

9 391438X

NUMERICAL TECHNIQUES FOR  
COMPUTATIONAL AEROACOUSTICS

GEORGI STEFANOV DJAMBAZOV

A thesis submitted in partial fulfilment of the  
requirements of the University of Greenwich  
for the Degree of Doctor of Philosophy

September 1998



## Abstract

The problem of aerodynamic noise is considered following the Computational Aeroacoustics approach which is based on direct numerical simulation of the sound field.

In the region of sound generation, the unsteady airflow is computed separately from the sound using Computational Fluid Dynamics (CFD) codes. Overlapping this region and extending further away is the acoustic domain where the linearised Euler equations governing the sound propagation in moving medium are solved numerically.

After considering a finite volume technique of improved accuracy, preference is given to an optimised higher order finite difference scheme which is validated against analytical solutions of the governing equations. A coupling technique of two different CFD codes with the acoustic solver is demonstrated to capture the mechanism of sound generation by vortices hitting solid objects in the flow. Sub-grid turbulence and its effect on sound generation has not been considered in this thesis.

The contribution made to the knowledge of Computational Aeroacoustics can be summarised in the following: 1) Extending the order of accuracy of the staggered leap-frog method for the linearised Euler equations in both finite volume and finite difference formulations; 2) Heuristically determined optimal coefficients for the staggered dispersion relation preserving scheme; 3) A solution procedure for the linearised Euler equations involving mirroring at solid boundaries which combines the flexibility of the finite volume method with the higher accuracy of the finite difference schemes; 4) A method for identifying the sound sources in the CFD solution at solid walls and an expansion technique for sound sources inside the flow; 5) Better understanding of the three-level structure of the motions in air: mean flow, flow perturbations, and acoustic waves. It can be used, together with detailed simulation results, in the search for ways of reducing the aerodynamic noise generated by propellers, jets, wind turbines, tunnel exits, and wind-streamed buildings.

## Declaration

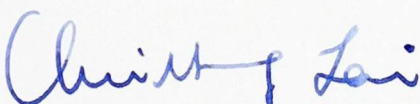
*I certify that this work has not been accepted in substance for any degree, and is not concurrently submitted for any degree other than that of Doctor of Philosophy (PhD) of the University of Greenwich. I also declare that this work is the result of my own investigations except where otherwise stated.*

Student:



Georgi Djambazov

Supervisor:



Dr Choi-Hong Lai

## Acknowledgements

The author wishes to express his gratitude to the University of Greenwich who are sponsoring this study, and especially to Professor Mark Cross who has initiated this scholarship.

I would like to thank my supervisors Dr Choi-Hong Lai, and Professor Koulis Pericleous, for their concern and guidance at the various stages of the research process.

And last but not least, I thank my wife Veronica for her understanding, patience, care, and support.

Georgi Djambazov

# Nomenclature

$A$	Cell face area; wave amplitude; quadrature coefficient
$a_j$	Spatial derivation scheme coefficient
$b_j$	Temporal integration scheme coefficient
$c$	Sound speed
$c_0$	Sound speed in uniform undisturbed medium
$d_j$	Artificial damping coefficient
$f$	Function; cell face
$f_I, f_T$	Flux across cell face
$f_1, f_2, f_3$	Perturbation force components
$\bar{f}_i$	Average momentum forcing terms
$F_i$	Linearised Euler source of momentum
$M$	Mach number
$N_w$	Number of points per wavelength
$p$	Pressure; pressure perturbation
$\bar{p}$	Mean pressure; CFD computed pressure
$p_{ij}$	Fluid stress tensor
$Q$	Source of mass

$S$	Kirchhoff's surface; Linearised Euler source term
$t$	Time
$T_{ij}$	Lighthill's stress tensor
$u, v$	Acoustic velocity variables; Bezier curve parameters
$u_0$	Uniform mean velocity
$v_1, v_2, v_3$	Cartesian velocity components
$\bar{v}_i$	Averaged velocity components; CFD computed velocity components
$x_1, x_2, x_3$	Cartesian coordinates with index summation
$x, y, z$	Cartesian coordinates
$\alpha_1, \alpha_2, \alpha_3$	Plane wave direction angles
$\gamma$	Isentropic exponent
$\Delta t$	Time step
$\Delta x, \Delta y$	Regular Cartesian grid spacing
$\lambda$	Wavelength
$\rho$	Density; density perturbation
$\bar{\rho}$	Averaged density; CFD computed density
$\sigma$	Courant number

# List of Figures

1.1	Conventional CFD solution of test problem . . . . .	11
2.1	Interaction between CFD and CAA codes . . . . .	20
2.2	Domains for the numerical simulation of aerodynamic sound . . . . .	22
3.1	Computational grid and radiating boundary interpolation . . . . .	26
3.2	The fully-explicit leap-frog scheme. 1D propagation of a single pulse is shown at regular time intervals. . . . .	27
3.3	The semi-implicit scheme. Test pulse after travelling 100 wavelengths.	28
3.4	Solid boundary - stepwise representation showing the effects of mesh refinement . . . . .	29
3.5	Sketch of Problem 1 . . . . .	30
3.6	Computational domain and instantaneous pressure field for Problem 1	31
3.7	Contours of instantaneous pressure showing interference patterns caused by the interaction of emitted waves from the source and reflected waves from the cylinder. . . . .	32
3.8	Comparison between numerical and benchmark solutions for Problem 1, at $r=5$ . . . . .	33
3.9	Comparison between numerical and benchmark solutions for Problem 1, at $r=7$ . . . . .	33
3.10	Comparison between numerical and analytic solution, and the effect of mesh density for Problem 2 . . . . .	34
3.11	Comparison of different numerical schemes for pure convection . . . . .	36
3.12	Acoustic perturbation travel in a boundary layer (solid lines) and in a uniform mean flow (broken lines) . . . . .	37
3.13	Test pulse travel and numerical error of the two-level scheme . . . . .	39
3.14	Notation for the 5-point finite volume schemes . . . . .	40

3.15	Finite volume test solution at Mach number 0.2 . . . . .	42
3.16	Applicability of the Refined Finite Volume Scheme . . . . .	43
4.1	Solution of the parametric equation for $a_1$ . . . . .	51
4.2	Parametric solutions for $b_0$ . . . . .	53
4.3	Mirroring of cell-centred values along $x$ . . . . .	55
4.4	Updated test solution from Figure 3.15 . . . . .	58
4.5	Multiple reflections in 1D domain . . . . .	59
4.6	Resonant reflections of sine pulse . . . . .	59
4.7	Acoustic scattering test . . . . .	61
4.8	Vertical and horizontal section of model domain . . . . .	62
4.9	Signals of the test simulation . . . . .	63
4.10	Pressure field at symmetry plane . . . . .	64
5.1	CFD solution at source of sound . . . . .	67
5.2	Combined solution of test problem . . . . .	71
5.3	Hydrodynamic perturbations and blade (left, scale: 1 m/s to 0.1 m), and acoustic pressure contours (right, spacing: 6 Pa) . . . . .	72
5.4	Acoustic signal in the specified cells above and below the centre of the blade. . . . .	73
5.5	Vortex convection with upwind (left) and QUICK (right) schemes . .	74
5.6	Aerofoil and mesh for inviscid flow computation . . . . .	78
5.7	Steady mean flow solution . . . . .	80
5.8	Flow and acoustic perturbation fields (showing superposition of mean- flow and acoustic domains) . . . . .	81
5.9	Flow and acoustic perturbation fields, continued (showing superposi- tion of flow and acoustic perturbations) . . . . .	82
5.10	Representation of solid boundaries . . . . .	83
5.11	Analysis of unsteady airflow . . . . .	86
5.12	Analytic and preliminary CFD solutions of test problem . . . . .	90
5.13	Acoustic expansion solution of test problem . . . . .	90
5.14	Vortex generation and acoustic expansion pressure contours (positive - solid lines, negative - dashed lines, spacing: 20 Pa). Velocity vector scale: 3 m/s to 0.2 m. Vertical dashed line marks vortex generation source patch. . . . .	92



# Contents

Acknowledgements . . . . .	ii
Nomenclature . . . . .	iii
List of Figures . . . . .	iv
<b>1 Introduction</b>	<b>1</b>
1.1 Aeroacoustics . . . . .	1
1.2 Acoustic Analogies . . . . .	2
1.2.1 The Equations of Sound in Air . . . . .	2
1.2.2 Lighthill's Analogy . . . . .	4
1.2.3 Kirchhoff's Method . . . . .	5
1.2.4 Extended Acoustic Analogies . . . . .	6
1.3 Direct Simulation of Sound . . . . .	8
1.3.1 General Review . . . . .	8
1.3.2 The need for a special approach to sound . . . . .	11
1.3.3 Numerical schemes and boundary conditions . . . . .	13
1.4 Objectives . . . . .	15
1.5 Thesis Layout . . . . .	15
<b>2 Domain Decomposition</b>	<b>17</b>
2.1 Decomposition of the Variables . . . . .	17
2.2 Near Field and Far Field . . . . .	21
<b>3 Finite Volume Algorithms</b>	<b>24</b>
3.1 The Case of Uniform Mean Flow . . . . .	24
3.1.1 Acoustic scattering benchmark problems . . . . .	29
3.2 Linearised Euler Solutions . . . . .	35
3.2.1 Three-point schemes . . . . .	35

3.2.2	Five-point schemes . . . . .	38
3.3	Accuracy and Efficiency Considerations . . . . .	43
<b>4</b>	<b>The Acoustic Module</b>	<b>45</b>
4.1	Numerical Schemes . . . . .	45
4.1.1	DRP schemes . . . . .	46
4.1.2	Staggered DRP-type schemes . . . . .	49
4.2	Boundary Conditions . . . . .	53
4.2.1	Solid boundaries . . . . .	53
4.2.2	Radiating boundaries . . . . .	55
4.3	Validation Tests . . . . .	57
4.4	A 3D Application: Resonant Cavity . . . . .	62
<b>5</b>	<b>Coupling CFD with Acoustics</b>	<b>66</b>
5.1	Aeroacoustic Sources on Solid Surfaces . . . . .	66
5.2	Coupling with the Structured Finite Volume Code PHOENICS . . . . .	70
5.3	Coupling with the Unstructured Mesh Finite Volume Code PHYSICA	73
5.3.1	Second order schemes . . . . .	74
5.3.2	Aerofoil geometry specification . . . . .	76
5.3.3	Simulation results . . . . .	79
5.4	Sound Generation Away from Solid Bodies . . . . .	84
5.5	Acoustic Expansion Technique . . . . .	86
<b>6</b>	<b>Conclusions &amp; Further Development</b>	<b>93</b>
	<b>Bibliography</b>	<b>95</b>
	<b>Appendix</b>	<b>102</b>
<b>A</b>	<b>Pseudo-Code of the Acoustic Module</b>	<b>102</b>
<b>B</b>	<b>Pseudo-Code of the PHYSICA Coupling</b>	<b>115</b>

# Chapter 1

## Introduction

### 1.1 Aeroacoustics

Sound may be good and bad. Most people enjoy music but hardly anyone likes the noise of a jet passing overhead. The work of many scientists and engineers is devoted to the improvement of the good sounds and to the reduction of the bad noise. (By definition, noise is a chaotic mixture of sounds.)

As people hear those sounds which propagate in air, three main types of sound sources can be defined: (a) vibrating solid objects or liquid surfaces, (b) resonating cavities full of air, and (c) oscillating vortex structures or pressure fluctuations in airflow.

Aeroacoustics, the science of aerodynamic sound, deals with the third type of sound sources, as well as with the propagation of the resulting acoustic waves through non-uniformly moving media.

This scientific discipline was initiated in the early years of jet aircraft (Lighthill, 1952) in response to the demand of finding ways to reduce the noise produced by jet engines. The theory was successful in quantifying the dependence of the noise levels on the jet velocity (Lighthill, 1954), and it found its application in the modern by-pass jet engines which make considerably less noise than the old ones, due to the reduced jet velocity at the same flow rate.

The rapid advance of computational power in the early nineties triggered a second ‘golden age’ of aeroacoustics. A separate branch evolved, Computational Aeroacoustics, which deals with the direct simulation of acoustic fields generated by flow and of the interaction of acoustic fields with flow. The phrase “direct simulation” is under-

stood to imply that results are obtained by computation and that the methodology proceeds directly from fundamental physical principles without reliance on empirical results or heuristic conjectures (Pierce, 1993).

It should be pointed out that this research is not concerned with sound propagation in still media which can be computed successfully by other methods using commercial codes, e.g. (SYSNOISE, 1995).

The following sections of this chapter provide an overview of the two main approaches to aerodynamic sound: theoretical and numerical, and define the objectives of this research.

## 1.2 Acoustic Analogies

In the early 1950's the only possible way of studying the mechanism of aerodynamic sound generation was by comparing theoretical predictions with experimental results. The most appropriate available theory to start with was the linear theory of classical acoustics (Lighthill, 1978).

### 1.2.1 The Equations of Sound in Air

Sound propagation in air is a particular case of fluid motion which is associated with compression waves travelling away from a given source. The air particles oscillate with each passing wave, but the magnitude of the particle velocity of these oscillations is many times less than any common airflow we observe (Turner and Pretlove, 1991). (At the threshold of pain for the human ear, the particle velocity amplitude is not more than 0.2 m/s.) On the other hand, the rate of change of the same particle velocity (and of the other quantities characteristic of the fluid state) cannot be considered small at all. All this means that the equations of fluid motion can be simplified into a particular form which describes adequately the nature of the acoustic motion.

If  $\rho$  is the fluid density,  $v_1, v_2, v_3$  are the Cartesian components of the fluid velocity, and the index summation convention is used for any suffix repeated in a single term, the basic equations of continuity and momentum conservation can be written as (Lighthill, 1952):

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j} (\rho v_j) = 0 \quad (1.1)$$

$$\frac{\partial}{\partial t}(\rho v_i) + \frac{\partial}{\partial x_j}(\rho v_i v_j + p_{ij}) = 0. \quad (1.2)$$

Here  $p_{ij}$  is the force in the  $x_i$  direction acting on a portion of fluid, per unit area with inward normal in the  $x_j$  direction. It includes pressure, viscous stresses, and any external forces (like gravity) acting on the fluid.

The linear acoustic theory studies the propagation of sound in a uniform continuous medium *at rest*. Due to the very small particle velocity, the derivatives of the term  $\rho v_i v_j$  in (1.2) can be neglected because, in the expanded form, each of them is multiplied by a small quantity. For the same reason any viscous forces are negligibly small, and  $p_{ij}$  becomes a simple  $p$  for pressure acting only in the normal direction on the elementary fluid volume faces. Also, due to the rapid speed of wave propagation, isentropic conditions can be assumed throughout the uniform medium which provide a simple pressure-density relation ( $p = K\rho^\gamma$ ,  $K = \text{const}$ ,  $\gamma = 1.4$  in air) and an expression for the speed of sound  $c_0$  (Pierce, 1981):

$$c_0^2 = \frac{\partial p}{\partial \rho} = \frac{\partial}{\partial \rho}(K\rho^\gamma) = \gamma \frac{p}{\rho} \quad (1.3)$$

Thus, in the absence of external forces, the simplified momentum equation becomes

$$\frac{\partial}{\partial t}(\rho v_i) + c_0^2 \frac{\partial \rho}{\partial x_i} = 0. \quad (1.4)$$

Without any external sources of matter, equations (1.1) and (1.4) govern the acoustic motion in air. By cross differentiation and simplification these equations can be reduced to the wave equation:

$$\frac{\partial^2 \rho}{\partial t^2} - c_0^2 \sum_{j=1}^3 \frac{\partial^2 \rho}{\partial x_j^2} = \frac{\partial^2 \rho}{\partial t^2} - c_0^2 \nabla^2 \rho = 0. \quad (1.5)$$

It describes the sound propagation in *still* air as well as the propagation of electromagnetic waves which are of a totally different physical origin. As it will be seen in Chapter 2, different equations govern the sound field in a *moving* medium, and the methods for studying electro-magnetic waves cannot be applied directly to sound.

The wave equation supports solutions in the form of linear combinations of plane and spherical waves that travel with the speed of sound  $c_0$ . These solutions can represent complex acoustic fields that result from the interactions of primary and secondary (reflected by solid boundaries) sound waves.

### 1.2.2 Lighthill's Analogy

Sir James Lighthill's (1952) idea was to rearrange the *exact* equation of momentum (1.2) into the form (1.6) which describes the propagation of sound in a uniform medium at rest due to externally applied fluctuating stresses:

$$\frac{\partial}{\partial t}(\rho v_i) + c_0^2 \frac{\partial \rho}{\partial x_i} = -\frac{\partial T_{ij}}{\partial x_j} \quad (1.6)$$

$$T_{ij} = \rho v_i v_j + p_{ij} - c_0^2 \rho \delta_{ij} \quad (1.7)$$

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j. \end{cases} \quad (1.8)$$

The set of equations (1.1), (1.6) and (1.7), which can be combined into an inhomogeneous wave equation, is known as Lighthill's acoustic analogy. It makes use of the solutions of the well-studied wave equation (1.5) for the investigation of aerodynamic sound.

Three different ways can be distinguished (Lighthill, 1952) of causing kinetic energy of the flow to be converted into acoustic energy: 1) *sources* of mass that emerges at a variable rate in a fixed region of space; 2) *dipoles*, which force the momentum in a fixed region of space to fluctuate; 3) *quadrupoles*, characterised by varying rates of momentum flux across fixed surfaces. It can be shown (Lighthill, 1978), that the dipole momentum fluctuation is equivalent to varying rates of mass flux across fixed surfaces. In this sense, the quadruple representation is a logical continuation of the representation of the other two simpler mechanisms of sound generation. Volume integral formulations have been derived in the three cases of *distributed* sources, dipoles, and quadrupoles respectively (Lighthill, 1952). They represent exact analytical solutions of the wave equation (1.5) with known non-zero right-hand side.

The acoustic analogy can be used in two ways.

**First**, assuming appropriate estimates for the right-hand side values, theoretical predictions for the noise generated by turbulence can be made which show satisfactory agreement with experimental results (Lighthill, 1954). This is done after careful analysis of the nature of the different types of acoustic sources, and establishing the significance of the *quadrupole* radiation (Lighthill, 1952) that appears to be the dominant mechanism of aerodynamic sound generation away from solid bodies.

This theoretical approach has not lost its significance even in the present age

of powerful computers, because the numerical methods used for the computation of acoustic problems very often introduce false “noise” in the solution (Crighton, 1993). Theoretical studies of various aerodynamic noise problems are conducted at the Engineering Department of Cambridge University. Thermo-acoustic oscillations, under-expanded jets impinging on plane surfaces, boundary layers, and even respiratory biomechanics phenomena have been considered (Ffowcs Williams and Dowling, 1996).

**Second**, in recent years, numerical data resulting from Computational Fluid Dynamics (CFD) simulations of airflow can be used to calculate the term  $T_{ij}$  in the region of sound generation, and then the acoustic ‘far field’ which extends outside the flow computational domain can be determined from the wave equation.

