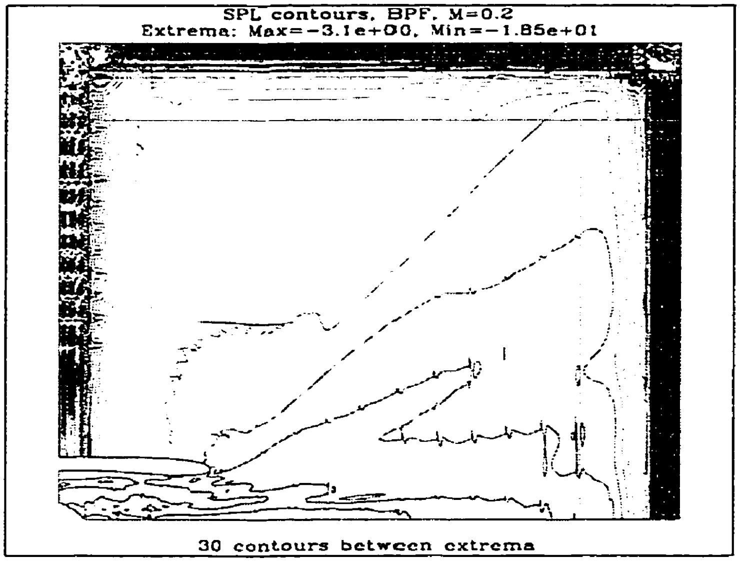


Figure 6.24: Sound pressure levels contours for the fan inlet, $M = 0.2$ forward flight case.

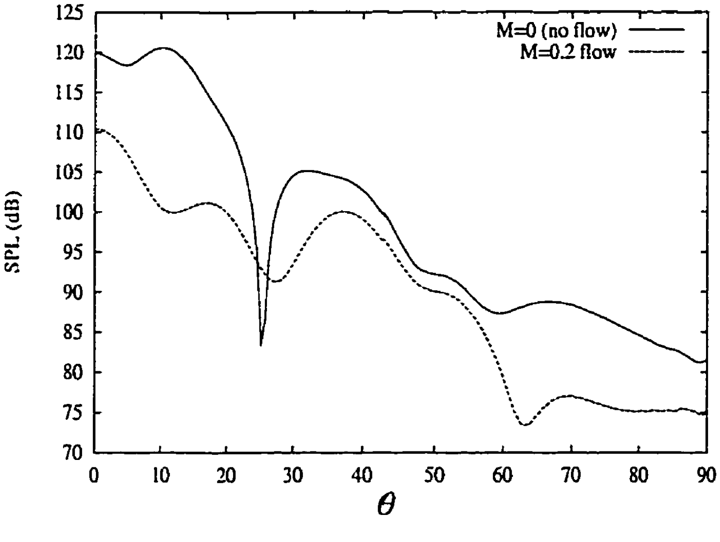


Figure 6.25: Far field directivity for the fan inlet in the $M = 0.2$ forward flight case and in the no mean flow ($M = 0$) case.

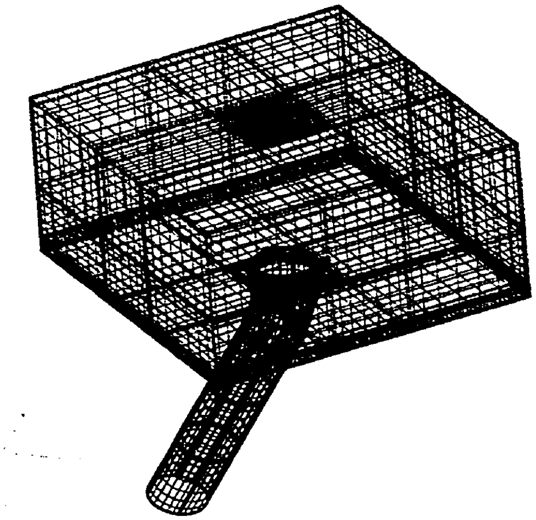
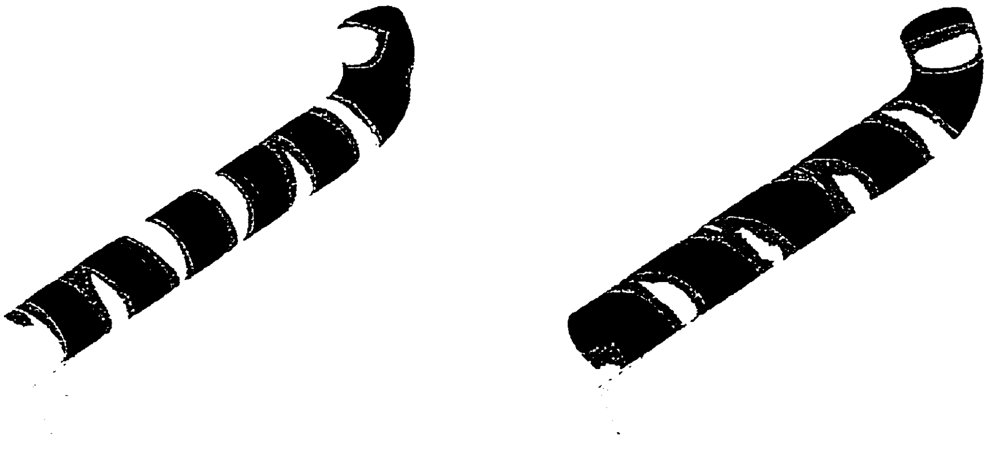


Figure 6.26: Wire frame of the mesh used for the turboshaft inlet model.



(a) Plane wave (b) First azimuthal mode

Figure 6.27: Snapshots of the acoustic pressure for the turboshaft inlet model.

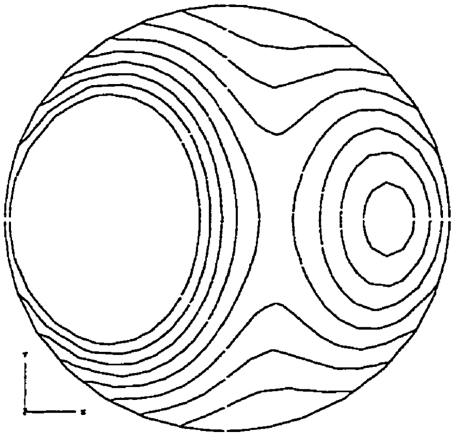


Figure 6.28: RMS pressure contours in the duct exit plane for the turboshaft engine model, plane wave.

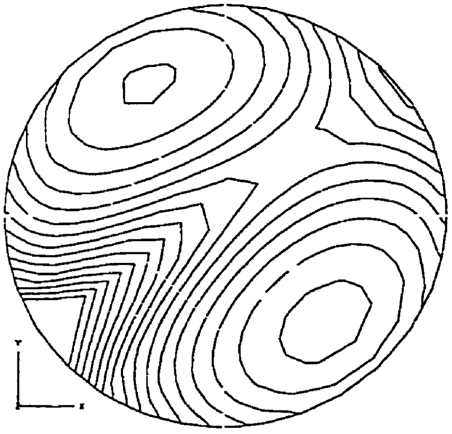
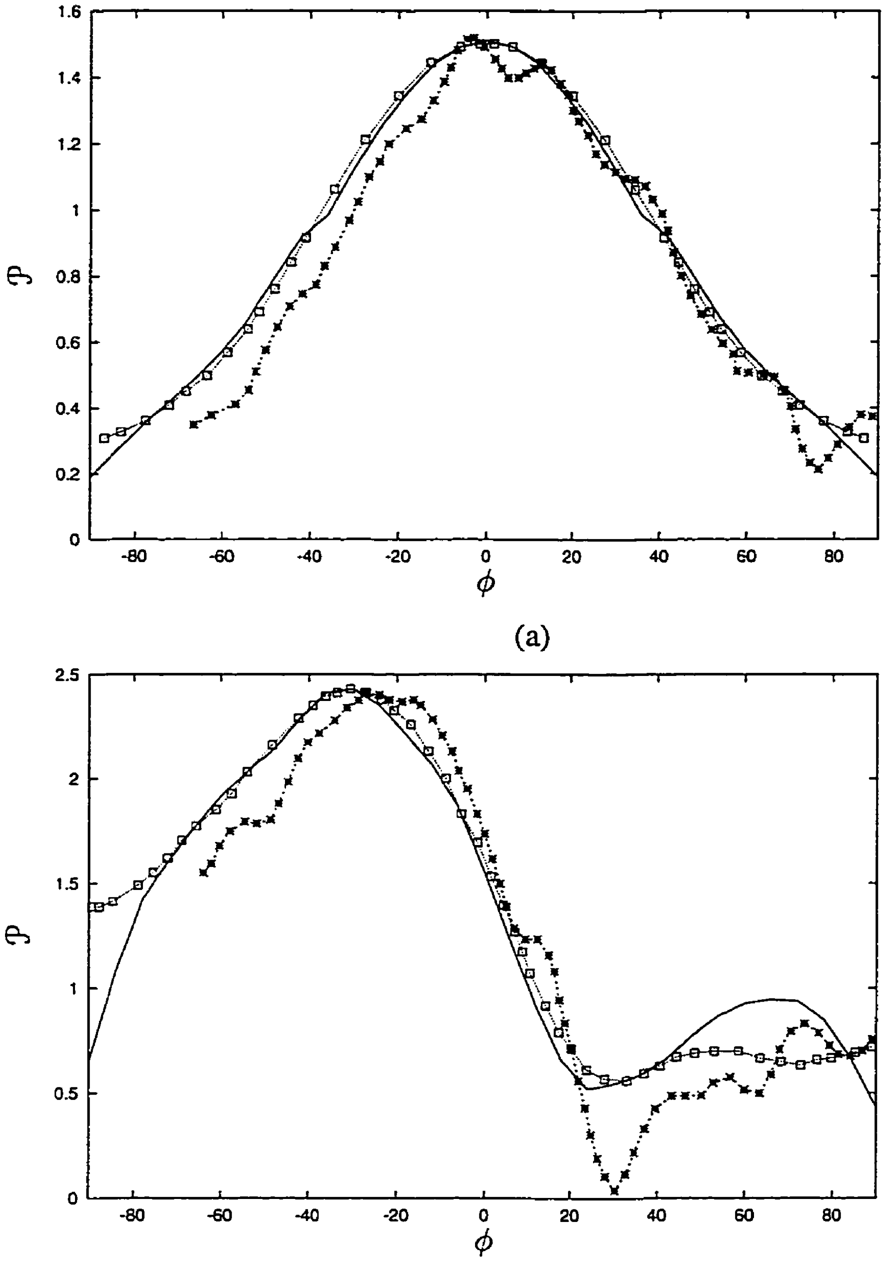


Figure 6.29: RMS pressure contours in the duct exit plane for the turboshaft engine model, first azimuthal mode.



(a)

(b)

Figure 6.30: RMS pressure directivity in the $x = 13.36$ plane for (a) plane wave (b) first azimuthal mode. Full line, present solution; squares, BIM computation (Malbéqui); asterisks, experiments (Malbéqui).

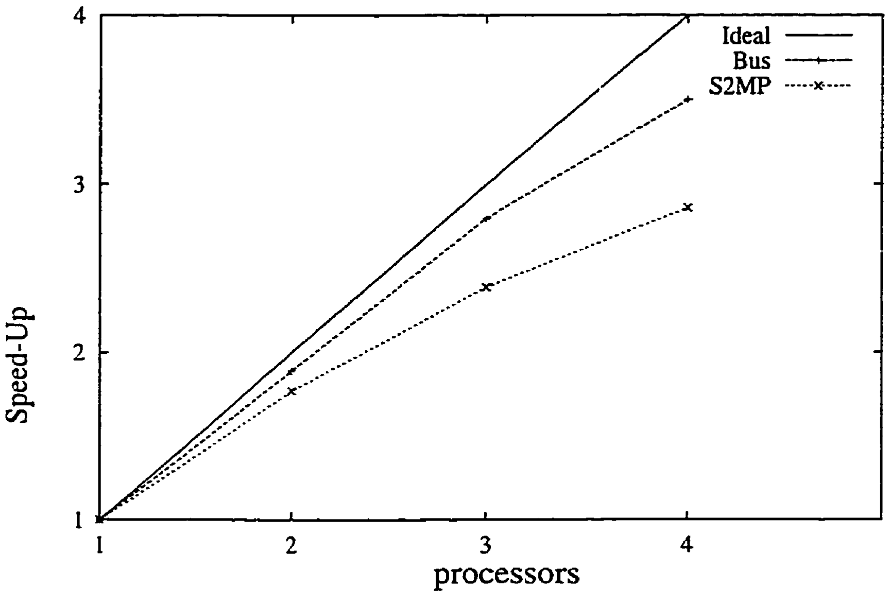


Figure 6.31: Speed-up obtained from parallelization on different architectures for the turboshaft engine inlet, $N = 11$.