The second approach is used nowadays by a number of research groups (Sarkar and Hussaini, 1993; Zhang et al., 1995) and even by Rolls Royce, manufacturers of jet engines. Evaluation of a volume integral is needed to determine the sound field resulting from a known distribution of quadrupole sources. In the general case when the sources depend on the sound field, they are unknown, and an integral equation has to be solved.

With aerodynamic noise generated at solid surfaces the quadrupole sources can be neglected in some cases (low Mach number wakes) when the *dipole* sources at the solid boundary become dominant. This is described by Curle’s equation (Curle, 1955) which is used now by other scientists (Kato et al., 1995; Yokono and Fujita, 1995) in their research. The oscillating surface pressure obtained from Large-Eddy Simulation has also been combined with the Lighthill-Curle acoustic analogy to predict the far-field sound (Spyropoulos and Holmes, 1997).

### 1.2.3 Kirchhoff’s Method

With sound sources contained in a certain region, and if the mean flow can be assumed uniform (or zero) outside of this region, *surface* integrals can be formulated which describe the outside acoustic field.

Kirchhoff’s surface formula was first published in 1882, and has been primarily used with light and other electro-magnetic problems. According to this theorem, any quantity  $\Phi(\mathbf{x}, t)$  which follows the wave equation outside a given surface  $S$  is defined by its values, spatial and temporal derivatives at an earlier time ( $\tau = t - r/c_0$ ) on

the surface  $S$  (Pierce, 1981):

$$4\pi\Phi(\mathbf{x}, t) = \int_S \left( \frac{\cos\theta}{r^2}\Phi - \frac{1}{r}\frac{\partial\Phi}{\partial n} + \frac{\cos\theta}{rc_0}\frac{\partial\Phi}{\partial t} \right)_\tau dS. \quad (1.9)$$

Here the vector  $\mathbf{x}$  defines the observation point,  $r$  is the distance between the observation point and the surface element  $dS$ ,  $n$  is the local normal direction to the surface, and  $\theta$  is the angle between  $n$  and the line connecting  $dS$  with the observation point. The interval of time  $r/c_0$  which the wave takes to reach the observation point is called the retarded time, and the subscript  $\tau$  indicates that all the quantities in the brackets are evaluated at this particular moment of time which is different for the different points on the surface.

If a CFD simulation is performed in the inner region (with suitable boundary conditions that allow the sound waves to leave the computational domain without reflection) the surface  $S$  can be defined near the boundaries of the CFD domain, and from the stored data, the necessary derivatives can be calculated. Then evaluation of the surface integral has to be carried out for each observation point of interest.

There are contradictory requirements for the location of the Kirchhoff surface: it should be far enough to contain all the sound sources and mean-flow non-uniformity, and still close enough to form a reasonably sized computational domain for the CFD simulation.

Extensions of the formula exist for moving and deformable surfaces (Ffowcs Williams and Hawkings, 1969; Farassat and Myers, 1988) which make the method very suitable for studying blade-vortex interactions and the noise produced by helicopter rotors (Lyrantzis, 1993). Recent efforts show that surface integral methods can also be applied to jet noise (Lyrantzis and Mankbadi, 1996; Pilon and Lyrantzis, 1997).

#### 1.2.4 Extended Acoustic Analogies

When the sound waves propagate in a moving medium rather than in still air, the assumptions that led to equation (1.4) are no longer valid. As it will be seen in the following chapters, the left-hand side of this equation has to be complemented with an additional term — the convection term — in order to obtain an adequate mathematical model of the acoustic field in non-uniformly moving media. (Of course, in a uniform background flow, equation (1.4) still holds with respect to a relative



frame of reference moving with the background velocity. The problem is that in many cases, e.g. jets, no background flow can be defined.) For this reason, serious difficulties arise with the application of Lighthill's analogy to CFD numerical results, especially at high Mach numbers ( $M = v_0/c_0$ ,  $v_0$  - average velocity characteristic of the flow).

Being an analogy between the real density fluctuations in a flow and "the small amplitude density fluctuations that would result from an appropriate quadrupole source distribution in a fictitious non-moving medium" (Goldstein and Mankbadi, 1993), Lighthill's approach actually involves in the source term of equation (1.6) acoustic *propagation* effects which "preclude the possibility of localising the source term". This means that the CFD simulation from which this source term will be calculated cannot be confined into a small domain and will become prohibitively expensive.

Several attempts have been made to transfer the effects of the interaction between the sound field and the mean flow (mainly convection and refraction) from the source term to the wave-operator part (the left-hand side) of the equation (1.6 and 1.5). This can be done by suitable rearrangement of the Navier-Stokes equations (1.2) in an operator form that reduces to the wave equation at large distances from the sound source and has a localised source term on right-hand side (Phillips, 1960; Lilley, 1974; Powell, 1964; Howe, 1975).

It appears that "the price which must be paid for including the convection and refraction effects in the wave-operator part of the equation is a great increase in the complexity of the solutions", and that "only limited solutions of Lilley's and Phillips' equations have been found" (Goldstein, 1993).

Nevertheless, there are research groups that use these equations to extract the sound from data obtained by direct numerical simulation (DNS) of compressible flows. Phillips' analogy has been used at QMW, London (Avital et al., 1998), and Lilley's equation - at Stanford University, California (Colonius et al., 1995a). The Stanford group has found that the form of the source term (in Lilley's analogy) proposed by Goldstein (Goldstein, 1984) gives good agreement with the DNS that included a portion of the acoustic field which could be compared with the acoustic analogy (Colonius et al., 1995b).

## 1.3 Direct Simulation of Sound

With the rapid increase of available computing power in the early 1990's it is most natural that scientists started looking at the possibilities of full numerical representation of the acoustic field that arises in unsteady airflow. An excellent starting basis for this is the tremendous progress that Computational Fluid Dynamics (CFD) has made over the past 20 years which made it possible to carry out aerodynamic and aircraft design from a computer terminal.

It should be pointed out from the very beginning that Computational Aeroacoustics (CAA) is *not* (and is not going to be soon) concerned with direct numerical simulation (DNS) of turbulent flow. DNS is prohibitively expensive and can only be used for the validation of *turbulent models* that are incorporated in general-purpose CFD codes. However, there is a less demanding option called Large-Eddy Simulation (LES) that may become affordable in the near future. LES can accurately capture the large scales of motion in turbulent flows using appropriate computational grids and special turbulent models for the sub-grid scales. But CAA is not the same as LES.

The direct numerical representation of the acoustic waves needs considerably less computing power than LES (not to mention DNS). This is due to the fact that aerodynamic noise is produced by vortex structures (eddies) in the flow that are usually several times smaller in size than the corresponding acoustic wavelengths ( $\lambda = c_0/f$ ,  $f$  - frequency). Therefore, compared to the flow, sound can be captured on coarser meshes which lead to reduced usage of computer memory and time.

Numerical simulations of the aerodynamic sound *sources*, however, require full Navier-Stokes solutions on meshes fine enough for the discretisation of the smallest flow structures that may be significant as sources of sound. Performing such simulations at reasonable computational cost means that the flow domain should be made as small as possible - much smaller than the space occupied by the studied acoustic field.

### 1.3.1 General Review

How to use LES (and CFD in general) *wisely* in order to study aerodynamic sound, is one of the subjects of computational aeroacoustics (CAA). The most natural approach is to split the acoustic calculation in two parts, one describing the generation

of sound, the other describing the propagation of sound. Then the CFD simulation is run in a smaller domain that covers only the ‘near field’ - the zone where sound waves are produced. After that the CAA computation can be performed over the whole region of interest to determine how the sound waves propagate in the surrounding medium (non-uniformly moving), how they are reflected by solid objects of complex shape that may be present nearby, and how all these waves interfere with each other. A good example of this approach is the numerical study of jet noise using LES in NASA Lewis Research Center (Shih et al., 1995).

The ‘cheapest’ CFD method which has accumulated a vast amount of experience during the years is solving the Reynolds-Average Navier-Stokes (RANS) equations with a suitable turbulence model. Being designed and validated for many industrial applications, this method performs best with steady flows, i.e. it is an excellent tool for simulating the mean flow in an aeroacoustic problem.

With the mean flow obtained from a separate RANS calculation, the whole perturbation field (including large-scale fluctuations of both acoustic and aerodynamic nature) can be predicted using LES. A special high accuracy solution procedure has been implemented for this purpose at the Aerospace Department of Pennsylvania State University (Morris et al., 1997). It performs best on parallel computers and shows reasonable agreement with experimental results. This approach is likely to produce a good research tool in the near future, but it will be still computationally expensive for practical applications.

Another way of splitting the computation is the separation of variables (Hardin, 1993). A two-step calculation has been proposed (Hardin and Pope, 1994) where the viscous flow is calculated from the time dependent *incompressible* Navier-Stokes equations, and the acoustic fluctuation is obtained from the inviscid compressible Euler equations. Based on the RANS solution  $P(x_i, t)$ , a new variable  $\rho_1$  is defined to represent the hydrodynamic density fluctuations (Hardin and Pope, 1994):

$$\rho_1(x_i, t) = \frac{1}{c^2} [P(x_i, t) - \bar{P}(x_i)] \quad (1.10)$$

where  $\bar{P}(x_i)$  is the time-averaged pressure of the incompressible solution, and the speed of sound  $c$  is defined assuming that the flow is isentropic (2.10).

The fluid variables are then decomposed into mean flow and perturbation components (Hardin and Pope, 1995):

$$\rho = \rho_0 + \rho_1 + \rho' , \quad v_i = U_i(x_i, t) + v'_i , \quad p = P(x_i, t) + p' .$$

Here  $U_i(x_i, t)$  is used to denote the time dependent velocity field of the incompressible RANS solution, and new equations for the perturbation density  $\rho'$ , velocity components  $v'_i$ , and pressure fluctuation  $p'$  are derived by substitution into the equations of fluid motion (1.1) and (1.2). Since  $\rho_0$  is the average density of the fluid at rest, the acoustic component  $\rho'$  represents density fluctuations about the 'corrected' incompressible density  $\rho_0 + \rho_1$  (Hardin and Pope, 1995). The viscous action on the acoustic quantities is neglected, and the resulting Euler equations can be solved on a different grid — there is no longer a need to resolve acoustic wavelengths and dissipating eddy scales on a single mesh.

The main advantage of this technique is that optimum grids, time steps, and numerical schemes can be used with both components of the solution. However, the perturbation velocity is not guaranteed to be of a purely acoustic nature; it may contain flow components due to the difference between the incompressible and the compressible flow field. Therefore, solid boundaries in the simulation have to be represented as smooth rather than stepwise. (In Chapter 4 it will be shown that the most accurate acoustic discretisation schemes require regular grids and, hence, stepwise solid boundaries.) Also, the common difficulty with all aeroacoustic computations remains: high Reynolds number incompressible flow must be solved. The method cannot be used in cases of compressible mean flow, e.g. when the mean velocity is close to the speed of sound.

Since the acoustic motion is a very small perturbation superimposed on the mean flow, the latter can be computed separately by solving the *compressible* Reynolds-Average Navier-Stokes (RANS) equations with a suitable turbulence model. Then, a spectrum of the turbulent perturbations is assumed that has the same average characteristics as those determined within the CFD simulation. The resulting time dependent turbulent field is used to calculate the sources of sound, and finally, the Euler equations are solved numerically only for the acoustic part of the participating variables (Viswanathan and Sankar, 1995).

With the increase of available (and affordable) computing power, the RANS solution can be replaced by LES so that the assumptions based on statistical data can be restricted to the smallest scales of turbulence which remain unresolved. Then the lower frequencies of the sound that are associated with the larger turbulent eddies will be computed directly from first principles which is the goal of Computational Aeroacoustics. The accuracy requirements to the LES numerical schemes in this

case will be less strict than to the LES of the full near field which is presented in (Shih et al., 1995).

The greatest difficulty with the large-eddy approach (LES) is the subgrid-scale turbulence model. *Dynamic* adjustment of its parameters to the resolved large-scale flow conditions has been applied to the benchmark problem of the Aeolian tone produced by a cylinder in a uniform flow (Spyropoulos and Holmes, 1997). The authors show that the 244,980 elements used to compute the high Reynolds number flow around the cylinder may not be enough for the dynamic subgrid model. Fractal-based LES subgrid models with explicit scale similarity laws offer future promise here (Dempsey and Pericleous, 1995).

### 1.3.2 The need for a special approach to sound

Once the task has been set of simulating the sound field directly by computation based on fundamental physical principles, many new questions and issues arise. Can we use an existing CFD code (with a reasonable number of mesh points per

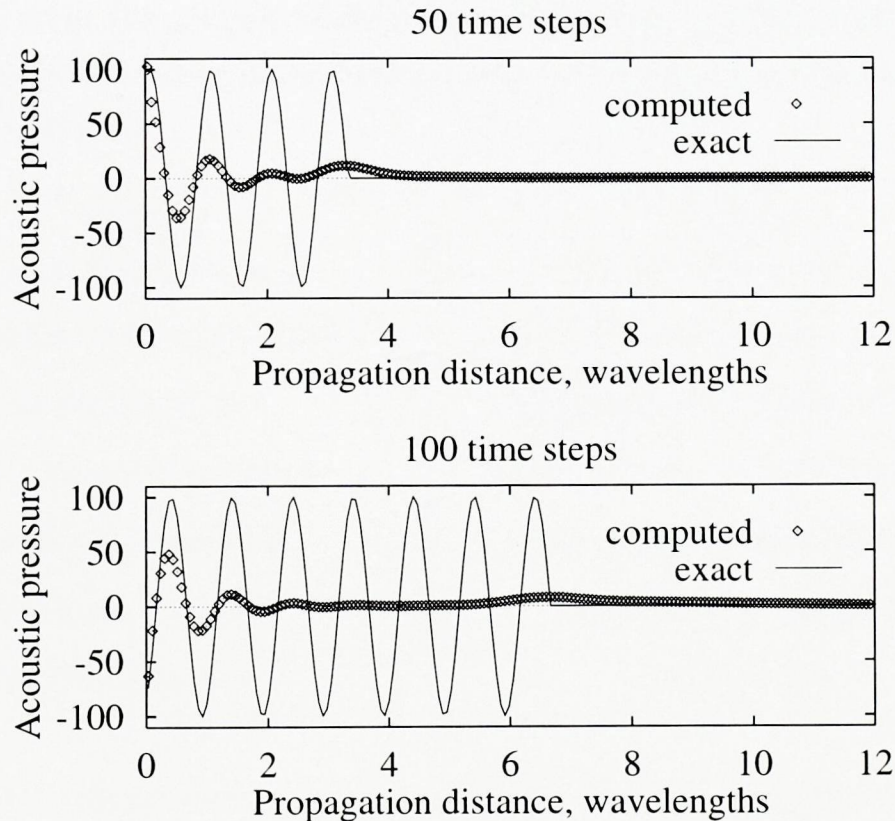


Figure 1.1: Conventional CFD solution of test problem

wavelength) to resolve the acoustic field? The answer to this question is negative as

it had been pointed out and made clear (Tam, 1995) at the time when this research work was started.

This can be illustrated by the simple test of one-dimensional propagation in a tube of sound waves generated by a piston at one end that starts oscillating at time zero. The resulting exact sound field (pressure distribution) can be found by solving the wave equation (1.5) analytically (Turner and Pretlove, 1991). Here the exact solution is compared with the solution computed by a Reynolds-Average Navier-Stokes (RANS) solver (PHOENICS, 1995) with its default numerical scheme (upwind fully implicit). As it can be seen on Figure 1.1, the numerical and the analytic solutions agree only in a very narrow region next to the source at the left end of the domain. Refining of the mesh does not seem to improve the result at all. Even switching to second-order schemes (available for the velocity variables within the same CFD code) produces almost the same output.

When CFD methods were developed, most of the aerodynamic problems considered were the time-independent ones, whereas acoustic phenomena are, by definition, time dependent. Table 1.1 shows how different the requirements for *accuracy* and *efficiency* are with the numerical solutions of the flow and the sound field respectively. All the issues summarised in this table will be explained in greater detail in the following chapters.

<u>CFD</u>	<u>CAA</u>
(Computational Fluid Dynamics)	(Computational Aeroacoustics)
Non-uniform/unstructured grid	Regular Cartesian grid
Fully implicit in time	Explicit/semi-implicit schemes
Finite volume discretization	Higher order finite difference schemes
Smooth solid boundaries	Boundaries may be stepwise
Small-scale structures	Extremely small magnitude
Turbulent models	Inviscid flow

Table 1.1: Requirements for accurate and efficient numerical solutions

Although the sound equations (1.4) are a particular form of the equations governing fluid flow, great differences exist in magnitude, energy and scale of the physical quantities involved. (Acoustic perturbations are typically at least 10 times weaker

than the corresponding hydrodynamic perturbations and a thousand times smaller than the mean flow that carries them. On the other hand acoustic wavelengths are typically several times larger than the corresponding structures in the flow.)

All this means that the algorithmic implementations are so different that they can hardly share any software modules. So, it will be best if a way is found of coupling a flow solver with an acoustic solver in such a manner that each of them does the job that it is best suited for. Mathematically, this means that the physical equations should be decoupled and treated separately.

### 1.3.3 Numerical schemes and boundary conditions

The main differences between CFD and CAA methods are in the numerical schemes for the spatial and the temporal derivatives that are involved in the equations of motion. With CFD algorithms the dominant requirement is for stability under any conditions. That is why mainly linear or even stepwise approximations are implemented (Patankar, 1980).

The acoustic field needs to be approximated by higher-order polynomials in order to obtain accurate solutions of the governing equations. A great number of numerical schemes can be constructed for this purpose. The method that is recommended by leading US scientists in Computational Aeroacoustics is based on the so called Dispersion-Relation-Preserving (DRP) schemes (Tam and Webb, 1993). They are in fact a set of finite difference schemes for the approximation of spatial derivatives and temporal integrations and for the artificial selective damping of under-resolved high frequencies. All the coefficients involved have been optimised with respect to accurate wave propagation over long distances. This is a fully explicit method, and thus can be easily parallelised. It needs a regular Cartesian grid and stepwise representation of solid bodies in the domain.

Apart from its originators (Tam and Webb, 1993), a number of research groups have adopted the DRP approach. It has been applied to sound radiation from the open end of a duct (Dong et al., 1997), and has provided the basic numerical scheme for Large-Eddy Simulations (Morris et al., 1997). Extensions of the method for non-linear wave propagation have also been reported (Baysal et al., 1997).

With the original DRP schemes all variables are stored at the same locations. This leads to the need of defining 'ghost points' into the solid body in order to satisfy both the boundary conditions and the differential equations at the solid wall

(Tam and Dong, 1993).

In three dimensions where solid bodies of irregular shapes are present the ghost points become inconvenient to handle. Apart from a map of the surfaces, a record of the type of difference equation to be used with each ghost point is needed, as well as non-symmetric difference schemes for the points next to the solid boundary. All these make the final code less efficient than it could be.