# Chapter 7

# Accounting for known mean flow

## 7.1 Introduction

The method previously developed can be used to compute sound propagation when the mean flow itself is a solution of the steady-state Euler equations obtained by the same solver, and with the same discretization (number of elements, $E$, and Gauss points, $N$). This is not always practical, for different reasons. For example, it may be desirable to compute the mean flow by another Euler solver, which could converge faster than a spectral method, or on a different grid, since the mean flow needs not be computed to the same high precision as the sound field. Also, the mean flow can be governed by viscous effects which, while not largely affecting sound propagation, may still make a Navier-Stokes solver more appropriate. Another such situation arises when the mean flow is known from experiments. It is considered here that a mean flow provided by one of these means can be interpolated onto the grid used for aeroacoustic computations. This interpolated solution, designated here as the 'known mean flow', must be accounted for when solving for noise propagation.

One of the most straightforward ways to account for a known mean flow is to linearize the governing equations around it, considering the acoustic perturbations small enough that second and higher order terms in these quantities are negligible. Such an approach has many advantages and is valid for most aeroacoustic problems. There are, however, situations when the nonlinear terms must be accounted for. This chapter describes an algorithm, obtained by a slight modification of the one presented in chapter 2, that allows the computation of nonlinear acoustic perturbations on a known mean flow. This algorithm is tested on a jet noise problem, for which the mean flow is known from experiments.

## 7.2 Equations and algorithm

To obtain an appropriate set of equations for the acoustic quantities, a necessary condition is for the known mean flow to yield the steady-state solution of the governing equations. In general, there is no reason why this should happen, and the steady state equations discretized by the spectral algorithm will give:

$$\frac{\partial \bar{Q}}{\partial t} + \sum_{i=1}^{d} \frac{\partial \bar{F}_i}{\partial x_i} + v\bar{H}\delta_{2d} = \sum_{i=1}^{d} \frac{\partial \bar{F}_i}{\partial x_i} + v\bar{H}\delta_{2d} = \mathfrak{R} \tag{7.1}$$

with the residual $\mathfrak{R}$ different from zero.

When this equation is discretized using the spectral method described in chapter 2, $\bar{Q}$ is the known mean flow at the Gauss points, and $\bar{F}_i$ are the corresponding flux components, computed at the Gauss-Lobatto points by the interpolation procedure. Considering now a set of governing equations modified as

$$\frac{\partial(Q-\bar{Q})}{\partial t}+\sum_{i=1}^{d}\frac{\partial(F-\bar{F}_i)}{\partial x_i}+v(H-\bar{H})\delta_{2d}=0 \tag{7.2}$$

it is easy to see that, when the steady solution is obtained ($\partial/\partial t=0$), they become

$$\sum_{i=1}^{d}\frac{\partial F_i}{\partial x_i}+vH\delta_{2d}=\mathfrak{R}. \tag{7.3}$$

which is satisfied by the known mean flow, conforming to equation (7.1). This shows that the known mean flow is a steady state solution to equation (7.2), which is equivalent to

$$\frac{\partial(Q-\bar{Q})}{\partial t}+\sum_{i=1}^{d}\frac{\partial F_i}{\partial x_i}+vH\delta_{2d}=\mathfrak{R} \tag{7.4}$$

and shows that the perturbations of interest, $Q-\bar{Q}$, satisfy an equation of the same form as the Euler equations (2.1), with the exception that a source term coming from the spatial derivatives of the known mean flow fluxes is present on the right-hand side.

The source term can be easily computed, using the staggered grid method, from the known mean flow at the start of the computation. Since the vector $Q$ is not the vector of primitive variables, but the conserved quantities, acoustic perturbations, as for example $p-\bar{p}$, must be computed from $Q-\bar{Q}$ at each time step. The parallel algorithm presented in section 2.8 can then be adapted to solve equation (7.2) as follows:

```
for e=1 to number_of_elements do_parallel
   interpolation(Q_mean,e);
   F_mean := compute_fluxes(Q_mean,e)
endfor;

for t=1 to Max_Time_Steps do
  for e=1 to number_of_elements do_parallel
      interpolation(Q,e)
  endfor;
  for f=1 to number_of_faces do_parallel
      if( interface) then
         (L,R,f) := inverse_map(Q,e) ;
         Riemann_solve (L,R,f);
         (Q,e) := map(L,R,f)
      else
         impose_boundary_conditions(f)
      endif
  endfor;
  for e=1 to number_of_elements do_parallel
      F := compute_fluxes(Q,e) - F_mean;
      differentiation(F,e)
  endfor;
  compute_acoustic_pressure(Q-Q_mean);
endfor;
```

## 7.3 Supersonic jet mixing noise problem

An ideally expanded supersonic circular jet is considered here to demonstrate the validity of the approach of the previous section. Since the shock cell structure is absent, the radiated sound field has only one main component, turbulent mixing noise. It is well known that the free shear layer jet flow is characterized by large-scale coherent turbulent structures shedding from the jet lip. The perturbations of the mean jet flow that generate the mixing noise can therefore be considered as the sum of the coherent structures and a random perturbation. Both experimental [77] and theoretical [78] results show that the major part of this noise is produced by the large-scale turbulent structures of the jet flow, which are convected at supersonic Mach numbers relative to the ambient speed of sound. The generated sound field, similar to the one created by a wavy wall traveling supersonically, is very intense, and is known as Mach wave radiation since it tends to propagate primarily at the Mach angle.

In the mixing layer of the jet the mean turbulent flow is self-similar and, since the spreading rate of the jet is very small, the turbulence statistics change very slowly in the downstream direction [79]. This implies that the mean turbulent flow is in a state of local equilibrium, and its large-scale fluctuations can be represented as a linear superposition of the eigenmodes of its linear stability equations[1]. Indeed, the jet mean flow velocity profile has an inflexion point, a situation usually conducive [80] to the large-scale (Kelvin-Helmholtz) type of instability. Moreover, the influence of the viscosity on the solution of the linear stability equations can be neglected for the large Reynolds numbers involved, hence the large-scale turbulent fluctuations and the instability wave modes of the Euler equations linearized about the jet mean flow are practically similar. The latter can be regarded as a mathematical representation of the former, according to Tam and Chen [79].