Other high-order numerical schemes have also been considered for the direct simulation of sound (Sankar et al., 1993; Zingg et al., 1996; Goodrich, 1997). Some have complicated algorithms that are very difficult to implement in three dimensions. Most of these schemes require filtering of the high-frequency components which cannot be resolved numerically. The temporal integration is usually realized by second or fourth order Runge-Kutta schemes. None of them can be computationally more efficient than the DRP schemes, and none of them has any other major advantages to them.

Numerical methods from *structural* acoustics (finite element methods) have also been applied successfully to sound in air (Atkins, 1997). Thanks to their compact form, these methods can be applied near the boundaries of the computational domain without modification, which cannot be done with most high-order methods. They are not fully implicit in time, and hence, they require smaller time steps. On the other hand, a global (although sparse) matrix has to be solved at each time step which is less efficient than the finite difference schemes.

The outer **boundaries** of the computational domain with direct acoustic simulations are somewhat artificial since the natural boundary condition for the acoustic disturbance is zero oscillation at infinity. The numerical boundary conditions have to simulate the outgoing waves without false reflection. Several approaches exist here, and every implementation is specific to the numerical scheme used in the main part of the domain.

There are two main types of boundary conditions that are usually used for the outer boundaries: radiating and absorbing. In both cases the essential requirement is that from the outer boundaries there is no reflection of waves back into the computational domain. (For solid bodies present in the domain, the conditions imposed on their boundaries are referred to as internal conditions, and are also considered separately with every specific discretisation scheme.)

With the *radiating* boundary conditions the outgoing waves from the domain are



simulated as accurately as possible. Such boundary conditions can be constructed on the basis of the theory of characteristics for hyperbolic equations (Hixon et al., 1995; Reitsma et al., 1993), on the basis of the asymptotic solutions of the governing equations (Tam and Webb, 1993), on the basis of the linear stability theory for small perturbations (Hayder and Hagstrom, 1995), or on the basis of a single frequency assumption (Agarwal and Huh, 1995).

*Absorbing* boundary conditions have to ensure complete artificial dissipation of the outgoing waves over a certain layer near the outer boundary of the computational domain. Such methods have been developed with the numerical simulation of electro-magnetic waves and are called ‘perfectly matched layers’. Their application to outgoing acoustic waves is not straightforward in the general (and most important) case of a moving medium. Recent attempts (Hayder et al., 1997; Hesthaven, 1997) show that this approach can also be successful.

## 1.4 Objectives

On the basis of the review and the analysis in the previous sections, the following objectives of this thesis have been identified:

1. Development of an efficient algorithm for the numerical simulation of the *sound field* in three dimensions, given the mean-flow field and the sources of sound. The implementation should be capable of accommodating solid bodies of complex shapes within the computational domain.
2. Definition of suitable *coupling techniques* between the acoustic simulator and existing CFD codes that will allow the sources of sound to be determined from the CFD solution together with the mean flow.
3. Achievement of a better understanding of the mechanisms of aerodynamic sound generation that can serve as a basis for the development of noise prediction methods and for the control of aerodynamic noise.

## 1.5 Thesis Layout

Apart from the introduction (Chapter 1), the thesis is structured in four main chapters (2, 3, 4, and 5), and a conclusions chapter (6) which also describes possible

extensions of the research.

In Chapter 2 the governing equations of the acoustic perturbations are derived, and the computational domains for their numerical solution are defined.

In Chapter 3 a finite volume acoustic technique is presented in two stages: propagation in uniform flow and general numerical solution of the acoustic equations. The applicability of the finite volume method to acoustic problems is tested against analytical solutions, and the need of more accurate procedures is pointed out.

In Chapter 4 the development of an acoustic software module is described which is based on optimised finite difference numerical schemes and staggered computational meshes. Validation tests against exact solutions are presented, and applications to the simulation of complex sound fields are discussed.

In Chapter 5 a coupling technique is presented of the acoustic module with two different CFD codes which allows the aerodynamic sources of sound on solid walls to be identified from the CFD solution and passed to the acoustic simulation. Examples in realistic geometries are included which show the mechanism of sound generation by perturbations of the flow. Aerodynamic sound sources on solid surfaces, as well as inside the fluid volume are considered.

In the final Chapter 6 conclusions are drawn about what has been achieved so far, and ways of extending the software coupling to the general case of the aerodynamic sound problem are discussed.

# Chapter 2

## Domain Decomposition

Domain Decomposition (DD) is a generic technique for solving *large* mathematical and computational problems by obtaining partial solutions of the different subproblems that build up the original problem being considered. The technique ensures that the overall effort of solving the combined subproblems is considerably less than the effort of solving the whole problem in a straightforward way.

The term ‘domain’ is most often used in a general sense and can refer to geometrical, physical or any other type of subdivision.

### 2.1 Decomposition of the Variables

Any fluid motion can be described by the time-dependent three-dimensional fields of the following variables: density, velocity vector and pressure. For each of them an *averaging* procedure can be defined over a part of the domain or over the whole domain depending on the needs of the specific model:

$$\bar{v} = \frac{1}{(b_x - a_x)(b_y - a_y)(b_z - a_z)(b_t - a_t)} \int_{a_x}^{b_x} dx \int_{a_y}^{b_y} dy \int_{a_z}^{b_z} dz \int_{a_t}^{b_t} v(x, y, z, t) dt. \quad (2.1)$$

Here  $v$  stands for any of the above mentioned variables,  $a$  and  $b$  define the boundaries of the averaging domain and are in general functions of the independent spatial variables  $x, y, z$  and time  $t$ , while  $\bar{v}$  is the averaged quantity.

The current value of any variable at any point can be represented as a sum of its local average value and a deviation from this value. The average field is also called ‘mean field’ or ‘mean flow’, and when the deviations are small, they are called ‘perturbations’.

For convenience, the following notation is assumed: the averaged quantities are over-lined, the perturbations have no special mark, and the vector components along the three spatial axes are denoted by subscripts. Thus  $\bar{p} + p$  is the total pressure at a point in space and time,  $\bar{\rho} + \rho$  is the density, and  $\bar{v}_i + v_i$ ,  $i = 1 \dots 3$ , is the fluid velocity.

With the equations, Einstein's summation convention is used: any subscript repeated in a single term defines a sum of this term with the subscript varying from 1 to 3.

The two main equations of fluid flow are the equation of continuity (2.2) and the Navier-Stokes equations of motion (2.3). With compressible flows they are supplemented by the energy equation and the ideal gas law, so that all the unknown variables can be defined.

$$\frac{\partial}{\partial t} (\bar{\rho} + \rho) + \frac{\partial}{\partial x_j} [(\bar{\rho} + \rho)(\bar{v}_j + v_j)] = 0 \quad (2.2)$$

$$(\bar{\rho} + \rho) \left[ \frac{\partial(\bar{v}_i + v_i)}{\partial t} + (\bar{v}_j + v_j) \frac{\partial(\bar{v}_i + v_i)}{\partial x_j} \right] + \frac{\partial(\bar{p} + p)}{\partial x_i} = \bar{f}_i + f_i \quad (2.3)$$

Here the terms  $\bar{f}_i$  and  $f_i$  account for both the internal fluid friction (viscous forces) and external forces in the mean flow and the perturbations respectively.

As it was pointed out in the previous chapter, extremely large differences exist in the relative magnitude of the mean flow and the acoustic perturbations. For a typical aeroacoustic problem, terms containing a perturbation quantity are very small compared to the rest; terms with derivatives of perturbation quantities, however, cannot be considered small. This suggests to rearrange the terms of the governing equations in a suitable way so that all the significant terms are grouped together.

When the brackets in (2.2) and (2.3) are expanded many new terms appear. To simplify the analysis, the equations of continuity (2.4) and momentum (2.5) of the mean flow are *subtracted* from equations (2.2) and (2.3).

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_j} (\bar{\rho} \bar{v}_j) = 0 \quad (2.4)$$

$$\bar{\rho} \left( \frac{\partial \bar{v}_i}{\partial t} + \bar{v}_j \frac{\partial \bar{v}_i}{\partial x_j} \right) + \frac{\partial \bar{p}}{\partial x_i} = \bar{f}_i \quad (2.5)$$

From the remaining terms all the small ones are moved to the right-hand side, and

the following set of equations is obtained:

$$\frac{\partial \rho}{\partial t} + \bar{v}_j \frac{\partial \rho}{\partial x_j} + \bar{\rho} \frac{\partial v_j}{\partial x_j} = Q \quad (2.6)$$

$$\bar{\rho} \left( \frac{\partial v_i}{\partial t} + \bar{v}_j \frac{\partial v_i}{\partial x_j} \right) + \frac{\partial p}{\partial x_i} = F'_i \quad (2.7)$$

where

$$Q = Q_{ext} - \left[ \rho \frac{\partial}{\partial x_j} (\bar{v}_j + v_j) + v_j \frac{\partial}{\partial x_j} (\bar{\rho} + \rho) \right] \quad (2.8)$$

$$F'_i = f_i - \rho \frac{\partial}{\partial t} (\bar{v}_i + v_i) - [\bar{\rho} v_j + \rho (\bar{v}_j + v_j)] \frac{\partial}{\partial x_j} (\bar{v}_i + v_i). \quad (2.9)$$

The term  $Q_{ext}$  here contains any external sources of mass that may be present in the volume of air studied such as vibrating solid bodies or surfaces. In a similar way, all external volume forces acting on the fluid are contained in the terms  $f_i$ . Apart from these, it can be seen that the *source terms* of (2.6) and (2.7)  $Q$  and  $F'_i$  are comprised only of terms that have a small perturbation quantity as a factor which is an indicator that they are very small compared to the terms on the left-hand side.

Sound propagation is hardly affected by viscosity (that is why noise is so difficult to suppress). This means that in the absence of external forces, the term  $f_i$  which comes from the Navier-Stokes equations is small and can take its place on the right-hand side of the momentum equation (2.7).

Having small right-hand sides of the equations can be exploited in two ways. First, in the absence of external sources and forces, the small terms can be neglected, and the right-hand sides become zeroes. Second, where non-linear wave propagation may occur (this phenomenon known as steepening of the wave is observed with very high magnitudes - many times higher than what the human ear can bear) the right-hand sides can be computed *iteratively* with the unknown quantities taken from the previous iteration at each step of the solution procedure.

Another consequence of the fact that the sound field is described by the small perturbation quantities  $\rho$  and  $p$  is the possibility of assuming *isentropic* conditions with the acoustic motion in air. The adiabatic nature of the physical process is due also to the extremely low dissipation of energy and to the rapid speed of sound propagation (about 344 m/s) which in the absence of convection ( $< 0.2$  m/s) does not leave any time for heat exchange. Then a pressure-density relation can easily

be obtained which also defines the speed of sound  $c$  in air (see also equation 1.3):

$$\frac{\partial p}{\partial \rho} = c^2 = \gamma \frac{\bar{p} + p}{\bar{\rho} + \rho} \approx \gamma \frac{\bar{p}}{\bar{\rho}}. \quad (2.10)$$

Here  $\gamma = 1.4$  for air is the adiabatic exponent.

Now the density perturbation  $\rho$  can be excluded from equation (2.6), and the **linearised Euler equations** (Goldstein, 1993) can be written as:

$$\frac{\partial p}{\partial t} + \bar{v}_j \frac{\partial p}{\partial x_j} + \bar{\rho} c^2 \frac{\partial v_j}{\partial x_j} = S \quad (2.11)$$

$$\frac{\partial v_i}{\partial t} + \bar{v}_j \frac{\partial v_i}{\partial x_j} + \frac{1}{\bar{\rho}} \frac{\partial p}{\partial x_i} = F_i \quad (2.12)$$

with  $S = c^2 Q$  and  $F_i = F'_i / \bar{\rho}$ . The right-hand sides  $S$  and  $F_i$  now accommodate all the small nonlinear terms along with the sources of mass and the forces. When equations (2.11) and (2.12) are solved numerically, the right-hand sides are considered as known functions of  $x_i$  and time  $t$ .

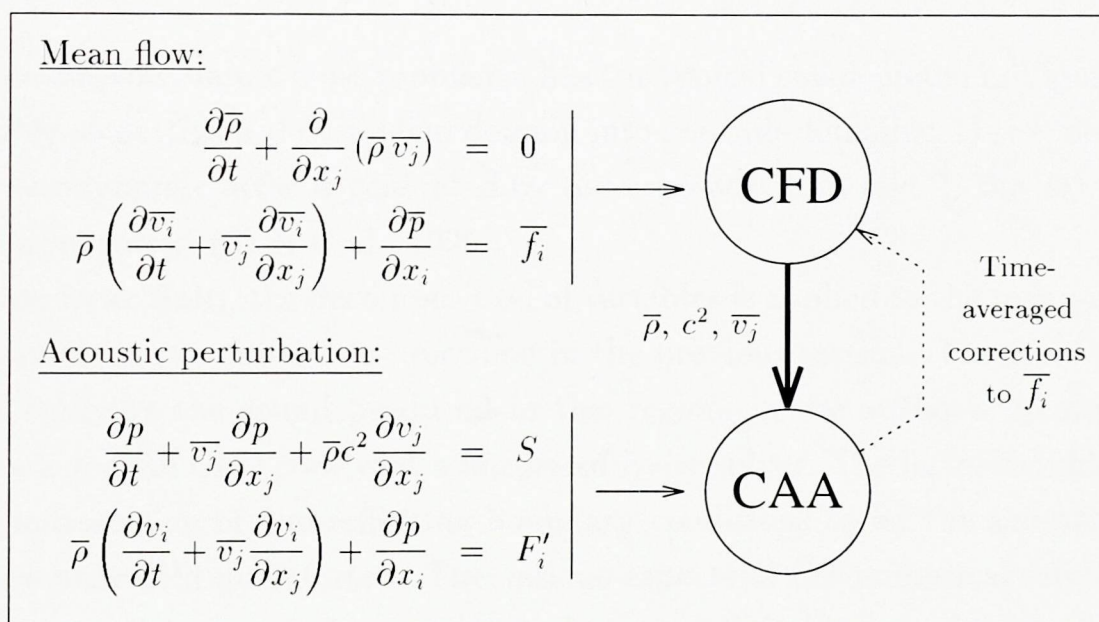


Figure 2.1: Interaction between CFD and CAA codes

As explained earlier, this variable decomposition allows separate treatment of the large-scale mean flow motion and the small sound perturbations superimposed upon it (Hardin, 1993). The mean-flow values  $\bar{\rho}$  and  $\bar{v}_j$  can be computed separately by a CFD code that solves the equations (2.4) and (2.5) together with the *full* energy equation and the equation of state of the gaseous medium. Then, at each time step,

the mean fields can be passed to the acoustic software to serve as a basis of an accurate and efficient numerical simulation of the sound field.

The sequential interaction between the CFD and the CAA codes is pictured in Figure 2.1. In Chapter 5 it will be pointed out that, due to numerical stability limitations, the acoustic time step is usually smaller than the CFD time step. That is why any feedback (denoted by the dotted line) of the sound on the flow field has to be averaged over the CFD time step before introduction into the flow simulation. It is generally assumed that such feedback can only have significant influence on the resulting sound field when resonance occurs (Lighthill, 1952). In most cases of aerodynamic sound generation, iteration in the loop shown in Figure 2.1 is not necessary.

The main ideas discussed in this section were presented at a Domain Decomposition Conference (Djambazov et al., 1997a).

## 2.2 Near Field and Far Field

With most aerodynamic noise problems, like the typical one depicted in Figure 2.2, it is possible to partition the problem domain into two sub-domains: 1) the ‘near field’ where aerodynamic noise is generated by the unsteady flow, and 2) the ‘far field’ of sound propagation (Shih et al., 1995).

In the **near field**, the decomposition of variables is applied to the mean-flow and the perturbation quantities as described in the previous section. The most efficient way of studying the sound produced in this region, in the author’s opinion, is to combine a general CFD code with a *linearised Euler* solver. The latter should include an implementation of non-reflecting boundary conditions to let the acoustic waves leave the near field undistorted. The various aspects of the numerical solution, the boundary conditions and the coupling techniques will be discussed in the following chapters.

With the **far field**, several options exist, and they were briefly discussed in Chapter 1: acoustic analogies, Kirchhoff’s method and direct simulation of the sound.

In the case of zero or uniform mean flow, Lighthill’s analogy and Kirchhoff’s surface integral appear to be the most efficient methods for the far-field sound since their implementations will involve only an evaluation of a volume or surface integral

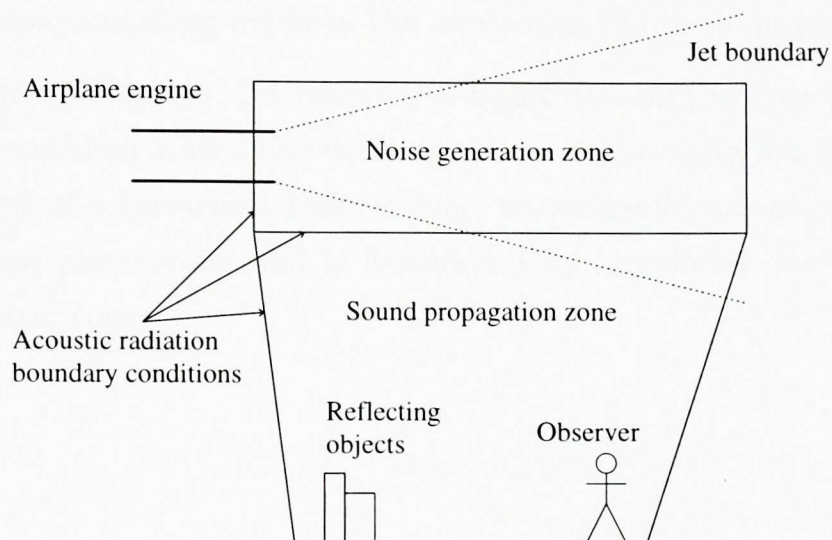


Figure 2.2: Domains for the numerical simulation of aerodynamic sound

for each of the observation points.

When the mean flow cannot be assumed uniform throughout the far field, some of the extended acoustic analogies or the direct sound simulation has to be considered. In the author's opinion, direct CAA methods based on optimised and efficient algorithms (Tam and Webb, 1993) are the better option especially for the lower frequency range of interest. This preference is partly due to the flexibility and the versatility of the direct methods.

If complex sound fields develop near the observation points (like those due to interference with reflected waves - see Figure 2.2), direct CAA calculation seems the only option, especially if the wind velocity (between buildings, etc.) has also to be taken into account.

A more **general** case of domain decomposition is the one with reflected waves from the far field affecting the sound generation process in the near field. This can happen when an airplane flies close to the ground near an airport. If possible (in terms of computational cost) the safest approach to this problem is to consider everything as near field. The CFD solution can still be obtained for a smaller region, and uniform flow can be assumed outside; the linearised Euler simulation, however, covers the whole domain of interest. (This is equivalent to using direct sound computation for the far field.)

If resonance occurs, time-dependent information from the linearised Euler solution has to be fed back into the CFD simulation of the noise sources. This means



that a full two-way coupling between the two codes has to be implemented.