Several models for supersonic jet mixing noise have been developed recently. Mankbadi *et al.* [81] and Baysal and Idres [82] solve the Euler equations linearized about the jet mean flow and use a time-harmonic excitation of the jet flow in the

[1] They can be put in the form of a single equation for the pressure or the vorticity disturbance, known as either the Rayleigh (in the inviscid case) or the Orr-Sommerfeld (in the viscous case) equation.

form of instability waves, the latter computed as solutions of the Rayleigh equation. A more direct approach that does not need the computation of the instability waves was used by Fenno *et al* [83]. The jet flow is excited by an acoustic pulse, as a result of which instability waves are generated. Due to the basic instability of the jet flow, the instability waves manifest themselves over time scales much longer than those required for the initial acoustic excitation to leave the domain, thus allowing the computation of the radiated sound. This approach is also followed in this thesis. Two types of excitation have been considered, the first one being an acoustic pulse, as in equation (6.1), and the second one an incoming duct mode similar to equation (4.3) but of a finite duration. Both cases led to identical results for the root mean square pressure.

To perform the study, coordinates have been non-dimensionalized using the jet exit diameter, and flow variables using the density and the speed of sound in the ambient flow. The equations are solved in axisymmetric mode. The exit of the jet is considered at $x = 0$, its axis at $r = 0$, and the grid covers the domain $(x, r) \in [-5, 50] \times [0, 25]$, with 5500 elements. The flow in the duct $(x < 0, r < 0.5)$, which does not influence the radiated noise, is considered to be a supersonic inviscid flow with $M = M_j = 2.1$, $M_j$ being the jet exit Mach number. The presence of the duct is necessary to construct the damping layers. The DL boundary condition is the only one that is acceptable for this study, because it also damps the vorticity waves generated within the mixing layer. It was found that the best combination of damping parameters in equation (4.8) for this case was $\sigma_M = 0.5$ and $\beta = 3$, with a damping layer width of about 5. No filtering or any kind of artificial dissipation has been used for the computation. The number of Gauss points was varied between $N = 3$ to $N = 5$. This is more than sufficient to resolve the acoustic field. In fact, for this case the resolution is mainly dictated by the large gradients present in the shear layer, especially in the initial region near the duct exit. The mean flow in the jet region corresponds to a Gaussian profile that has been found by Troutt and McLaughlin [77] to fit well the experimental results for a moderate Reynolds number ($\mathcal{R}$=70000) jet,

$$\frac{\bar{u}(\eta_j)}{U_j} = \begin{cases} \exp\left[-2.773(\eta_j + 0.5)^2\right], & \eta_j > -0.5 \\ 1, & \eta_j \leq -0.5 \end{cases} \tag{7.5}$$

where $U_j$ is the velocity at the center line of the jet and the non-dimensional coordinate $\eta_j = (r - r_{0.5})/t_s$, with $r_{0.5}$ the radial location where the velocity is $0.5U_j$, and $t_s$ the local shear-layer thickness. The variation of $U_j$, $r_{0.5}$ and $t_s$ as obtained from experiments is provided in [77] for $x \leq 16$. Since the emphasis here is on demonstrating the methodology, and not obtaining quantitative evaluations of jet noise, a very simple linear fit to the data has been used, as follows:

$$U_j = \begin{cases} M_j, & x \leq 7 \\ M_j \cdot \left(1 - \frac{x-7}{43}\right), & x > 7 \end{cases} \tag{7.6}$$

$$r_{0.5} = \begin{cases} 0.5 + 0.1 \cdot \frac{x}{7}, & x \leq 7 \\ 0.6 + 0.0482 \cdot (x - 7), & x > 7 \end{cases} \tag{7.7}$$

$$t_s = \begin{cases} 2 \cdot \left(0.05 + 0.45 \cdot \frac{x}{7}\right), & x \leq 7 \\ 1 + 0.1 * (x - 7), & x > 7 \end{cases} \tag{7.8}$$

In addition, the radial velocity component is considered zero throughout the flow field, and the pressure across the jet is considered constant by the boundary layer approximation. For an isothermal jet, it follows that the density is also constant.

Figure 7.1 presents a snapshot of the disturbance pressure just after the initial excitation completely left the domain. The instability waves can be clearly noticed in the figure. They initially grow until they reach the end of the potential core of the jet (approximately $x = 7$), and decay afterwards. The end of the potential core is also the region mainly responsible for the generation of the acoustic field. This has the characteristics generally observed for such jets, the main radiation lobe making an angle of about 42 degrees with the jet axis. This angle is slightly larger than the one observed experimentally in [77], a phenomenon that has also been noticed by other researchers [82] when only axisymmetric modes are considered. A secondary lobe can be noticed at about 30 degrees, which is the Mach angle for this jet. The root mean square pressure is displayed in figure 7.2. The boundary conditions can be seen to perform very well, with no noticeable reflections from the boundaries. The FWM boundary condition has been used at the end of the damping layer without stability problems. Thus, overall the results demonstrate that the method can also be used to compute nonlinear sound propagation on a known mean flow.

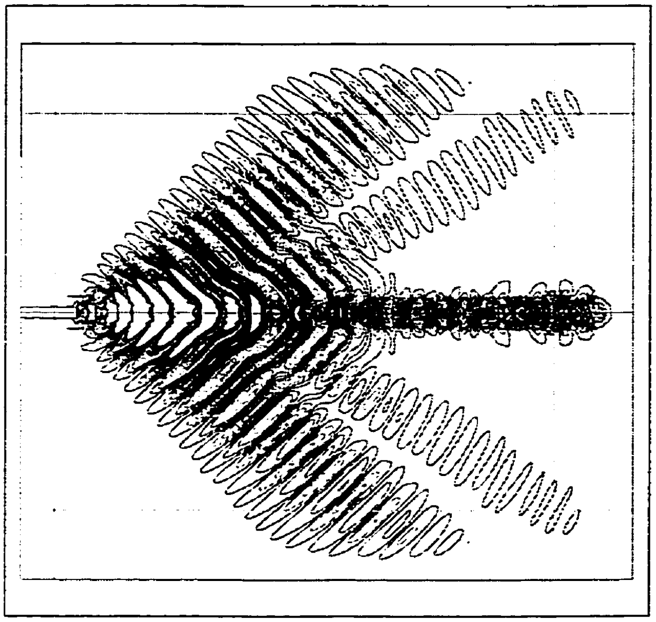


Figure 7.1: Snapshot of acoustic pressure contours for the turbulent jet.