In the following chapters, the research concentrates on the near field of the aerodynamic noise problem where the decomposition of the variables is applied. After the development of a linearised Euler solver, an example case of sound generation by unsteady flow past an aerofoil is considered by sequential one-way coupling of CFD and acoustic codes.

# Chapter 3

## Finite Volume Algorithms

Finite volume methods for the discretisation of the governing equations of fluid flow have established themselves as robust and reliable with CFD simulations (Patankar, 1980; PHOENICS, 1995; Croft et al., 1995). In the *near field* region of aerodynamic sound problems a finite volume CAA technique could easily interface to standard CFD codes. The alternative possibility of using Finite Element methods has not been considered here although it has been applied recently (Atkins, 1997) to sound in air.

In this chapter the possibilities are investigated for creating a finite volume solution procedure for the *linearised Euler equations*. As it was explained previously these equations (2.11) and (2.12) describe the essence of the sound propagating in a non-uniformly moving medium, and are therefore crucial for Computational Aeroacoustics (CAA).

### 3.1 The Case of Uniform Mean Flow

In many cases the gradients of the mean velocity of the flow can be ignored. If the boundary layer of the flow is much thinner than the simulated acoustic wavelength, or if the magnitude of the mean velocity is small enough compared to the speed of sound, the external flow can be successfully approximated by a uniform velocity field.

In this case a *relative* frame of reference is considered that moves with the velocity of the mean flow. That makes the study identical to the simulation of sound waves in still air.

The linearised Euler equations (2.11) and (2.12) can be simplified (with  $\bar{v}_j = 0$ ) and expanded in two dimensions using the substitutions

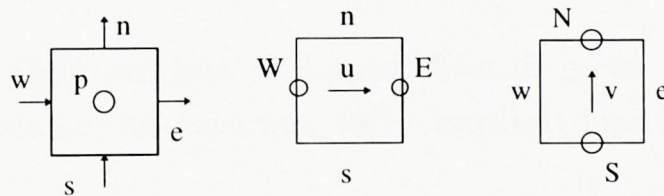
$$\begin{aligned} u &= \bar{\rho}c v_1, \\ v &= \bar{\rho}c v_2, \\ F_x &= \bar{\rho}c F_1, \quad F_y = \bar{\rho}c F_2. \end{aligned} \quad (3.1)$$

The resulting acoustic equations (3.2) can also be obtained directly from (1.1) and (1.4) of Chapter 1.

$$\begin{aligned} \frac{\partial p}{\partial t} + c \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) &= S \\ \frac{\partial u}{\partial t} + c \frac{\partial p}{\partial x} &= F_x \\ \frac{\partial v}{\partial t} + c \frac{\partial p}{\partial y} &= F_y \end{aligned} \quad (3.2)$$

With uniform (or zero) mean flow the speed of sound  $c$  can be assumed constant in the whole domain and can be calculated from (2.10).

When discretising the three unknown functions  $p, u$  and  $v$ , a **fully staggered** (along both space and time) grid has been chosen. This is done because it allows (as shown below) a fully explicit, stable, second-order accurate scheme to be formulated (Morton and Mayers, 1994). Its accuracy can then be extended to third order by allowing the scheme to become implicit while retaining a strong diagonal dominance that guarantees fast convergence.



Using this cell-centred regular Cartesian mesh and the notation pictured above, a finite-volume set of equations can be obtained by successive integration of (3.2) along each of the axes  $x, y$  and  $t$ :

$$\begin{aligned} &\int_{C_p} (p - p_{old}) dx dy + \\ + c &\left[ \int_{old}^{new} dt \int_s^n (u_e - u_w) dy + \int_{old}^{new} dt \int_w^e (v_n - v_s) dx \right] = \int_{old}^{new} dt \int_{C_p} S dx dy \end{aligned} \quad (3.3)$$

$$\int_{C_u} (u - u_{old}) dx dy + c \int_{oldM}^{newM} dt \int_s^n (p_E - p_W) dy = \int_{oldM}^{newM} dt \int_{C_u} F_x dx dy$$

$$\int_{C_v} (v - v_{old}) dx dy + c \int_{oldM}^{newM} dt \int_w^e (p_N - p_S) dx = \int_{oldM}^{newM} dt \int_{C_v} F_y dx dy$$

Here  $C_p$ ,  $C_u$ , and  $C_v$  denote the finite volume cells for pressure,  $u$ -velocity, and  $v$ -velocity respectively.

The discretised values of pressure are stored in the cell centres (with upper case indexes), while the discretised velocity components are stored on cell faces (with lower case indexes) in the *middle* of each time step ( $t_{oldM} = t_{old} + \Delta t/2$ ,  $t_{newM} = t_{new} + \Delta t/2$ ). The storage locations are shown in Figure 3.1. External sources of mass and momentum are also discretised in a staggered way as shown by the symbols ‘Mass so.’ and ‘Mom.so.’.

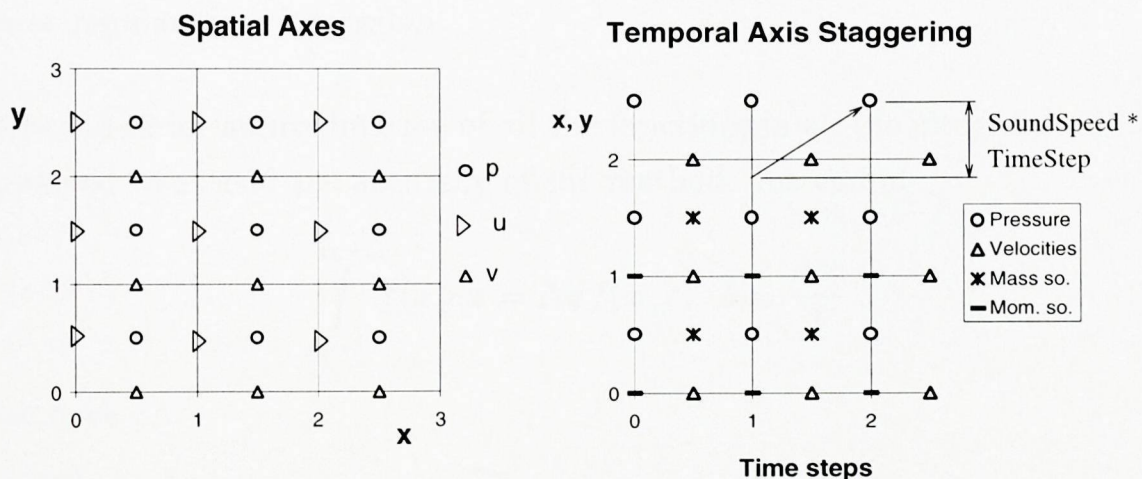


Figure 3.1: Computational grid and radiating boundary interpolation

The integrals in (3.3) are first evaluated through mean values providing the easy-to-program, second-order accurate, **fully explicit** leap-frog scheme (Morton and Mayers, 1994):

$$\begin{aligned} p &= p_{old} - \sigma_x(u_e - u_w) - \sigma_y(v_n - v_s) + S\Delta t \\ u &= u_{old} - \sigma_x(p_E - p_W) + F_x\Delta t \\ v &= v_{old} - \sigma_y(p_N - p_S) + F_y\Delta t \\ \sigma_x &= \frac{c\Delta t}{\Delta x}, \quad \sigma_y = \frac{c\Delta t}{\Delta y} \end{aligned} \tag{3.4}$$

As it can be seen from Figure 3.2 this scheme is accurate enough within about 5 wavelengths, accumulating not more than 8% error, and it can only be used on its own in small domains.

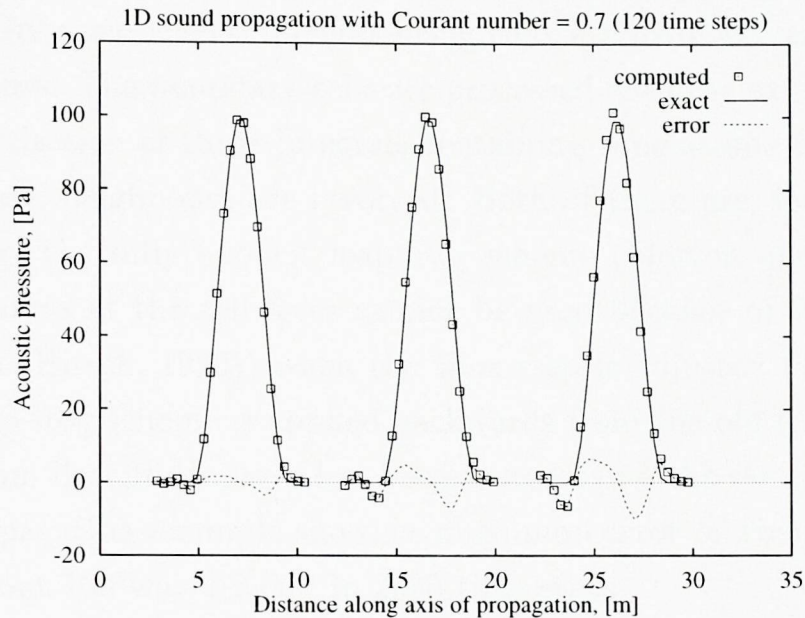


Figure 3.2: The fully-explicit leap-frog scheme. 1D propagation of a single pulse is shown at regular time intervals.

A second-order approximation of all the functions in all the integrals in (3.3) can be employed to extend the accuracy of the method. Instead of

$$\int_{x_0-\delta}^{x_0+\delta} f(x)dx = \Delta x f(x_0), \quad \delta = \frac{\Delta x}{2}$$

we now have

$$\int_{x_0-\delta}^{x_0+\delta} f(x)dx = \Delta x [Af(x_0 - \Delta x) + (1 - 2A)f(x_0) + Af(x_0 + \Delta x)], \quad (3.5)$$

applied to all integrals. In 2D this means

$$\int_{cell} f(x, y)dxdy = \left[ A \sum_{nb} f_{nb} + (1 - 4A)f_{cell} \right] \Delta x \Delta y \quad (3.6)$$

with ‘*nb*’ used to denote all four neighbouring cells. The coefficient  $A = \frac{1}{24}$  can be derived easily by integrating the second order interpolation polynomial defined by the three points  $f(x_0 - \Delta x)$ ,  $f(x_0)$ , and  $f(x_0 + \Delta x)$  within the specified limits. The same value is obtained also in two and in three dimensions.

This **semi-implicit scheme** proved to be accurate enough to take the sound generated out of the ‘near field’. The resulting linear system is solved iteratively at each time step starting with a very good initial guess computed using the explicit

scheme (3.4). Only one level of neighbouring cells are involved, and not second or further neighbours. The boundary cells are processed with the explicit scheme only. With the discretisation of those integrals containing time as one of the arguments, ‘future’ and ‘past’ neighbours are involved. Both of them are determined at each iterative step by the fully explicit leap-frog scheme (Morton and Mayers, 1994). Stored ‘past’ values at the cell faces cannot be used because of the Courant limit (*CFL* condition (Hirsch, 1990)) when the time step is adjusted to the whole cells; instead, the leap-frog scheme is applied backwards from the old time level.

The algorithm described above has been tested using the same one-dimensional plane wave propagation example showing maximum error of about 5% of the amplitude over about 100 wavelengths in 2000 time steps.

The test solution as depicted in Figure 3.3 is obtained with Courant number  $\sigma_x = 0.7$  (see equation 3.4) which is relevant for two-dimensional simulation. In 3D cases, the corresponding limit is 0.577. At the limit in 1D  $\sigma_x = 1$  the two schemes are exact (i.e. they have no approximation error).

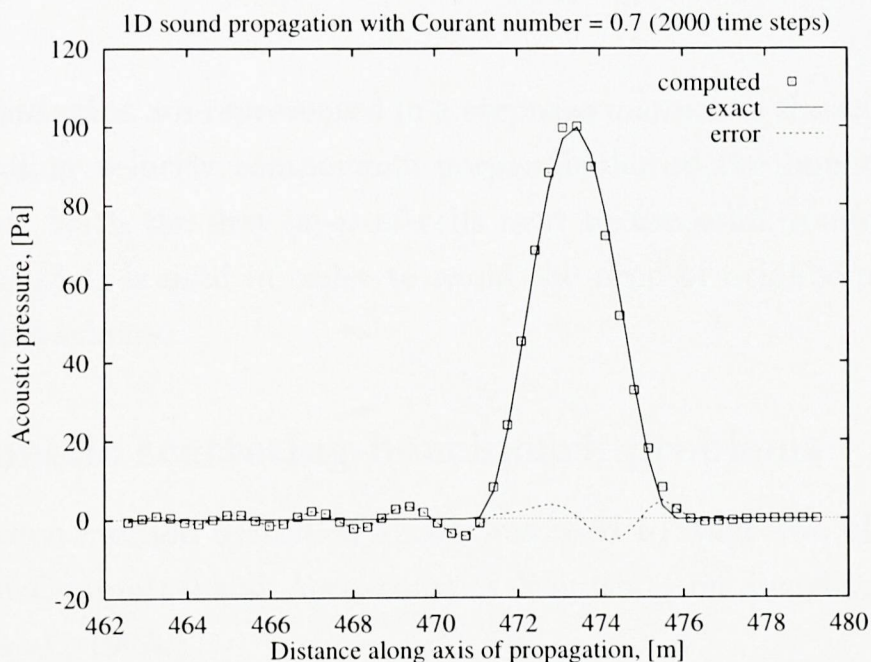


Figure 3.3: The semi-implicit scheme. Test pulse after travelling 100 wavelengths.

Acoustic radiation **boundary conditions** are implemented assuming *plane wave* propagation in the boundary regions of the domain: velocity components at the boundary faces are computed from the velocity field at the previous step by interpolation at the appropriate points inside the domain (see the arrow in Figure 3.1). The direction of radiation has to be prescribed for each boundary cell. Then the

interpolation point inside the domain can be determined by considering the distance which the wave covers in one time step. A detailed description of the plane wave boundary approach can be found in Chapter 4 (see equation 4.16).

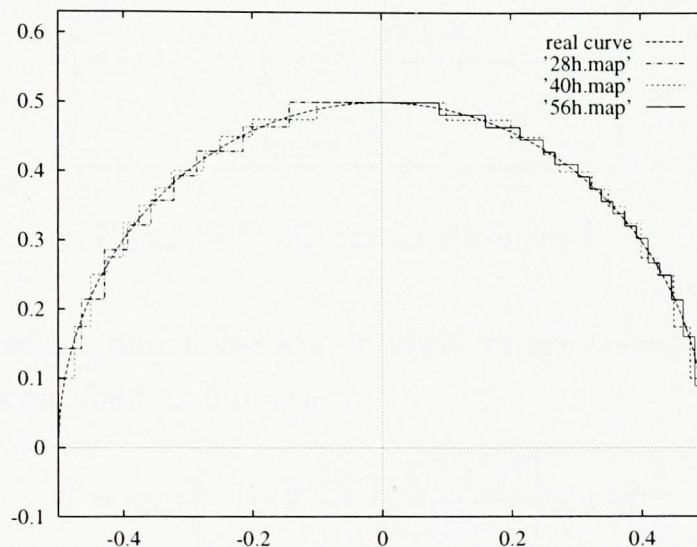


Figure 3.4: Solid boundary - stepwise representation showing the effects of mesh refinement

**Solid boundaries** are represented in a stepwise manner as shown in Figure 3.4. The corresponding velocity components perpendicular to the boundary cell faces are set to zero. With the first layer of cells next to the solid boundary the fully-explicit scheme (3.4) is used in order to avoid the need of neighbouring values for the semi-implicit scheme.

### 3.1.1 Acoustic scattering benchmark problems

The finite volume method described above was used to solve two of the problems at the ‘Second Computational Aeroacoustics Workshop on Benchmark Problems’ (Djambazov et al., 1997c).

#### Problem 1

The physical problem concerns the sound field generated by a propeller scattered off the fuselage of an aircraft. The pressure loading on the fuselage is of interest as an input to the interior noise problem. The benchmark was set to test in particular the accuracy of curved boundary conditions which in this work were modelled in a stepwise fashion as shown above (Figure 3.4).

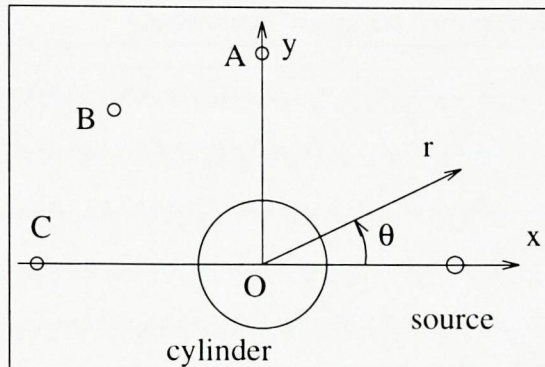


Figure 3.5: Sketch of Problem 1

The idealised fuselage and noise source position are shown in Figure 3.5. The source of sound is prescribed as follows:

$$S = \exp \left[ -(\ln 2) \frac{(x - 4)^2 + y^2}{(0.2)^2} \right] \sin \omega t \quad (3.7)$$

with  $\omega = 8\pi$ .

Results were to be compared against the benchmark solution which was not known beforehand at positions A, B and C.

A uniform Cartesian grid of up to  $800 \times 400$  cells was used for this benchmark, over a domain of approximately  $14 \times 7$  units (Figure 3.6). The cell size was 0.01786 units, and the time step 0.0125 units (normalised w.r.t.  $D/c$ , the cylinder diameter over the sound speed). The instantaneous pressure field generated by the sound source is shown in Figure 3.6. The distance between the cylinder and source is an integer number of wavelengths, leading to the creation of a standing wave between them. A sound “shadow” is evident at the rear of the cylinder. The successive amplification/attenuation of the sound signal as the emitted and reflected signals meet, is better seen in Figure 3.7, which shows an instantaneous map of the pressure throughout the domain.

The benchmark solution, published in (Kurbatskii, 1997), was analytic and independent of radius for  $r \rightarrow \infty$ . The numerical solution however can only be realised for a finite radius; for this reason results were produced for comparison by taking the square of the pressure at two radii,  $r = 5$  and  $r = 7$ . A comparison of the computed and analytic time-averaged results is shown in Figures 3.8 and 3.9. There is a close correlation between the two results, with the  $r = 7$  solution being the closer of the two as expected.