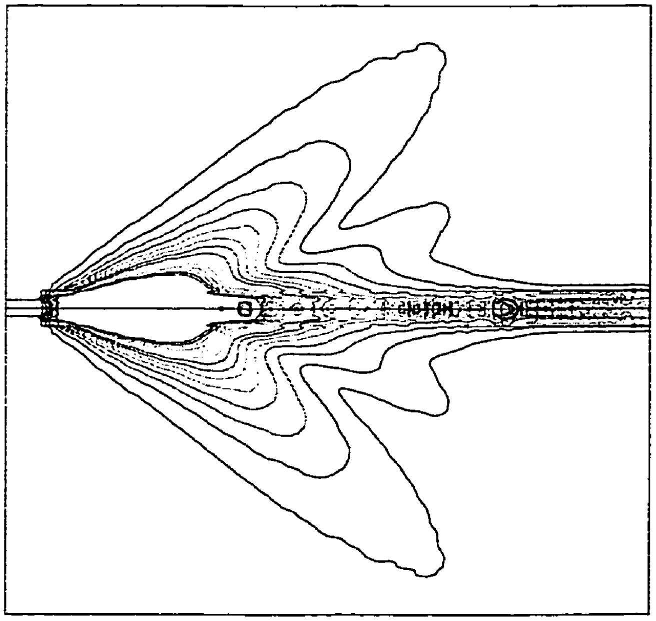


Figure 7.2: Root mean square pressure contours for the turbulent jet.

# Chapter 8

# Conclusions and recommendations for future work

A multidomain collocation spectral method has been developed in this thesis for the solution of the three-dimensional nonlinear Euler equations of fluid dynamics. The algorithm has been supplemented with all the necessary extensions required to model acoustic phenomena. These extensions include a set of low-storage Runge-Kutta schemes for integration in time, that keep their formal order of accuracy when applied to nonlinear systems of equations and are optimized for wave propagation. They also include boundary conditions for the specification of incoming duct acoustic modes and treatment of the far-field boundaries, and specific modules for the computation of the unsteady acoustic pressure as well as the root mean square pressure from the time-dependent flow variables. In addition, numerical filters, that prove useful to ensure stability of the solution when using some particular boundary conditions, have also been incorporated into the algorithm.

The numerical results obtained with this method, which most probably include the first applications of spectral methods to complex three-dimensional geometries, are quite encouraging. Even though the number of points per wavelength was very small in most cases, the results match the analytical solutions extremely well in the linear cases for which these exist. Grid refinement studies show that the small discontinuities between the elements, which are of the order of the truncation error in the computed quantities, decrease very fast when the number of Gauss-Chebyshev points is increased. These results suggest that multidomain spectral methods are a very viable alternative for solution of aeroacoustics problems (or equivalent wave propagation problems, such as electromagnetics) in three dimensions.

Of considerable help for the discretization of complex geometries in this case is the fact that the algorithm can use an unstructured grid of hexahedra. Although in three dimensions the element grids are usually generated as structured multi-block grids, they become unstructured when seen as a single grid. For the present method, there is no longer need to keep a particular index structure for each block, and the connection between the elements, either within the same block or belonging to different ones, is constructed completely within the code. This makes the method as general as any finite element method that uses hexahedral grids.

## 8.1 Automatic grid generation

In chapter 5, a way to construct the spectral grids by transfinite interpolation, which provides a fast isoparametric mapping, has been developed. Although there are advantages to this methodology, it is probably not appropriate to use in an industrial environment, where geometries are usually defined using CAD programs. A further development, undertaken at Pratt & Whitney Canada in collaboration with the CFD Laboratory, is to build the face grids using the projection on the CAD surfaces of the grids obtained through transfinite interpolation. In this way, the spectral grids can be constructed by widely-used commercial mesh generation software, such as ICEM-CFD.

## 8.2 Physical modeling extensions

There are aspects that have not been touched upon in this thesis, for example the treatment of shock and contact discontinuities. The treatment of such phenomena with multidomain spectral methods is not straightforward, although some researchers consider that this problem has been solved at the mathematical level for the global (one-domain) spectral discretization. Possible directions to extend the method for discontinuous flows include the spectral viscosity method, first introduced by Tadmor [84, 85], the use of Gegenbauer polynomials advocated by Gottlieb and Shu [86], or the Chebyshev-Legendre super spectral viscosity method studied in detail by Ma [87]. The main difficulty in the multidomain case is propagating discontinuities through domain interfaces without generating oscillations.

In many situations, the noise produced by turbulence is the interesting phenomenon. In order to compute it directly from the basic equations of fluid dynamics the viscous terms must be discretized, with the possible use of a turbulence model. While the viscous terms can be computed relatively easily in the multidomain approach, the turbulence model has to be chosen with extreme care, to avoid stability problems when coupled with a spectral discretization. As the fan blades rotate with respect to the struts and stator vanes, the additional incorporation of a moving grid strategy would allow the direct prediction of rotor-stator interaction noise sources. Another important issue that must be addressed in order to develop an useful tool for fan acoustic analysis is the development of a time-domain model for possible acoustic liners along the walls. The issues above constitute as many directions of future development of the multidomain spectral method presented herein.

# Appendix A

# Basic properties of Chebyshev polynomials

The Chebyshev polynomial of degree $j$, $T_j(x)$, $-1 \leq x \leq 1$, is defined by

$$T_j(x) = \cos\left(j \arccos(x)\right). \tag{A.1}$$

Chebyshev polynomials are solutions of the differential equation

$$(1-x^2)\frac{d^2T_j}{dx^2} - x\frac{dT_j}{dx} + j^2T_j(x) = 0 \tag{A.2}$$

that are bounded at $x = \pm 1$, and satisfy an orthogonality relation of the form

$$\int_{-1}^{1} \frac{T_j(x)T_i(x)}{\sqrt{1-x^2}}dx = \frac{\pi}{2}C_j\delta_{ij} \tag{A.3}$$

where $C_0 = 2$, $C_j = 1$ otherwise. They satisfy the recurrence relation

$$T_{j+1}(x) = 2xT_j(x) - T_{j-1}(x), \tag{A.4}$$

hence the first polynomials are $T_0(x) = 1, T_1(x) = x$, $T_2(x) = 2x^2 - 1, T_3(x) = 4x^3 - 3x$, and so on.

The Chebyshev polynomials also satisfy orthogonality relations over discrete sets of points. To define these sets, it is useful to let $x = -\cos(\theta)$, such that when $\theta$ varies from 0 to $\pi$, $x$ varies from $-1$ to $1$. A first set of points, known as Gauss-Chebyshev-Lobatto points, is defined by the $N+1$ equally spaced $\theta$ values

$$\theta_n = \frac{\pi n}{N},\ n = 0, 1, \ldots, N; \qquad \{\theta_n\} = \left\{0, \frac{\pi}{N}, \frac{2\pi}{N}, \ldots, \frac{(N-1)\pi}{N}, \pi\right\}$$

over which the following orthogonality relation holds:

$$\frac{1}{2}T_i(-1)T_j(-1) + \sum_{n=2}^{N-1} T_i(x_n)T_j(x_n) + \frac{1}{2}T_i(1)T_j(1) = \begin{cases} 0, & i \neq j \\ N/2, & i = j \neq 0 \\ N, & i = j = 0 \end{cases} \tag{A.5}$$

where $x_n = -\cos(\theta_n), n = 0, 1, \ldots, N$.

The other set of points is obtained as mid-points in $\theta$ of the Gauss-Chebyshev-Lobatto points. Hence the set contains $N$ points, denoted as Gauss-Chebyshev points, and defined by

$$\bar{\theta}_m = \frac{(2m-1)\pi}{2N}, m = 1, \ldots, N; \qquad \{\bar{\theta}_m\} = \left\{\frac{\pi}{2N}, \frac{3\pi}{2N}, \ldots, \frac{(2N-1)\pi}{2N}\right\}$$

with the corresponding orthogonality relation for $\bar{x}_m = -\cos(\bar{\theta}_m), m = 1, \ldots, N$:

$$\sum_{m=1}^{N} T_i(\bar{x}_m)T_j(\bar{x}_m) = \begin{cases} 0, & i \neq j \\ N/2, & i = j \neq 0 \\ N, & i = j = 0 \end{cases} \tag{A.6}$$

Although the points in the two sets are equispaced in $\theta$, they are not equally spaced in $x$. In fact, the distance between the two Gauss-Chebyshev-Lobatto points nearest of one extremity of the interval in $x$ is

$$\Delta x_m = 1 - \cos\left(\frac{\pi}{N}\right) = \frac{1}{2}\left(\frac{\pi}{N}\right)^2 - \frac{1}{24}\left(\frac{\pi}{N}\right)^4 + \ldots = \mathcal{O}(1/N^2) \tag{A.7}$$

It follows from the fact that Chebyshev polynomials are actually cosines in disguise that they have an equi-ripple behaviour in $[-1, 1]$. A consequence of this is that interpolation of arbitrary functions $f(x)$ based upon the Gauss-Chebyshev-Lobatto nodes gives polynomials that are always within a very small factor from the optimal polynomial interpolation of $f$ in the maximum norm. Indeed, letting $P_C^f$ be the Chebyshev interpolant and $P_{opt}^f$ be the optimal polynomial interpolant of $f$, one has

$$||f - P_C^f||_\infty \leq (1 + \Lambda_C)||f - P_{opt}^f||_\infty \tag{A.8}$$

where $\Lambda_C$ is the Lebesgue constant for Chebyshev interpolation, that depends only on the order of the interpolant as $\Lambda_C = \mathcal{O}(\ln N)$. For comparison, for equi-spaced interpolation (as in finite difference methods), the Lebesgue constant is $\Lambda_{es} = \mathcal{O}[2^N/(N\ln N)]$.

# Bibliography

[1] Lighthill, M.J., “On sound generated aerodynamically. I. General theory”, *Proceedings of the Royal Society of London*, Series A, Vol. 221, No. 1107, pp. 564-587, 1952.

[2] Lighthill, M.J., “On sound generated aerodynamically. II. Turbulence as a source of sound”, *Proceedings of the Royal Society of London*, Series A, Vol. 222, No. 1148, pp. 1-32, 1954.

[3] Tam, C.K.W., “Computational Aeroacoustics: Issues and Methods”, AIAA Paper 95-0677, January 1995.

[4] Groeneweg, J.F., Sofrin, T.G., Rice, E.J., and Gliebe, P.R., “Turbomachinery Noise”, *Aeroacoustics of Flight Vehicles, Theory and Practice*, Vol. 1, Hubbard, H.H., Editor, Acoustical Society of America, pp. 151-210, 1995.

[5] Huff, D.L., “Fan Noise Prediction: Status and Needs”, AIAA Paper 98-0177, January 1998.

[6] Tyler, J.M., and Sofrin, T.G., "Axial Flow Compressor Noise Studies", *SAE Transactions*, Vol. 70, 1962, pp. 309-332.

[7] Kaji, S., and Okazaki, T., "Generation of sound by rotor-stator interaction", *J. Sound Vib.*, Vol. 13, 1970, pp. 281-307.

[8] Namba, M., "Three-dimensional analysis of blade force and sound generation for an annular cascade in distorted flows", *J. Sound Vib.*, Vol. 50, 1977, pp. 479-508.

[9] Ffowcs Williams, J.E., and Hawkings, D.L., "Sound generation by turbulence and surfaces in arbitrary motion", *Philosophical Transactions of the Royal Society of London*, Series A, Vol. 264, pp. 321-341, 1969.

[10] Schulten, J.B.H.M., "Sound generation by ducted fans and propellers as a lifting surface problem", Ph.D. Thesis, University of Twente, Enschede, Holland, 1993.

[11] Schulten, J.B.H.M, "Vane sweep Effects on Rotor/Stator Interaction Noise", AIAA Paper 96-1694, May 1996.

[12] Atassi, H., Fang, J., and Hardy, B., "Sound Radiated From a Loaded Cascade in Nonuniform Transonic Flow", AIAA Paper 96-1756, May 1996.

[13] Rangwalla, A.A., and Rai, M.M., "A Numerical Analysis of Tonal Acoustics in Rotor-Stator Interactions", *J. Fluids and Structures*, Vol. 7, 1993, pp. 611-637.

[14] Mani, R., "Refraction of Acoustic Duct Waveguide Modes by Exhaust Jets", *Quarterly of Appl. Math.*, Vol. 30, 1973, pp. 501-520.

[15] Savkar, S.D., "Radiation of Cylindrical Duct Acoustic Modes with Flow Mismatch", *J. Sound Vib.*, Vol. 42, 1975, pp. 363-386.

[16] Parrett, A.V., and Eversman, W., "Wave envelope and finite element approximations for turbofan noise radiation in flight", *AIAA Journal*, Vol. 24, 1986, pp. 753-760.

[17] Eversman, W., and Roy, I.D., "Ducted fan acoustic radiation including the effects of nonuniform mean flow and acoustic treatment", AIAA Paper 93-4424, October 1993.

[18] Caruthers, J.E., French, J.C., and Raviprakash, G.K., "Green function discretization for numerical solution of the Helmholtz equation", *J. Sound Vib.*, Vol. 187, 1995, pp. 553-568.