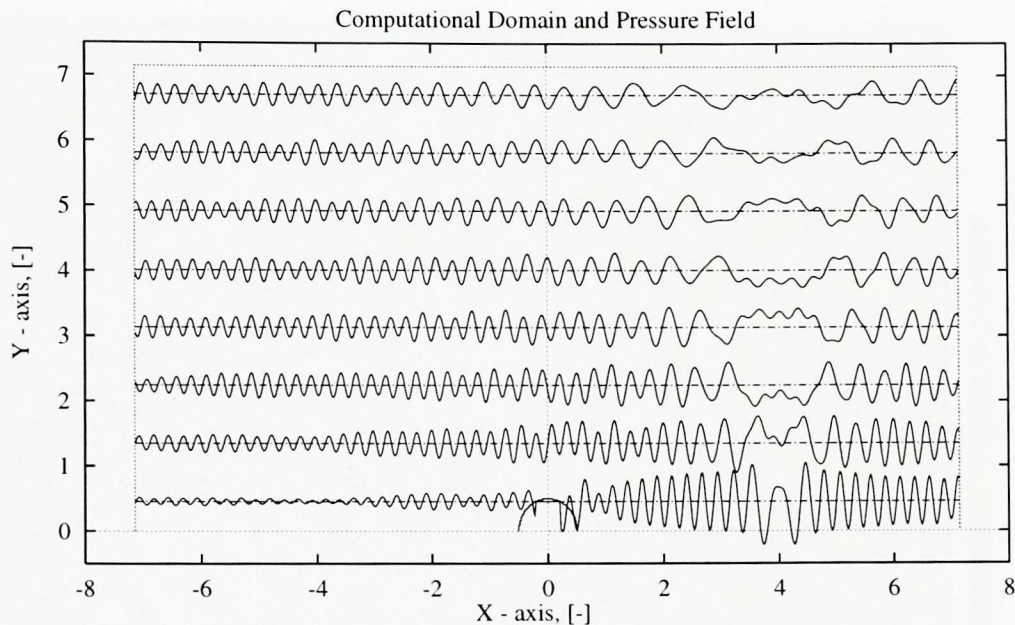


Figure 3.6: Computational domain and instantaneous pressure field for Problem 1

## Problem 2

This is the same as Problem 1, except that there is no time-periodic source, i.e.  $S = 0$ . Instead, an initial pressure pulse is specified which then propagates in time. This is then an initial value problem, with  $u = v = 0$  at  $t = 0$ , and

$$p = \exp \left[ -(\ln 2) \frac{(x - 4)^2 + y^2}{(0.2)^2} \right]. \quad (3.8)$$

The same grid spacing ( $h = D/56$ ), time step ( $\Delta t = 0.01$ ) and solid boundary mapping as with Problem 1 was used, but the extent of the domain was reduced to  $600 \times 300$  cells - enough to contain the target semi-circle of radius  $r = 5$  units.

Solutions are compared against the analytic ones at three points, A ( $r = 5$ ,  $\theta = 90^\circ$ ), B ( $r = 5$ ,  $\theta = 135^\circ$ ), C ( $r = 5$ ,  $\theta = 180^\circ$ ), as shown in Figures 3.10(a)–3.10(c). The computations in this case were performed using three different meshes to check grid dependence of the results ( $h = D/40$  and  $h = D/28$ ). In all three cases, the mesh is fine enough to resolve the pressure pulse, but the stepwise discretisation of the cylinder is different (see Figure 3.4).

Comparison between the benchmark and numerical solutions shows that the technique predicts the passage of the pressure pulse over the obstacle quite accurately. However, and quite surprisingly, the coarser mesh seems to give a better agreement with the benchmark results in position A, at a right angle to the cylinder. It appears that the finest mesh introduces high frequency oscillations to the solution. These

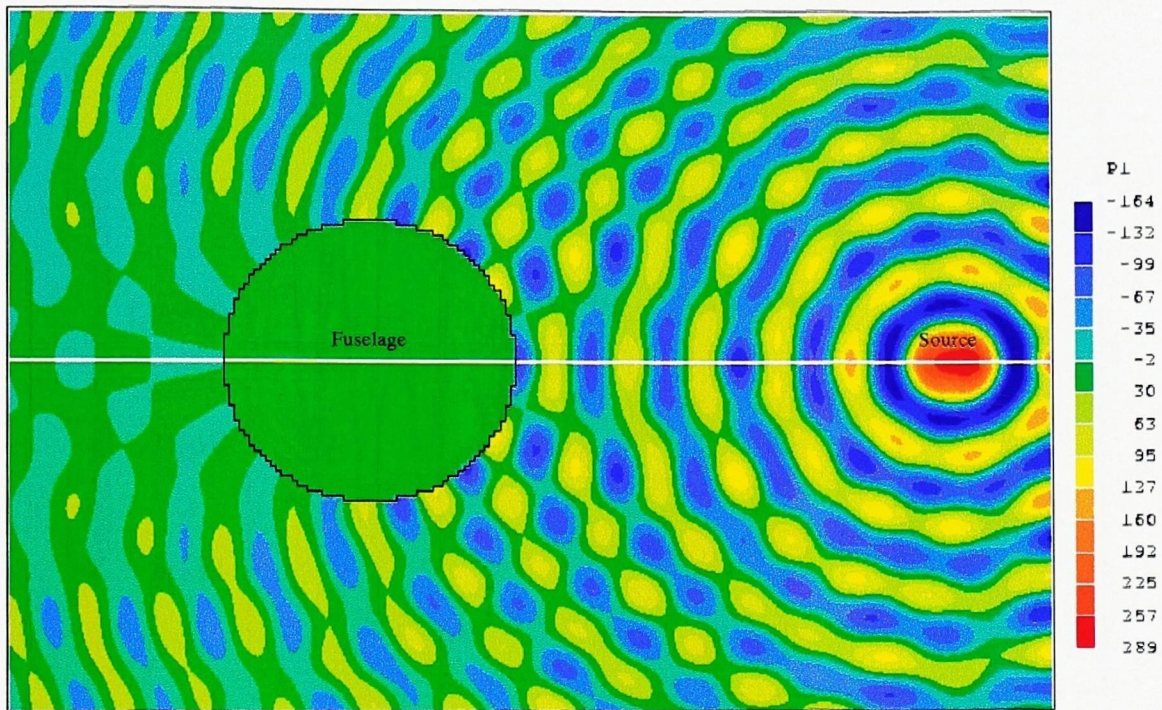


Figure 3.7: Contours of instantaneous pressure showing interference patterns caused by the interaction of emitted waves from the source and reflected waves from the cylinder.

oscillations are almost certainly due to a correlation between the pulse wavelength and the cylinder wall step size.

Typical times for these runs on a Sun Sparc 20 workstation are: 1 hour for the coarse mesh, 3 hours for the medium mesh, and 10 hours for the fine mesh. This indicates an approximately linear relationship between mesh size and computational time.

Compared to the rest, the accuracy of the results presented here was encouraging. This suggested that after the implementation of the convection terms, the resulting linearised Euler solver will be suitable for the *near field* of aeroacoustic calculations.

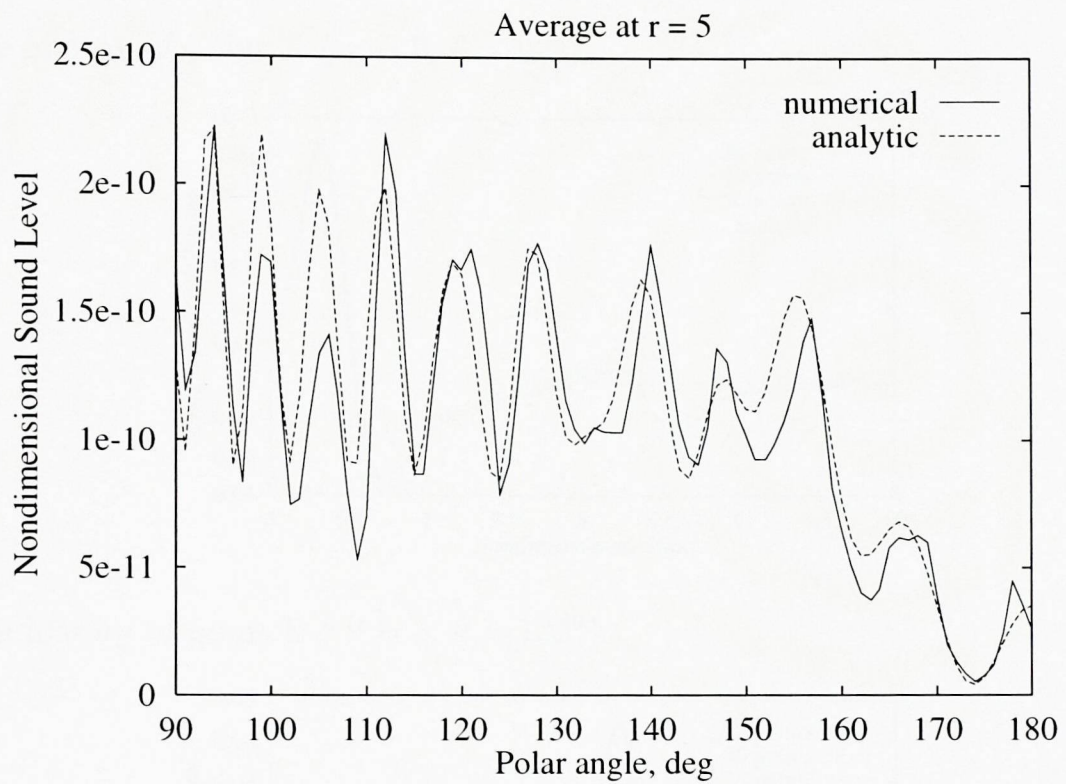


Figure 3.8: Comparison between numerical and benchmark solutions for Problem 1, at  $r=5$

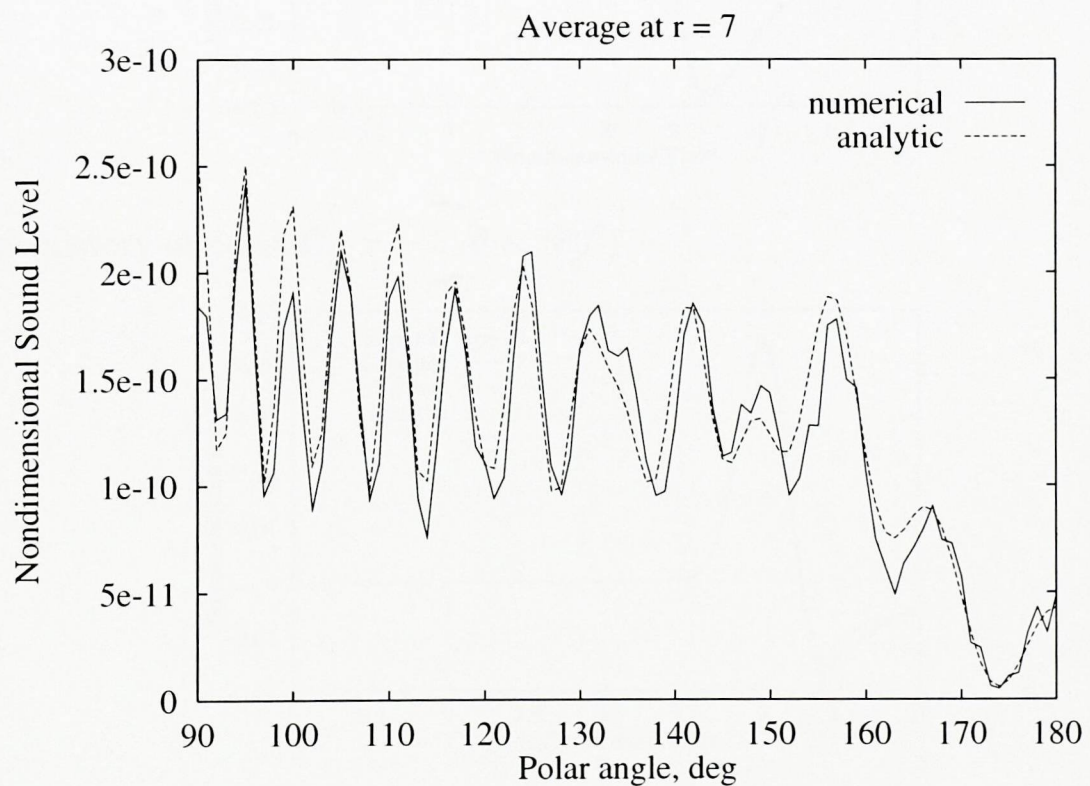
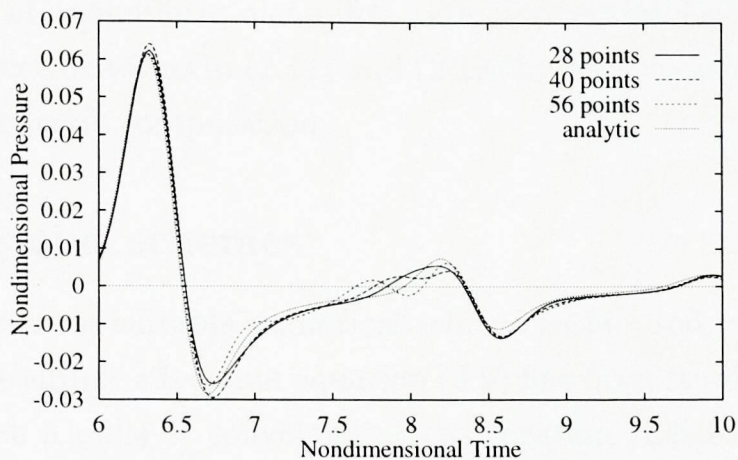
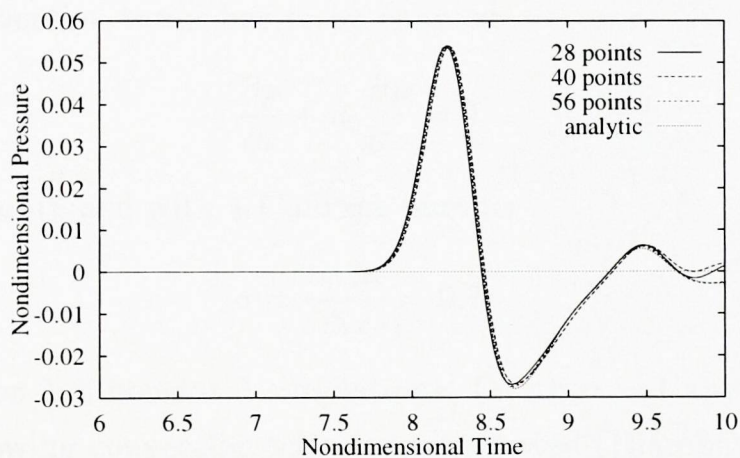


Figure 3.9: Comparison between numerical and benchmark solutions for Problem 1, at  $r=7$

(a) Time history of point A ( $r = 5, \theta = 90^\circ$ )



(b) Time history of point B ( $r = 5, \theta = 135^\circ$ )



(c) Time history of point C ( $r = 5, \theta = 180^\circ$ )

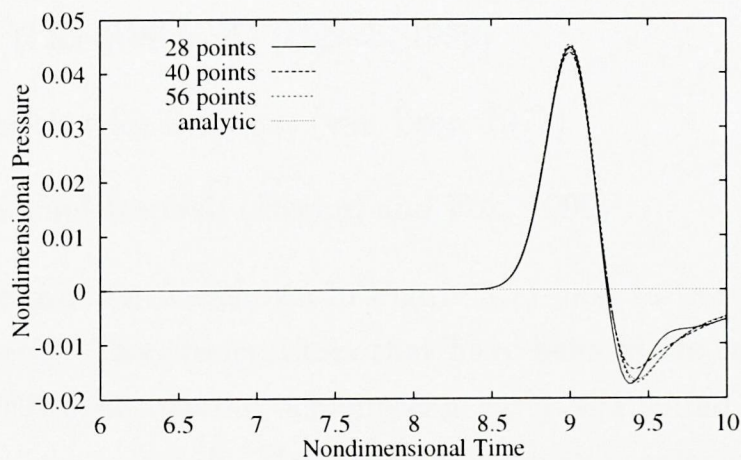


Figure 3.10: Comparison between numerical and analytic solution, and the effect of mesh density for Problem 2

## 3.2 Linearised Euler Solutions

In the general case of nonuniform mean flow careful attention has to be paid to the corresponding convection terms in (2.11) and (2.12) in order to achieve the accuracy necessary for aeroacoustic computations.

### 3.2.1 Three-point schemes

In order to find the most suitable numerical scheme to be used with the linearised Euler equations, the simple advection equation (3.9) has been studied comparing the accuracy with which a pulse is convected at the constant mean velocity  $u_0$  over 5 wavelengths. This testing distance was chosen because it is the acceptable accuracy limit of the fully-explicit pure propagation scheme (3.4) with which the numerical scheme for the convection terms has to be coupled.

$$\frac{\partial p}{\partial t} + u_0 \frac{\partial p}{\partial x} = 0 \quad (3.9)$$

Results have been obtained with a Courant number

$$\sigma = \frac{u_0 \Delta t}{\Delta x} = 0.7 \quad (3.10)$$

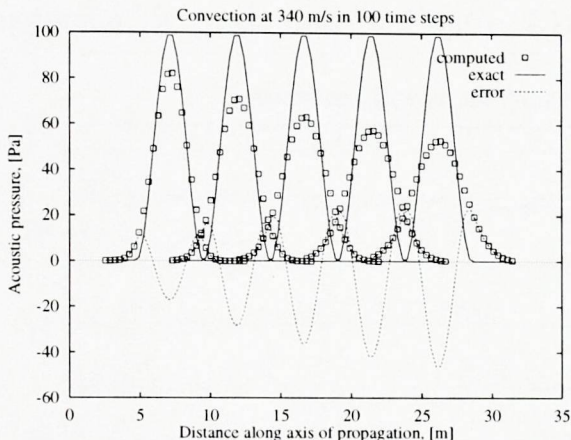
which is relevant for 2-dimensional simulations. (With  $\sigma = 1$  any of these schemes is exact.) The following convection schemes were tested (Djambazov et al., 1997b):

1. Upwind fully explicit
2. MacCormack (Lax-Wendroff) (Hirsch, 1990)
3. van Leer (explicit with limiters) (van Leer, 1977)
4. Upwind/downwind Implicit (Degani and Fox, 1996)

Solutions of the test problem are shown in Figure 3.11 next to the equations defining each numerical scheme. The observations that have been made can be summarised as follows. The CFD-type upwind scheme can carry the sound signal accurately for no more than *one* wavelength. MacCormack's scheme seems more accurate but it becomes unstable if combined with the pure-propagation scheme (3.4). Schemes bounded by means of limiters (van Leer, 1977) introduce false noise to the sound signal.

**Upwind Fully Explicit:**

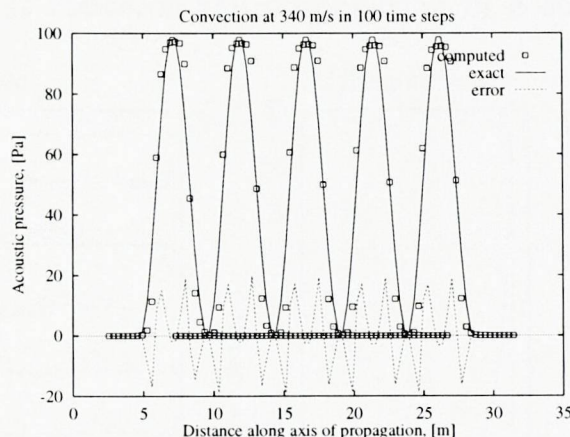
$$p = \sigma p_{upwind}^{old} + (1 - \sigma)p^{old}$$



The CFD-type upwind scheme can carry the sound signal accurately for *one* wavelength only.