[19] Özyörük, Y., and Long, L.N., "Computation of Sound Radiating from Engine Inlets", *AIAA Journal*, Vol. 34, 1996, pp. 894-901.

[20] Özyörük, Y., and Long, L.N., "A New Efficient Algorithm for Computational Aeroacoustics on Parallel Processors", *J. Comp. Physics*, Vol. 125, 1996, pp. 135-149.

[21] Tam, C.K.W., and Webb, J.C., "Dispersion-Relation-Preserving Finite Difference Schemes for Computational Acoustics", *J. Comp. Physics*, Vol. 107, 1993, pp. 262-281.

[22] Lele, S. J., "Compact Finite Difference Schemes with Spectral-like Resolution", *J. Comp. Physics*, Vol. 103, 1992, pp. 16-42.

[23] Baysal, O., Kaushik, D.K., and Idres, M.M., "Low Dispersion Scheme for Nonlinear Acoustic Waves in Nonuniform Flow", AIAA Paper 97-1582, June 1997.

[24] Shen, H., and Tam, C.K.W., "Numerical Simulation of the Generation of Axisymmetric Mode Jet Screech Tones", AIAA Paper 98-0283, January 1998.

[25] Casper, J., "Finite-Volume Implementation of High-Order Essentially Nonoscillatory Schemes in Two Dimensions", *AIAA Journal*, Vol. 30, 1992, pp. 2829-2835.

[26] Casper, J., and Meadows, K.R., "Using High-Order Accurate Essentially Non-Oscillatory Eschemes for Aeroacoustic Applications", AIAA Paper 95-0163, January 1995.

[27] Stanescu, D., and Habashi, W.G., "Essentially Non-Oscillatory Euler Solutions on Unstructured Meshes Using Extrapolation", *AIAA Journal*, Vol. 36, 1998, pp. 1413-1416.

[28] Fornberg, B., *A practical guide to pseudospectral methods*, Cambridge Univ. Press, 1996.

[29] Canuto, C., Hussaini, M.Y., Quarteroni, A., and Zang, T.A., *Spectral Methods in Fluid Mechanics*, Springer-Verlag, 1987.

[30] Patera, A.T., "A Spectral Element Method for Fluid Dynamics: Laminar Flow in a Channel Expansion", *J. Comp. Physics*, Vol. 54, 1984, pp. 468-488.

[31] Cai, W., Gottlieb, D., and Shu, C.-W., "Essentially Nonoscillatory Spectral Fourier Methods for Shock Wave Calculations", *Mathematics of Computation*, Vol. 52, 1989, pp. 389-410.

[32] Sidilkover, D., and Karniadakis, G.E., "Non-oscillatory Spectral Element Chebyshev Method for Shock Wave Calculations", *J. Comp. Physics*, Vol. 107, 1993, pp. 10-22.

[33] Kopriva, D.A., and Kolias, J.H., "A Conservative Staggered-Grid Chebyshev Multidomain Method for Compressible Flows", *J. Comp. Physics*, Vol. 125, 1996, pp. 244-261.

[34] Hesthaven, J.S., "A Stable Penalty Method for the Compressible Navier-Stokes Equations. III. Multi Dimensional Domain Decomposition Schemes", *SIAM J. Sci. Computing*, accepted.

[35] Dowling, A.P., and Ffowcs Williams, J.E., *Sound and sources of sound*, Halsted Press, 1983.

[36] Kinsler, L.E., Frey, A.R., Coppens, A.B., and Sanders, J.V., *Fundamentals of Acoustics*, 3rd edition, John Wiley & Sons, 1982.

[37] Hirsch, C., *Numerical Computation of Internal and External Flows*, J. Wiley, 1990.

[38] Hoffmann, K.A., *Computational Fluid Dynamics for Engineers*, Engineering Education System, Austin, Texas, 1989.

[39] Godunov, S.K., "Difference method for the numerical computation of discontinuous solutions of equations of hydrodynamics", *Mat. Sbornik* Vol. 47, 1957, pp. 271-306.

[40] Engquist, B., and Osher, S., "Stable and entropy satisfying approximations for transonic flow calculations", *Mathematics of Computation*, Vol. 34, 1980, pp. 45-75.

[41] Engquist, B., and Osher, S., "One-sided difference approximations for nonlinear conservation laws", *Mathematics of Computation*, Vol. 36, 1981, pp. 321-352.

[42] Roe, P.L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes", *J. Comp. Physics*, Vol. 43, 1981, pp. 357-372.

[43] Stanescu, D., *Computational Methods for Compressible Aerodynamic Flows*, M.Eng. thesis, Dept. of Mech. Eng., McGill University, August 1994.

[44] LeVeque, R.J., "Hyperbolic Conservation Laws and Numerical Methods", *von Karman Institute Lecture Series 1990-03*, March 5-9, 1990.

[45] Roe, P.L., "Discrete Models for the Numerical Analysis of Time-Dependent Multidimensional Gas Dynamics", *J. Comp. Physics*, Vol. 63, 1986, pp. 458-476.

[46] Lacor, C., and Hirsch, Ch., "Genuinely Upwind Algorithms for the Multidimensional Euler Equations", *AIAA Journal*, Vol. 30, 1991, pp. 56-63.

[47] Powell, K.G., van Leer, B., and Roe, P.L., "Towards a Genuinely Multi-Dimensional Upwind Scheme", *von Karman Institute Lecture Series 1990-03*, March 5-9, 1990.

[48] Butcher, J.C., *The Numerical Analysis of Ordinary Differential Equations*, John Wiley & Sons, 1987.

[49] Hu, F.Q., Hussaini, M.Y., and Manthey, J., "Low-Dissipation and Low-Dispersion Runge-Kutta Schemes For Computational Acoustics", *J. Comp. Physics*, Vol. 124, 1996, pp. 177-191.

[50] Williamson, J.H., "Low-Storage Runge-Kutta Schemes", *J. Comp. Physics*, Vol. 35, 1980, pp. 48-56.

[51] Carpenter, M.H. and Kennedy, C.A., "A Fourth-Order 2N-Storage Runge-Kutta Scheme", NASA TM-109112, June 1994.

[52] Zingg, D.W., and Chisholm, T.T., "Runge-Kutta Methods for Linear Ordinary Differential Equations", RIACS Technical Report 97.07, July 1997.

[53] Thompson, K.W., "Time-Dependent Boundary Conditions for Hyperbolic Systems, II", *J. Comp. Physics*, Vol. 89, 1990, pp. 439-461.