**van Leer – explicit with limiters on the flux values  $\bar{p}_e$  and  $\bar{p}_w$ :**

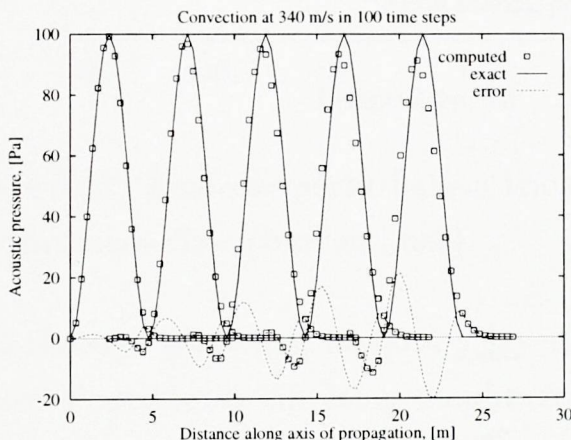
$$p = p^{old} - \sigma(\bar{p}_e - \bar{p}_w)$$



Schemes bounded by means of limiters introduce false noise to the sound signal.

**MacCormack/Lax-Wendroff (explicit):**

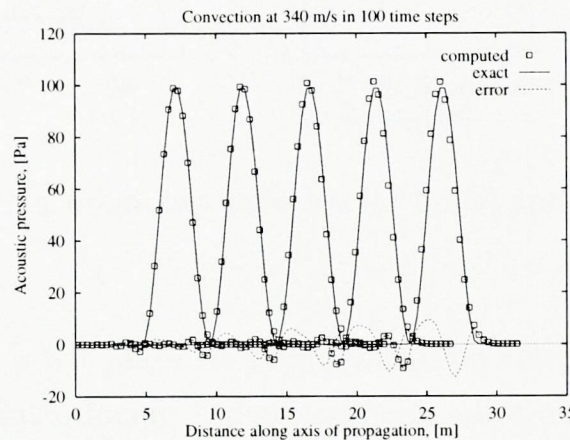
$$p = p^{old} - \frac{\sigma}{2}(p_E^{old} - p_W^{old}) + \frac{\sigma^2}{2}(p_E^{old} - 2p^{old} + p_W^{old})$$



MacCormack's scheme seems more accurate but it becomes unstable if combined with the pure-propagation scheme.

**Upwind/downwind Implicit:**

$$(1 - \frac{\sigma}{2})p + \frac{\sigma}{2}p_{downwind} = (1 - \frac{\sigma}{2})p^{old} + \frac{\sigma}{2}p_{upwind}^{old}$$



A convection scheme that has the same accuracy as the fully-explicit propagation scheme and can be coupled with it using finite volumes.

Figure 3.11: Comparison of different numerical schemes for pure convection

The upwind/downwind scheme is selected to be combined with the pure propagation scheme (3.4) because they have the same accuracy, and the finite volume formulation is also space centred. This combined method can be used in small computational domains (not longer than 5 wavelengths). It takes between 4 and 8 iterations at each time step depending on the magnitude of the mean velocity. The method and the computer code are easily extensible to three dimensions.

An example solution of the full linearised Euler equations using the combination of second order numerical schemes explained above for both the convection and the propagation, is presented in Figure 3.12. It concerns the propagation of a sound

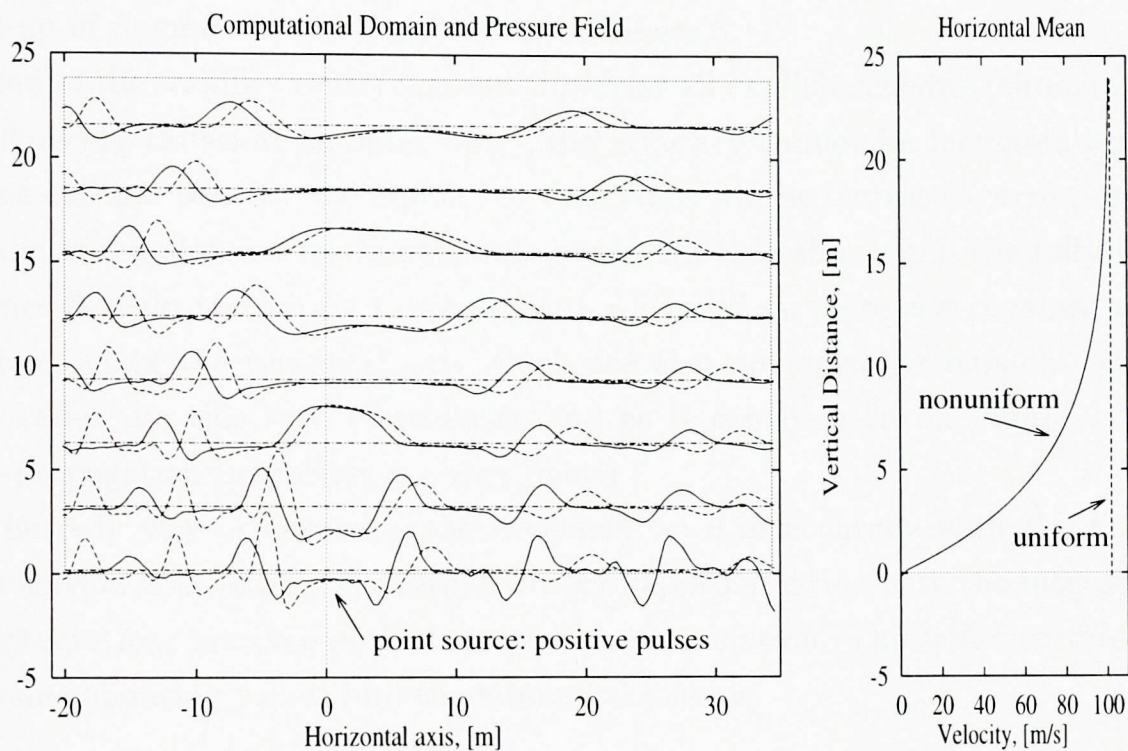


Figure 3.12: Acoustic perturbation travel in a boundary layer (solid lines) and in a uniform mean flow (broken lines)

wave through a laminar boundary layer on a flat plate. A point source on the plate surface simulates a vibrating diaphragm, introducing a displacement pulse on the surface. The boundary layer velocity profile is assumed to be the one plotted with solid line in the right graph of Figure 3.12. In a real coupled simulation this profile will be computed by a CFD code. The CFD domain can be smaller than the acoustic one in order to reduce the computational cost. Then uniform mean flow is assumed outside the CFD region.

Phase displacement of the signal propagating in a boundary layer is observed (relative to the signal in a uniform flow), which is a function of the mean velocity magnitude. The pressure perturbation is “compressed” upstream of the source, and “expanded” downstream as expected (Djambazov et al., 1998a).

### 3.2.2 Five-point schemes

In realistic cases when the computational domain is of size greater than five wavelengths, the accuracy of the three-point finite volume schemes is insufficient. (The build-up of numerical error can be seen in Figure 3.11.)

Due to the stability limit (Courant limit) for all explicit schemes (those involving neighbouring values at an older time), the accuracy cannot be increased by reducing the cell size because the number of time steps will be increased correspondingly, and the accumulated error remains the same. On the other hand, the fully implicit schemes that do not have a Courant limit are based on piecewise constant approximations along the temporal axis which are also not accurate enough. (General CFD codes use this type of schemes, and as it can be seen on Figure 1.1, their wave-propagation properties are very poor.)

The only way of achieving the necessary level of accuracy with the linearised Euler solvers is increasing the formal order of approximation with the interpolations and integrations involved in the finite volume formulation. This will inevitably bring more neighbouring values into the numerical scheme.

#### Two time levels

Considering an approximating function  $f(x, t)$  that is of second order in space and of first order in time, it can be seen below that 3 ‘old’ and 2 ‘new’ values are enough to define its coefficients:

$$f(x, t) = a_0 + a_1x + a_2x^2 + a_3xt + a_4t. \quad (3.11)$$

When the mean velocity along  $x$  is positive these values are:

$$\begin{aligned} W_0 &= f(-\Delta x, 0) \quad , \quad U_0 = f(0, 0), \quad D_0 = f(\Delta x, 0) \\ W &= f(-\Delta x, \Delta t) \quad , \quad U = f(0, \Delta t), \end{aligned} \quad (3.12)$$

and the coefficients determined from the corresponding linear equations are:

$$\begin{aligned} a_0 &= U_0, \quad a_1 = \frac{D_0 - W_0}{2\Delta x} \quad , \quad a_2 = \frac{(W_0 + D_0)/2 - U_0}{(\Delta x)^2} \\ a_3 &= \frac{U - U_0 - (W - W_0)}{\Delta x \Delta t} \quad , \quad a_4 = \frac{U - U_0}{\Delta t}. \end{aligned} \quad (3.13)$$

Then the interpolated value needed for the calculation of the convective fluxes is

$$f\left(\frac{\Delta x}{2}, \frac{\Delta t}{2}\right) = \frac{1}{8}W_0 + \frac{3}{8}D_0 - \frac{1}{4}W + \frac{3}{4}U. \quad (3.14)$$



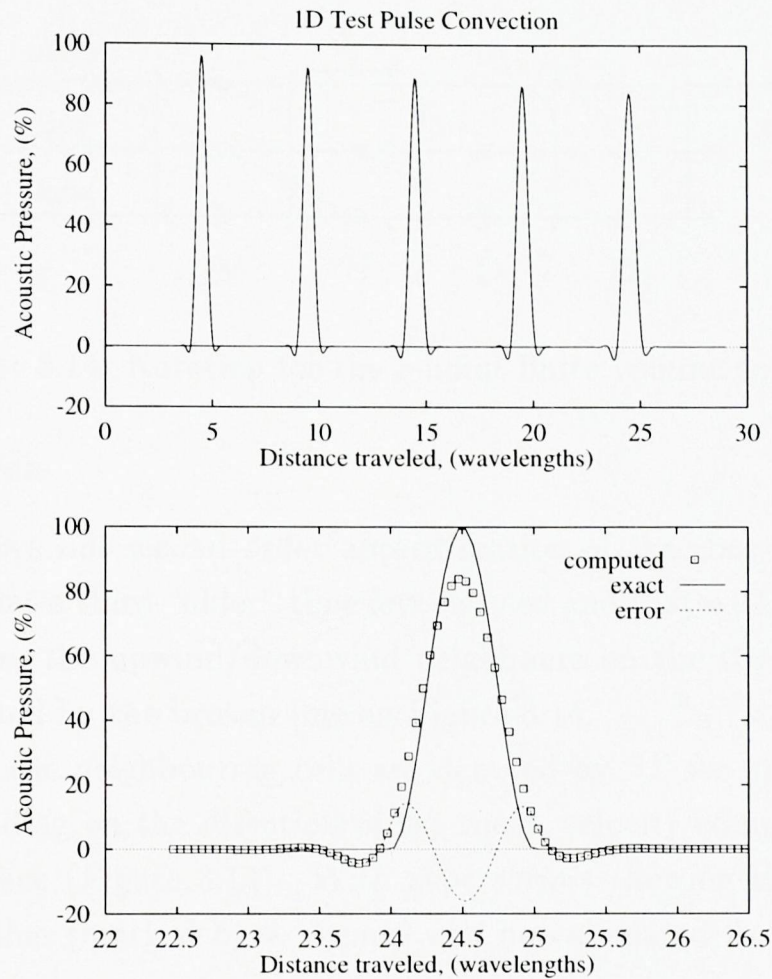


Figure 3.13: Test pulse travel and numerical error of the two-level scheme

Since only ‘upwind’ neighbours at the ‘new’ time level are involved for each cell face, this formula will produce stable iterative schemes within the usual Courant limit for explicit time stepping.

To test this approximation, the simple equation  $\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} = 0$  containing only a transient term and a convection term is considered. The second order quadrature (3.5) is applied to the transient term, and the finite volume form of the convection term is a difference between two fluxes expressed by means of (3.14).

An example solution is shown in Figure 3.13 of a pressure pulse carried by a constant mean velocity over 25 wavelengths. The whole computation takes 5000 time steps with 20 points per wavelength, and the graphs are plotted at regular intervals.

The numerical error observed for this typical set of simulation parameters is between 15% and 20% which is not acceptable for practical applications because it may prove to be higher than the reflected waves in a real physical acoustic field.

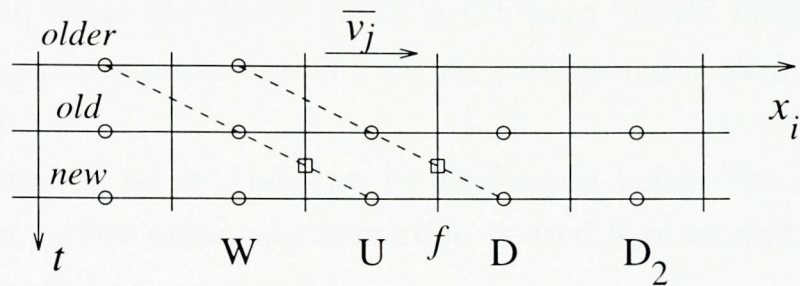


Figure 3.14: Notation for the 5-point finite volume schemes

### Three time levels

In order to achieve full second order approximation of the convective fluxes, it is suggested here that a third “older” time level is used and that with the interpolation of the face values, the upwind/downwind neighbours on the three time levels are involved as denoted by the broken line on Figure 3.14.

For a face  $f$ , the neighbouring cells are denoted by ‘U’ for upwind and ‘D’ for downwind depending on the direction of the mean velocity component  $\bar{v}_j$  perpendicular to that face (Figure 3.14). With *superscripts* showing the time level, the mid-time face value (marked by a square) can be calculated from a second order approximation:

$$f^{mid} = -\frac{1}{8}W^{older} + \frac{3}{4}U^{old} + \frac{3}{8}D^{new}. \quad (3.15)$$

This value will be used to calculate the convection flux across face  $f$  that is due to the mean velocity  $\bar{v}_j$  and the gradient of the variable considered (pressure  $p$  or velocity perturbation  $v_i$ ).

Each of the linearised Euler equations (2.11) and (2.12) contains three convection terms (one term for each spatial axis). For brevity only the first term of the first equation (2.11) is considered here, namely:

$$\bar{v}_1 \frac{\partial p}{\partial x_1} = u_0 \frac{\partial p}{\partial x} \quad (3.16)$$

To obtain its finite volume form this term is integrated successively over the cell volume and along the temporal axis (from the *old* to the *new* time value). With the volume integral, the first direction of integration can be  $x$  which is the direction of the partial derivative. The result is a *flux* difference, and each of the two fluxes  $f_I$  has the form:

$$f_I = u_0 \int_l^h dz \int_s^n dy \int_{old}^{new} f(y, z, t) dt \quad (3.17)$$

where  $l$ ,  $h$ ,  $s$  and  $n$  are the ‘low’, ‘high’, ‘south’ and ‘north’ faces of the cell, and  $f(y, z, t)$  is the pressure perturbation  $p$  at the corresponding *face* perpendicular to  $x$  (‘east’ or ‘west’).

Now the temporal integration can be performed before the other two. It is suggested that a *second* order approximation is used here as well (tests using first order showed insufficient accuracy):

$$f_T = \int_{old}^{new} f(t) dt = \frac{1}{6} f^{old} + \frac{2}{3} f^{mid} + \frac{1}{6} f^{new}. \quad (3.18)$$

The values  $f^{old}$  and  $f^{new}$  are best interpolated from the corresponding cell centres with a third order polynomial (see Figure 3.14):

$$f = \frac{9}{16}(U + D) - \frac{1}{16}(W + D_2), \quad (3.19)$$

and  $f^{mid}$  is known from (3.15).

Finally, the remaining double spatial integral along  $y$  and  $z$  should also be calculated from second order approximations to be consistent with the treatment of the transient term and the other terms described in the previous section. For the computed values  $f_T$  on each face this can be done using the quadrature formula (3.6).

This complex numerical scheme was implemented in one dimension in order to be tested against analytical solutions of the linearised Euler equations.

With the initial conditions two waves are created: one propagates along the direction of the mean velocity, and the other one travels against the mean flow. The simulation time is enough for the first wave to travel 20 wavelengths. In Figure 3.15 the whole 1D domain and the two waves are shown in the middle and at the end of the simulation. The mean velocity is positive, so the first wave ends to the right of the domain, and the second wave has propagated towards the left end of the domain.

It can be seen that the numerical error depends on the direction of the mean flow: the wave that propagates along the mean flow accumulates less error but some trailing false oscillations can be observed. The modified upwind/downwind scheme with extended accuracy presented in this section is stable within the usual Courant limit and exhibits stronger numerical dissipation for waves propagating opposite the mean flow.

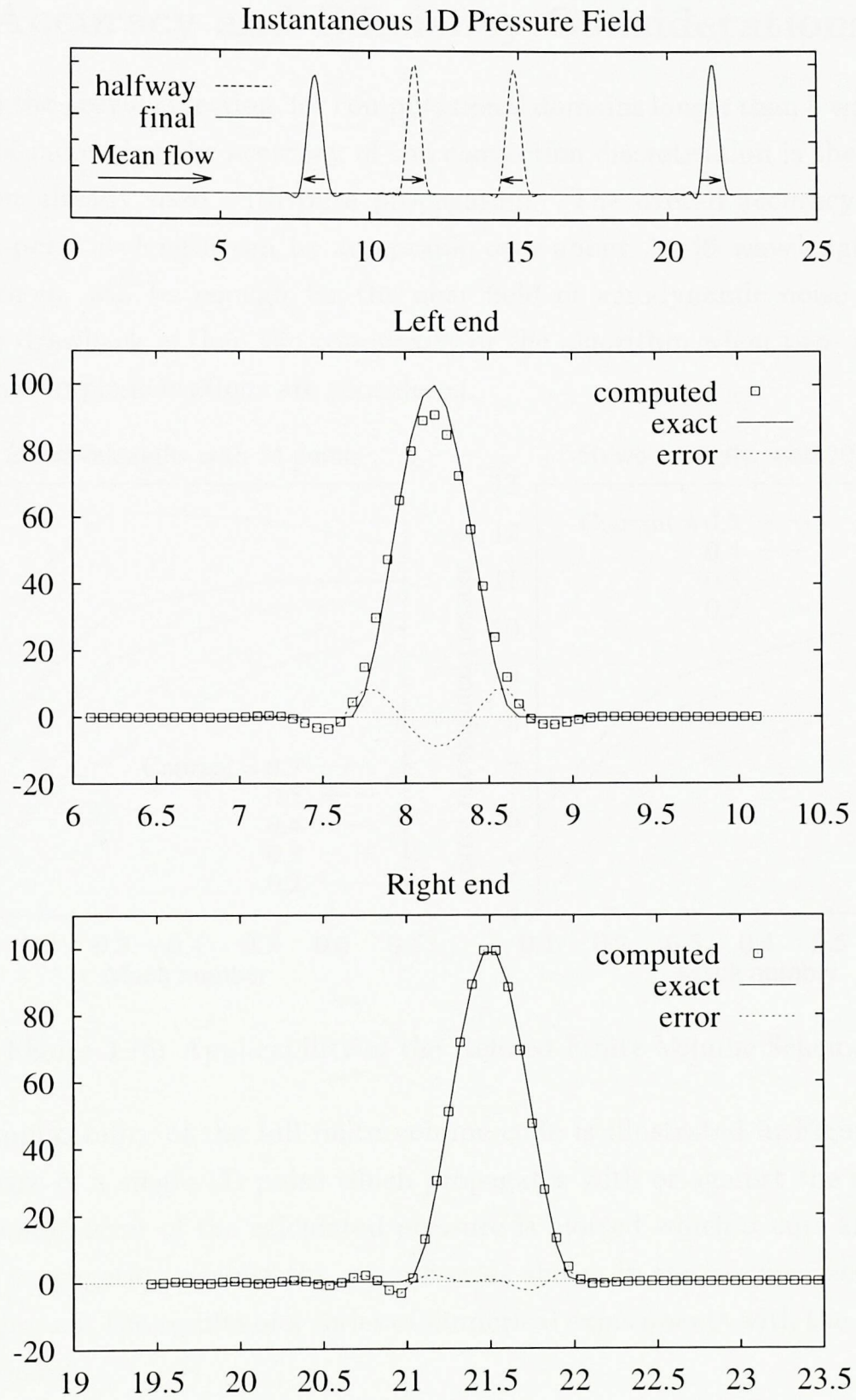


Figure 3.15: Finite volume test solution at Mach number 0.2

### 3.3 Accuracy and Efficiency Considerations

As seen in the previous section, for computational domains longer than 5 wavelengths one way of increasing the accuracy of the convection discretisation is the parabolic integration already used with pure propagation. The overall accuracy with 20–50 points per wavelength can be acceptable over about 20–40 wavelengths which, in most cases, will be enough for the near field of aerodynamic noise problems. The main drawback is then the complexity of the algorithm when two- and three-dimensional implementations are considered.

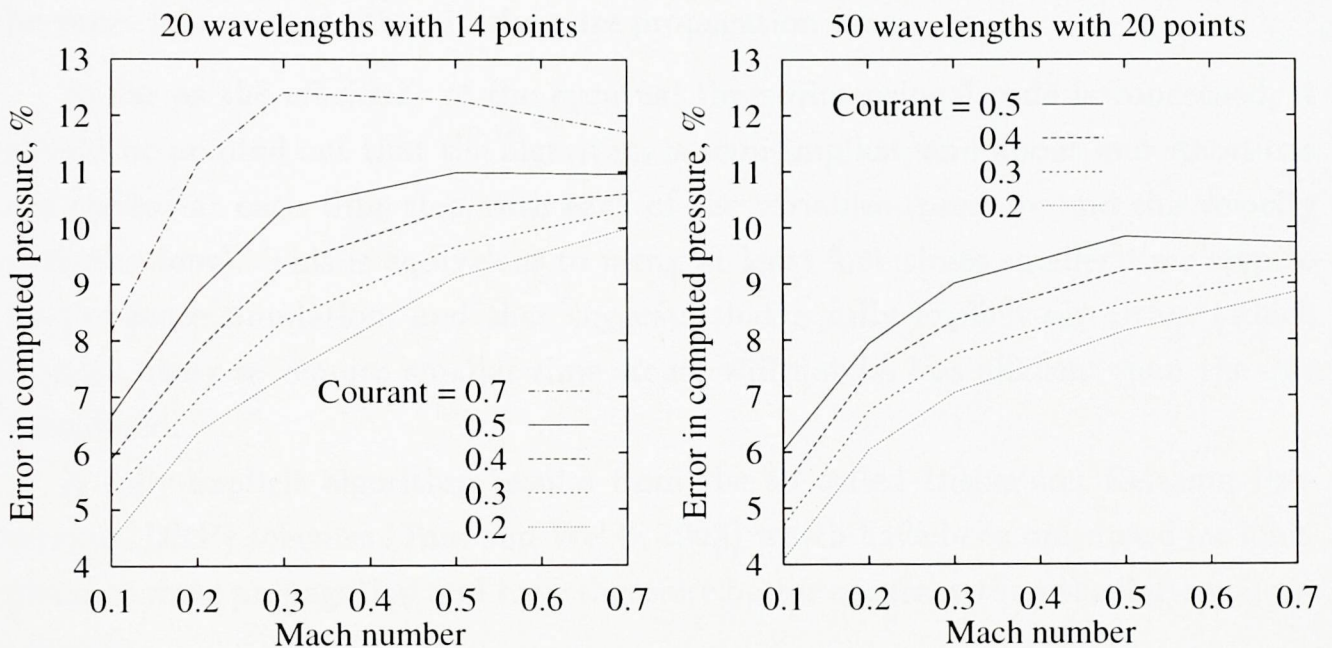


Figure 3.16: Applicability of the Refined Finite Volume Scheme

The applicability of the full finite volume code is illustrated in Figure 3.16 for the test case of a single 1D pulse which propagates with or against the mean flow. The maximum error of the calculated pressure is plotted which occurs always with the pulse propagating against the mean flow as shown in the previous section. The graphs represent the results of a series of numerical experiments with the three-level 5-point scheme.

With the mean convection velocity  $u_0$  and the speed of sound  $c_0$  the Courant number is  $\sigma = (u_0 + c_0)\Delta t/\Delta x$ , and the Mach number is  $M = u_0/c_0$ . Courant numbers higher than 0.5 are of no interest with this numerical scheme because of its 3D stability limit. The simulation time in each case is the time which the positive wave takes to propagate along the mean flow direction:  $t = w\lambda/(u_0 + c_0)$  where  $\lambda$  is

the wavelength, and  $w$  is the number of wavelengths travelled (either 20 or 50).

With a fixed Mach number and cell size  $\Delta x$  increasing the Courant number is equivalent to using larger time steps. Then the spacing between the nodes of the second order interpolation (3.15) is also increased (see dashed line in Figure 3.14), and this leads to larger approximation errors.

With a fixed Courant number and increasing Mach number, the relative influence of the convection term increases, and since its discretisation error is larger than with pure propagation (see Figure 3.3), the overall numerical error also increases.

It can be seen that, with the inclusion of the convection terms, the method becomes less accurate than in the pure propagation case.

As far as the efficiency of the eventual three-dimensional code is concerned, it should be pointed out that the algorithm is semi-implicit, and about four iterations are needed at each time step with each of the variables (pressure and the velocity perturbations). This is equivalent to using at least four times smaller time step to do the same simulation, and that suggests that a fully explicit algorithm (which is most likely to require smaller time steps) will not be less efficient than the one considered.

A fully explicit algorithm results from the so called Dispersion Relation Preserving (DRP) schemes (Tam and Webb, 1993) which have been optimised for long-distance wave propagation and have therefore better accuracy than the finite volume schemes.

The proper decision at this stage is to give up the finite volume convection of acoustic signals in favour of the DRP schemes which will be discussed in the next chapter.

# Chapter 4

## The Acoustic Module

Aeroacoustics is about sound produced by unsteady airflow; computational aeroacoustics aims at making numerical simulations of this process based on fundamental physical principles. In the first two chapters, it was shown that the acoustic motion is a perturbation of small magnitude superimposed on the flow, and that the sound field is best simulated by a special linearised Euler solver that can interface to CFD software.

Here, the development of an acoustic software module is described which is based on optimised numerical algorithms for the accurate representation of the sound field in an efficient way by solving the linearised Euler equations given the sound sources and the mean flow.

The acoustic module has two aspects of application. First, it can be used on its own with known mean flow and sound sources, and an example of this kind is included in the last section of the chapter. Second, the module can be coupled with a CFD package to study the time-dependent noise generation by oscillating vortex formations inside the flow which is subject of the next chapter.

### 4.1 Numerical Schemes

The accuracy and the efficiency of the acoustic computations depend to a great extent on the numerical scheme used to discretise the sound field. In the previous chapter it was shown that finite volume schemes can only be used in small domains and are also difficult to implement in three dimensions with extended accuracy.

Here *finite difference* numerical schemes of higher order are considered. A new

fully-staggered version for one class of optimised schemes is proposed together with a mirroring procedure that allows the numerical treatment of solid boundaries to remain as simple as with the finite volume approach.

### 4.1.1 DRP schemes

For the numerical solution of the linearised Euler equations (2.11) and (2.12) the optimal discretisation technique in terms of accuracy and efficiency so far appears to be based on the Dispersion-Relation-Preserving (DRP) schemes (Tam and Webb, 1993). Their development and use was reviewed in Chapter 1. Here technical details are included because these schemes are used for the convection terms in the present acoustic module.

The DRP schemes are in fact a set of finite difference schemes for the approximation of spatial derivatives and temporal integrations and for the artificial selective damping of under-resolved high frequencies. All the coefficients involved have been optimised with respect to accurate wave propagation over long distances. They are fully explicit in time, and thus can be easily parallelised. They need a regular Cartesian grid and stepwise representation of solid bodies in the domain.

With the original DRP schemes all variables are stored at the same locations (there is no staggering between pressure and velocity components). The spatial and temporal discretisations are independent of each other and are considered separately.

#### Spatial discretisation

Let  $f(x, y, z)$  be the instantaneous field of any of the acoustic field variables ( $p$  or  $v_i$ , see equations 2.11 and 2.12). For the approximation of the first partial derivative  $\partial f/\partial x$  at any node of a uniform grid, three values of  $f$  to the left and to the right of this node are used (Tam and Webb, 1993):

$$\frac{\partial f}{\partial x} = \frac{1}{\Delta x} \sum_{j=-3}^3 a_j f(x + j\Delta x). \quad (4.1)$$

If the coefficients  $a_j$  are determined from a Taylor expansion of the function  $f$ , sixth order of accuracy can be achieved with these 7 points. In contrast to those ‘standard’ finite difference schemes, with the DRP scheme only *fourth* order requirements are imposed on the coefficients, and the remaining free coefficients are determined in such a way that the Fourier transform of the finite difference scheme on the right of



(4.1) is a close approximation of the Fourier transform of the partial derivative on the left.

Two ways of minimising the approximation error have been considered by the designer of the DRP schemes (Tam, 1997). The first set of coefficients can be found in the original publication (Tam and Webb, 1993). The second approach is more important from the standpoint of wave propagation, and with it the coefficients of the scheme are (Tam, 1997):

$$\begin{aligned} a_0 &= 0, & a_1 &= -a_{-1} = 0.7708824 \\ a_2 &= -a_{-2} = -0.1667059 \\ a_3 &= -a_{-3} = 0.0208431. \end{aligned} \tag{4.2}$$

### Temporal discretisation

From the linearised Euler equations (2.11) and (2.12) the temporal derivatives of the unknown functions can be expressed in terms of the spatial derivatives. Then ‘older’ temporal derivatives can be stored in the computer memory and used in a discretisation scheme in the form:

$$f^{(n+1)} = f^{(n)} + \Delta t \sum_{j=0}^3 b_j \left( \frac{\partial f}{\partial t} \right)^{(n-j)} \tag{4.3}$$

where superscripts denote the time level on a uniform grid along the temporal axis with time step  $\Delta t$ .

The last term on the right side of (4.3) may be regarded as a weighted average of the temporal derivatives at the last 4 mesh points. From the four constants  $b_j$  that are to be selected, three are chosen so that when the terms in (4.3) are expanded in a Taylor series in  $\Delta t$  they match to order  $(\Delta t)^2$ . This leaves one free parameter,  $b_0$ , and the relations for the other coefficients are (Tam, 1997):

$$b_1 = -3b_0 + \frac{53}{12}, \quad b_2 = 3b_0 - \frac{16}{3}, \quad b_3 = -b_0 + \frac{23}{12}. \tag{4.4}$$

The optimal value of the remaining coefficient  $b_0$  is determined by requiring the Laplace transform of the finite difference scheme to be a good approximation of that of the partial derivative. The optimisation depends on two parameters, and their values are selected based on consideration of the range of useful frequency and numerical damping rate. Then the recommended (Tam, 1997) coefficients become:

$$b_0 = 2.302558 \quad , \quad b_1 = -2.491008 \tag{4.5}$$

$$b_2 = 1.574341 \quad , \quad b_3 = -0.385891 .$$

This scheme is proved (Tam, 1997) to be conditionally stable with sufficiently small time steps.

### Damping of extraneous numerical solutions

In a realistic sound field there are always ‘short waves’ that cannot be resolved properly on the mesh chosen, no matter how fine it is made. Since those waves excite dispersive and parasitic waves in the discretisation schemes, they have to be filtered out of the numerical simulation.

This is done by introducing artificial selective damping (Tam, 1997) into the momentum equations (2.12). The dissipation term that is added to the right-hand side, for example, of the first momentum equation has the form:

$$D_i = -\frac{\nu_{art}}{(\Delta x)^2} \sum_{j=-3}^3 d_j u_{i+j}. \quad (4.6)$$

Here  $u_i$  is the velocity component in the  $x$  direction at node  $i$  of the mesh,  $\nu_{art}$  is a coefficient of artificial viscosity, and with the sum the neighbouring nodes  $j$  along  $x$  are involved.

The coefficients  $d_j$  are selected after analysis of the Fourier transform of the momentum equation with the damping term in a way that ensures small damping for the long (resolved) waves and large damping values for the short (unresolved) waves. One set of recommended (Tam, 1997) coefficients is:

$$\begin{aligned} d_0 &= 0.2873928425, & d_1 &= d_{-1} = -0.2261469518 & (4.7) \\ d_2 &= d_{-2} = 0.1063035788, & d_3 &= d_{-3} = -0.0238530482. \end{aligned}$$

### Solid boundary treatment

Regular Cartesian grids are needed to achieve optimal wave-propagation properties of the differencing schemes. Hence, the solid boundaries are discretised in a stepwise manner, and boundary velocity components perpendicular to the wall are set to zero.

The non-staggered mesh leads to the need of defining ‘ghost points’ into the solid body in order to satisfy both the boundary conditions and the differential equations at the solid wall (Tam and Dong, 1993). For the pressure variable, one layer of non-physical nodes are considered behind the solid wall, and their values are determined

from the boundary condition of zero pressure gradient at the wall. At the same time, for the pressure nodes on the solid surface the usual differential equation (2.11) is solved.

Symmetrical differencing stencils as those described above cannot be used near boundaries. Instead, asymmetrical schemes with DRP properties have been derived and implemented in two dimensions (Tam, 1997).

In three dimensions where solid bodies of irregular shapes are present, the ghost points become inconvenient to handle. Apart from a map of the surfaces, a record of the type of difference equation to be used with each ghost point is needed, as well as records of the non-symmetric difference schemes for the points next to the solid boundary. All these make the final code less efficient than it could be.

### 4.1.2 Staggered DRP-type schemes

Staggering of the computational meshes for the scalar and vector variables of the linearised Euler equations (2.11), (2.12) is proposed to simplify the solid boundary treatment and to extend further the efficiency and the accuracy of the solver.

As with finite volumes (Djambazov et al., 1997c) the computational domain is divided into cells. The stepwise solid boundaries consist of cell faces. Scalar quantities, such as pressure, are stored in the centres of the cells while velocity components are stored at the cell faces in the *middle* of the time steps.

Equations (2.11), (2.12) consist of transient, convection, propagation and source terms. They can all be integrated over a time step so that the transient term becomes a simple difference, and the remaining convection and propagation integrals need a higher order approximation. (The source terms are less likely to need it.)

It is suggested that the DRP scheme (Tam and Webb, 1993) is used with its differentiation and temporal integration steps for the *convection* integrals only. Spatial derivatives involving 7-point stencils at 4 older time steps are needed for the calculation of the new time step. (Of course, only the necessary derivatives are stored to save computational time and memory.)

The propagation terms  $\bar{\rho}c^2 \frac{\partial v_j}{\partial x_j}$  and  $\frac{1}{\bar{\rho}} \frac{\partial p}{\partial x_i}$  are computed with 6-point stencils for the spatial derivatives exploiting the staggered grid. Optimised coefficients can be determined that exhibit the properties of the DRP schemes. The temporal integration is also optimised for long simulation of sinusoidal waves.

All computational stencils used are symmetrical and they will remain the same

throughout the domain. At solid boundaries mirroring of the values provides the missing points; thus solid walls are modelled by symmetry planes making use of the staggered grid. The resulting computer code is much simpler to use and to maintain than the corresponding non-staggered one.

The coefficients for spatial derivation and for temporal integration of the staggered propagation terms of the linearised Euler equations will be obtained separately in this section.

### Spatial Derivatives

With the staggered mesh, the first partial derivatives of the pressure and of the three velocity components along the three spatial axes (see equations 2.11 and 2.12) have to be approximated in the middle of the intervals along the axes of derivation.

With the mirroring procedure at solid boundaries, and with a suitable radiating boundary procedure (both described in the next section), the full 6-point stencils can be considered available everywhere in the computational domain. Then along  $x$  the first derivative of a function  $f$  can be expressed as

$$\frac{\partial f}{\partial x} = \frac{1}{\Delta x} \sum_{j=-2}^3 a_j f \left[ x + \left( j - \frac{1}{2} \right) \Delta x \right] \quad (4.8)$$

Requiring the scheme to be exact up to *fourth* order imposes a number of restrictions on the coefficients  $a_j$  leaving only one free parameter:

$$\begin{aligned} a_{1-j} &= -a_j, \quad j = 1 \dots 3 \\ a_2 &= \frac{1}{2} \left( \frac{25}{24} - a_1 \right) \\ a_3 &= \frac{1}{10} \left( a_1 - \frac{9}{8} \right) \end{aligned} \quad (4.9)$$

An optimal value can be found for the remaining coefficient  $a_1$  with respect to the wave propagation properties of the numerical scheme. One way to achieve this is to require the differencing scheme to be exact for the function

$$f(x) = c_0 + c_1 \sin kx + c_2 \cos kx \quad (4.10)$$

for any value of the constants  $c_0, c_1, c_2$ . This leads to a parametric equation for  $a_1$ :

$$\left( s_1 - \frac{1}{2}s_3 + \frac{1}{10}s_5 \right) a_1 = \alpha - \frac{25}{48}s_3 + \frac{9}{80}s_5 \quad (4.11)$$

where

$$s_1 = \sin \alpha, \quad s_3 = \sin 3\alpha, \quad s_5 = \sin 5\alpha$$

with a parameter

$$\alpha = k \frac{\Delta x}{2} = \frac{\pi}{N_w}$$

where  $N_w$  is the number of points per wavelength for this particular frequency. The range of values that coefficient  $a_1$  can take is shown in Figure 4.1. With the increase of the number of points per wavelength the value of coefficient  $a_1$  becomes less sensitive. Of course, since the computational cost also increases, one should make a reasonable choice around 20–25 points for the typical wavelengths expected.