[54] Hixon, R., Shih, S.-H., and Mankbadi, R.R., "Evaluation of Boundary Conditions for Computational Aeroacoustics", *AIAA Journal*, Vol. 33, 1995, pp. 2006-2012.

[55] Atkins, H., and Casper, J., "Nonreflective Boundary Conditions for High-Order Methods", *AIAA Journal*, Vol. 32, 1994, pp. 512-518.

[56] Bayliss, A., and Turkel, E., "Far Field Boundary Conditions for Compressible Flows", *J. Comp. Physics*, Vol. 48, 1982, pp. 182-199.

[57] Bayliss, A., and Turkel, E., "Outflow boundary conditions for fluid dynamics", *SIAM J. Sci. Stat. Comput.*, Vol. 3, 1982, pp. 250-259.

[58] Ta'asan, S., and Nark, D.M., "An Absorbing Buffer Zone Technique of Acoustic Wave Propagation", AIAA Paper 95-0164, January 1995.

[59] Colonius, T., Lele, S.K., and Moin, P., "Boundary Conditions for Direct Computation of Aerodynamic Sound Generation", *AIAA Journal*, Vol. 31, 1993, pp. 1574-1582.

[60] Freund, J.B., "Proposed Inflow/Outflow Boundary Condition for Direct Computation of Aerodynamic Sound", *AIAA Journal*, Vol. 35, 1997, pp. 740-742.

[61] Hu, F.Q., "On Absorbing Boundary Conditions for Linearized Euler Equations by a Perfectly Matched Layer", *J. Comp. Physics*, Vol. 129, 1996, pp. 201-219.

[62] Hayder, M.E., Hu, F.Q., and Hussaini, M.Y., "Towards Perfectly Absorbing Boundary Conditions for Euler Equations", ICASE Report 97-25, May 1997.

[63] Tam, C.K.W., "Advances in Numerical Boundary Conditions for Computational Aeroacoustics", AIAA Paper 97-1774, June 1997.

[64] Hesthaven, J.S., "On the Analysis and Construction of Perfectly Matched Layers for the Linearized Euler Equations", *J. Comp. Physics*, submitted.

[65] Tam, C.K.W., "Computational Aeroacoustics: Methods and Applications", AIAA Short Course Notes, May 1997.

[66] Don, W.S., and Gottlieb, D., "High Order Methods for Complicated Flows Interacting with Shock Waves", AIAA Paper 97-0538, January 1997.

[67] Atkins, H.L., "Continued Development of the Discontinuous Galerkin Method for Computational Aeroacoustic Applications", AIAA Paper 97-1581, May 1997.

[68] Gordon, W.J., and Hall, C.A., "Construction of curvilinear co-ordinate systems and applications to mesh generation", *Int. J. Num. Meth. Eng.*, Vol. 7, 1973, pp. 461-477.

[69] Fletcher, C.A.J., *Computational Techniques for Fluid Dynamics*, 2nd ed., Springer-Verlag, 1991.

[70] Kosloff, D., and Tal-Ezer, H., "A Modified Chebyshev Pseudospectral Method with an $O(N^{-1})$ Time Step Restriction", *J. Comp. Physics*, Vol. 104, 1993, pp. 457-469.

[71] Bismuti, P.J., and Kopriva, D.A., "Solution of Acoustic Scattering Problems by a Staggered-Grid Spectral Domain Decomposition Method", NASA CP-3352, Second Computational Aeroacoustics Workshop on Benchmark Problems, November 1996.

[72] Dong, T.Z., Shih, S.H., Mankbadi, R.R., and Povinelli, L.A., "A Numerical Study of Duct Geometry Effect on Radiation of Engine Internal Noise", AIAA Paper 97-1604, May 1997.

[73] Tam, C.K.W., and Hardin, J.C., Editors, "Second Computational Aeroacoustics (CAA) Workshop on Benchmark Problems", NASA CP-3352, November 1996.

[74] Rice, E.J., "Multimodal Far-Field Acoustic Pattern Using Mode Cutoff Ratio", *AIAA Journal*, Vol. 16, 1978, pp. 906-911.

[75] Preisser, J.S., Silcox, R.J., Eversman, W., and Parrett, A.V., "A Flight Study of Tone Radiation Patterns Generated by Inlet Rods in a Small Turbofan Engine", AIAA Paper 84-0499, January 1984.

[76] Malbéqui, P., Glandier, C., and Reynier, C., "Sound Propagation and Radiation in a Curved Duct", *AIAA Journal*, Vol. 34, 1996, pp. 1778-1784.

[77] Troutt, T.R., and McLaughlin, D.K., "Experiments on the flow and acoustic properties of a moderate-Reynolds-number supersonic jet", *J. Fluid Mech.*, Vol. 116, 1982, pp. 123-156.

[78] Tam, C.K.W., and Burton, D.E., "Sound generated by instability waves of supersonic flows. Part 2. Axisymmetric jets", *J. Fluid Mech.*, Vol. 138, 1984, pp. 273-295.

[79] Tam, C.K.W., and Chen, P., "Turbulent Mixing Noise from Supersonic Jets", *AIAA Journal*, Vol. 32, 1994, pp. 1774-1780.

[80] Drazin, P.G., and Reid, W.H., *Hydrodynamic Stability*, Cambridge Univ. Press, 1981.

[81] Mankbadi, R.R., Hixon, R., Shih, S.-H., and Povinelli, L.A., "Use of Linearized Euler Equations for Supersonic Jet Noise Prediction", *AIAA Journal*, Vol. 36, 1998, pp. 140-147.

[82] Baysal, O., and Idres, M.M., "Euler Computation of Jet Noise: Instability Generated Waves and Their Propagation" AIAA Paper 98-2262, June 1998.

[83] Fenno, C.C., Bayliss, A., and Maestello, L., "Interaction of sound from supersonic jets with nearby structures", ICASE Report 97-31, June 1997.

[84] Tadmor, E., "Convergence of spectral methods for nonlinear conservation laws", *SIAM J. Numer. Analysis*, Vol. 26, 1989, pp. 30-44.

[85] Tadmor, E., "Shock capturing by the spectral viscosity method", *Computer Methods Appl. Mech. Eng.*, Vol. 80, 1990, pp. 197-208.

[86] Gottlieb, D., and Shu, C.-W., "On the Gibbs phenomenon and its resolution", *SIAM Review*, Vol. 39, 1997, pp. 644-668.

[87] Ma, H., "Chebyshev-Legendre super spectral viscosity method for nonlinear conservation laws", *SIAM J. Numer. Analysis*, Vol. 35, 1998, pp. 893-908.