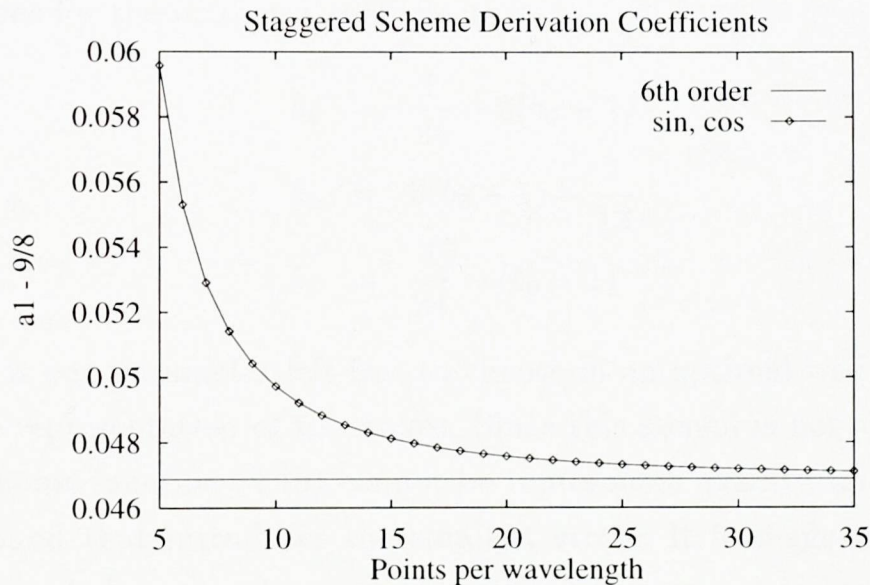


Figure 4.1: Solution of the parametric equation for  $a_1$

It can be seen from Figure 4.1 that there is no ideal value of the unknown coefficient  $a_1$ . The optimum value of  $a_1$  will depend on the spectrum of the sound field studied. If only one frequency is present, it can be resolved with 6–7 points per wavelength, and  $a_1$  can be tuned accordingly. This is, however, a risky choice because if resonance is possible, the numerical noise which involves other frequencies will be amplified. When a range of frequencies is expected, the spacing of the computational mesh should not be less than 15–16 points per wavelength for the highest of them, and  $a_1$  should be chosen accordingly from Figure 4.1. In this way the differencing scheme (4.8) will remain as close to the exact representation as possible.

### Temporal Integration

With the time staggering of the velocity components the propagation terms of (2.11) and (2.12),  $\bar{\rho}c^2 \frac{\partial v_j}{\partial x_j}$  and  $\frac{1}{\rho} \frac{\partial p}{\partial x_i}$ , are evaluated in the middle of each time step. These values can be used to calculate the new pressure and velocities at the end of their time steps if the following approximate integration is applied:

$$\int_{-\frac{\Delta t}{2}}^{\frac{\Delta t}{2}} f(t) dt = \Delta t \sum_{m=0}^3 b_m f(-m\Delta t). \quad (4.12)$$

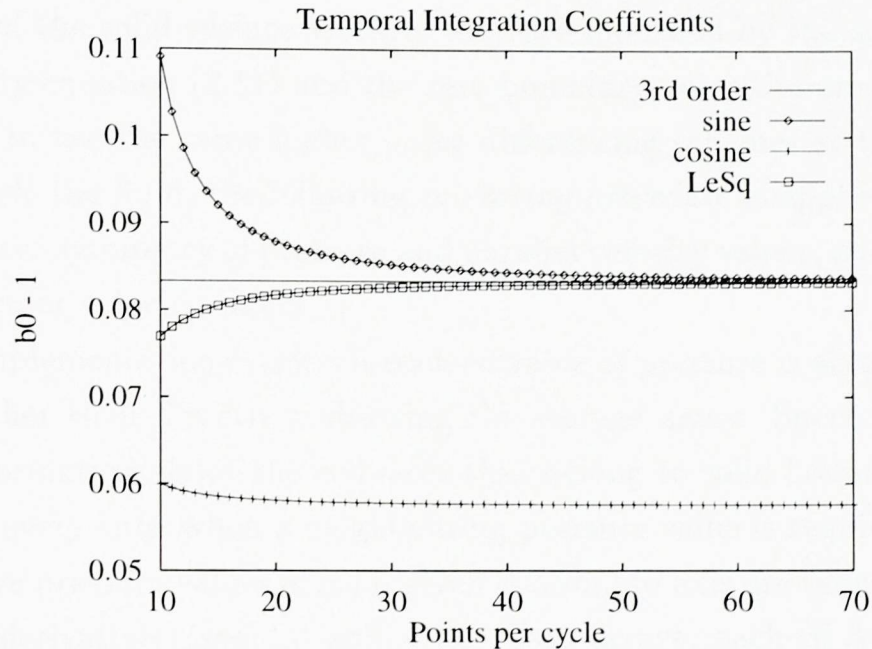
Requiring this numerical scheme to be exact up to second order provides the following restrictions for the unknown coefficients  $b_m$ :

$$\begin{aligned} b_1 &= \frac{1}{24} - 3(b_0 - 1) \\ b_2 &= 3(b_0 - 1) - \frac{1}{12} \\ b_3 &= \frac{1}{24} - (b_0 - 1). \end{aligned} \quad (4.13)$$

Again there is one parameter left free to choose in an optimal way with respect to the accurate representation of the waves. Since this stencil is not symmetrical, the general harmonic function (4.10) cannot be represented exactly, and a compromise has to be found that introduces the smallest error. It is suggested that a least-square fit is made between the solutions of the two parametric equations obtained separately with the sine and the cosine term of (4.10). This is done in order to minimise the approximation error of the scheme assuming that the sine and cosine terms will have equal amplitudes in a real sound signal. The result of this fit is plotted on Figure 4.2 under the title “LeSq”.

With more than 20 points per cycle the optimal value of the free coefficient  $b_0$  does not change much and is very close to the value corresponding to a third-order scheme. Because of the Courant limit for explicit time integration most applications are likely to have not less than 20 points per cycle, and  $b_0$  should be selected from a point with abscissa greater than 20.

This temporal integration scheme is of lower formal order than the classical fourth order Runge-Kutta formula for ordinary differential equations which can also be applied to the linearised Euler equations (Zingg et al., 1996; Hixon, 1996). Since the time step has to be small enough for the stability of the explicit scheme, the

Figure 4.2: Parametric solutions for  $b_0$ 

third-order temporal scheme does not decrease the overall accuracy determined by the spatial differencing scheme. The advantage of the DRP-type scheme is that it does not imply evaluation of spatial derivatives other than those already computed at older time steps.

## 4.2 Boundary Conditions

### 4.2.1 Solid boundaries

Since the linearised Euler equations (2.11) and (2.12) do not take into account viscous effects, solid walls can be modelled by symmetry planes. For stationary surfaces this means that the velocity component perpendicular to the wall is zero, and that the normal derivatives of pressure and of the other two velocity components are also equal to zero at the solid boundary.

Solid bodies in the computational domain are discretised in a stepwise manner (see top part of Figure 4.7 and also Figure 3.4). The cells with centres inside the body are considered as internal cells, they are declared blocked and are not processed.

The solid walls are considered as comprised of cell faces, and the corresponding velocity components at these faces (with the staggered mesh, those are always components perpendicular to the faces) are set to zero.

If parts of the solid surface vibrate, they are modelled by the source term  $S$  of the continuity equation (2.11) and the zero boundary velocities are retained.

In order to use the same higher order differencing schemes at the solid surface as those inside the fluid, the following *mirroring procedure* is applied to every solid boundary face: symmetry of pressure and parallel velocity values, and antisymmetry of perpendicular velocity values.

In the implementation every cell-centred value of pressure is accessed via a function call rather than directly addressing the storage array. Special integer arrays hold the information about the cell faces that belong to solid boundaries, and they are checked every time when a neighbouring pressure value is required.

Successive pressure values along a given coordinate axis are required to compute the spatial derivatives (see 4.8 and 4.1). Two arrays, each of 3 elements, have to be defined: one to the left, and one to the right of a given centre point or face, respectively. Starting from the centre specified outwards, the pressure values are copied into the auxiliary 3-element arrays. If a solid face is encountered, the advancing direction is reversed. This provides a mirror image of the points in front of the wall for the missing points behind the wall which is equivalent to setting zero normal derivatives with both differencing schemes (4.8) and (4.1).

A listing of the mirroring procedure along the  $x$  axis (east-west direction) is shown in Figure 4.3. The same subroutine is used to obtain mirrored values of velocity components that are parallel to the wall. (At the solid surface the resultant derivative will be multiplied by a zero mean velocity, but this may not be the case with the neighbouring cells away from the wall which also involve 7-point stencils.)

For the mirroring of the velocity components perpendicular to the wall, a similar procedure has been implemented. The only difference is that the sign of the copied elements is reversed together with the advance direction every time a wall is found.

The six mirroring subroutines (two along each axis) have been implemented in three dimensions. They can be used without changes with 2D and 1D problems. Then there will be a small overhead with the reflections along the unused axes.

The mirroring approach described above makes the solid boundary treatment of the higher order finite difference schemes as simple as with finite volume algorithms. It is only possible with staggered storage of the pressure and the velocity components.



```

subroutine pEW(p, iC, j, k, pvec)
C   Get problem dimension limits from include file:
    include 'n.cmn'
C   Define argument arrays (implicit integers with i,j,k),
C   p, iC, j, k - input ; pvec - output:
    dimension p(0:idim, 0:jdim, 0:kdim), pvec(-3:3)
C   Global solid boundary maps:
    common /walls/ keepu(0:idim, 0:jdim, 0:kdim),
+           keepv(0:idim, 0:jdim, 0:kdim),
+           keepw(0:idim, 0:jdim, 0:kdim)

    pvec(0) = p(iC, j, k)
    do 9 idir = -1, 1, 2
        i = iC
        inc = idir
        do 9 nb = 1, 3
            look = i
            if ( inc .LT. 0 ) look = i - 1
            if ( keepu(look,j,k) .GT. 0 ) then
                inc = -inc
            else
                i = i + inc
            endif
            pvec(idir*nb) = p(i,j,k)
9        continue
    end

```

Figure 4.3: Mirroring of cell-centred values along  $x$ 

## 4.2.2 Radiating boundaries

In conjunction with the DRP schemes, radiation boundary conditions have been developed in two dimensions based on the asymptotic solutions of the linearised Euler equations (Tam and Webb, 1993). They involve optimised backward finite differences using 7-point stencils close to the outer boundaries of the domain. Since their software implementation is long and complex, an attempt is made here to design a less accurate but much simpler set of acoustic radiation boundary conditions.

On a regular Cartesian grid, a regular computational domain is considered (rectangular box). All sources of sound are assumed to be well inside the box, and therefore, the acoustic waves leaving the box may be considered locally (over each cell) as *plane* waves.

These ‘radiating’ boundaries (the outer boundaries of the domain) have no physical meaning, since the natural boundary condition of the external aeroacoustic problem is zero oscillation at infinity. Numerical radiating boundary conditions have to be defined in order to simulate the acoustic waves leaving the finite computational

domain. This can be done by considering the analytical solutions of the linearised Euler equations (2.11 and 2.12, in a form similar to 3.2) without any sources of mass or momentum:

$$\begin{aligned}\frac{\partial p}{\partial t} + \bar{v}_j \frac{\partial p}{\partial x_j} + c \frac{\partial v_j}{\partial x_j} &= 0 \\ \frac{\partial v_i}{\partial t} + \bar{v}_j \frac{\partial v_i}{\partial x_j} + c \frac{\partial p}{\partial x_i} &= 0.\end{aligned}\tag{4.14}$$

In the presence of a *uniform* mean flow with velocity components  $\bar{v}_1$ ,  $\bar{v}_2$  and  $\bar{v}_3$ , the general solution of these equations for a plane wave normal to the direction  $(\cos \alpha_1, \cos \alpha_2, \cos \alpha_3)$  is

$$\begin{aligned}p &= f[(x_1 - \bar{v}_1 t) \cos \alpha_1 + (x_2 - \bar{v}_2 t) \cos \alpha_2 + (x_3 - \bar{v}_3 t) \cos \alpha_3 - ct] = \\ &= f[x_j \cos \alpha_j - (\bar{v}_j \cos \alpha_j + c)t], \\ v_1 &= p \cos \alpha_1, \quad v_2 = p \cos \alpha_2, \quad v_3 = p \cos \alpha_3, \\ \cos^2 \alpha_1 + \cos^2 \alpha_2 + \cos^2 \alpha_3 &= 1\end{aligned}\tag{4.15}$$

with  $f$  denoting any function of the argument in brackets. This can be verified by substituting (4.15) into (4.14) where the subscript summation convention is used.

The solution (4.15) may be used to define a boundary radiation formula based on interpolation from inside the domain at the previous (old) time step. Considering a local frame of reference with origin at a given boundary node ( $x_j = 0, j = 1 \dots 3$ ) at the old time level ( $t = 0$ ) and a plane wavefront that intersects the  $x_i$  axis at the origin at time  $t = \Delta t$ , the 'old' intersection (at  $t = 0$ ) can be found of the same wavefront with the same axis  $x_i$ , and it will have coordinates

$$\begin{aligned}x_j &= 0, \quad j \neq i \\ x_j &\neq 0, \quad j = i.\end{aligned}$$

Since  $f$  is any function, it can be assumed that the wavefront that is being traced has an argument of zero. Thus the equation for the internal intersection point becomes

$$x_i = \frac{\bar{v}_j \cos \alpha_j + c}{\cos \alpha_i} \Delta t.\tag{4.16}$$

The boundary value at time  $t = \Delta t$  can be determined by means of simple one-dimensional interpolation along the  $x_i$  axis from the 'old' values at time  $t = 0$  with the interpolation point defined by (4.16).

In order to use one-and-the-same computational stencil for the spatial derivatives inside the domain and near the outer boundaries, three layers of interpolated nodes are needed. With the present implementation second order interpolation is used:

$$f(x) = f_0 + \frac{x}{2\Delta x} \left[ f_R - f_L + \frac{x}{\Delta x} (f_L + f_R - 2f_0) \right] \quad (4.17)$$

$$f_L = f(-\Delta x), \quad f_0 = f(0), \quad f_R = f(\Delta x).$$

This makes false reflections of magnitude about 1–2 % of the amplitude with plane waves perpendicular to the boundaries. For long-time simulations, and especially with resonance, a full implementation is needed with asymmetrical differencing stencils at boundaries and with only one layer of interpolated boundary nodes.

The direction of the plane waves defined by  $\cos \alpha_j$ ,  $j = 1 \dots 3$  is a free parameter, and has to be specified by the analyst for each boundary cell for each model solved.

In the implementation of the acoustic module a facility is provided for the calculation of the cosine values and the resultant interpolation coordinates on the basis of ‘source points’ (origins of spherical waves) that can be different for each boundary cell. Thus complex wave patterns resulting from interference of primary and reflected (by bodies of complex shape) waves can be taken into account. In simplified simulations, one and the same set of source coordinates can be assigned to all boundary cells.

There are cases when the direction of the plane waves leaving the domain at a given boundary point varies with time in a periodic manner. (One such example will be discussed in the last section of this chapter.) Then the best strategy is to specify an average direction of radiation on an intuitive basis. This is usually accompanied by enlarging the computational domain so that the deviation from the average direction becomes smaller, but this is very expensive in three dimensions.

Several attempts were made to define a procedure for the automatic time dependent calculation of the local radiation axis but they all generated instability in the solution.

### 4.3 Validation Tests

The computer code implementing the combined DRP and staggered-mesh linearised Euler solver with mirroring procedures at solid walls is at first tested in one and two dimensions on problems that have exact analytical solutions.

In Figure 3.15 a finite volume solution for one-dimensional waves with mean flow showed maximum numerical error of about 10%. Here, the same simulation is repeated with the finite difference acoustic module.

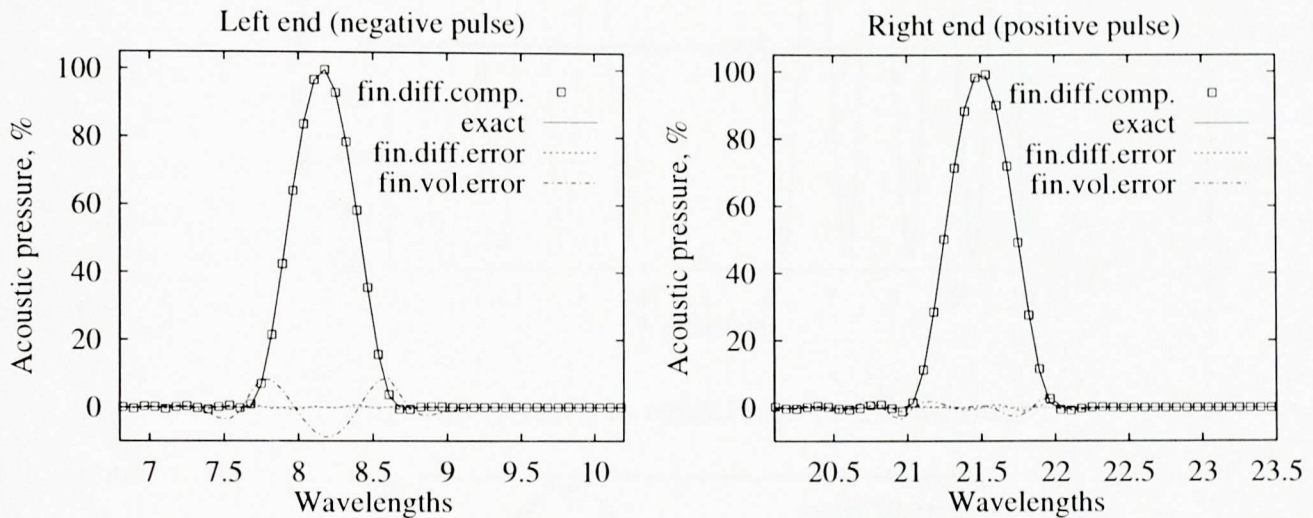


Figure 4.4: Updated test solution from Figure 3.15

As it can be seen in Figure 4.4, the finite difference error is less than 1%, and it stays the same with both waves (the positive wave propagates in the direction of the mean flow, while the negative wave propagates opposite to the mean flow; the Mach number is 0.2 as before). This very small error was an expected result because the finite difference schemes used in the acoustic module have been optimised for wave propagation in the presence of mean flow.

Two more severe tests are included below which were not attempted with finite volumes because of their limited accuracy.

Figure 4.5 contains graphs of the one-dimensional pressure distribution after a certain number of reflections (shown next to each line key) of an initial pulse containing four sinusoidal waves. The wave packets that have travelled 5, 15 and 25 times the length of the domain appear on the right-hand side of the plot, while the ones that have travelled 10 and 20 lengths appear on the left. In the middle, false oscillations can be seen that are due to the discontinuity of the first derivative of the initial pulse.

No artificial viscosity was introduced for this example in order to test the damping properties of the staggered scheme.

Apart from the zone around the discontinuity, almost no accumulation of error can be observed.

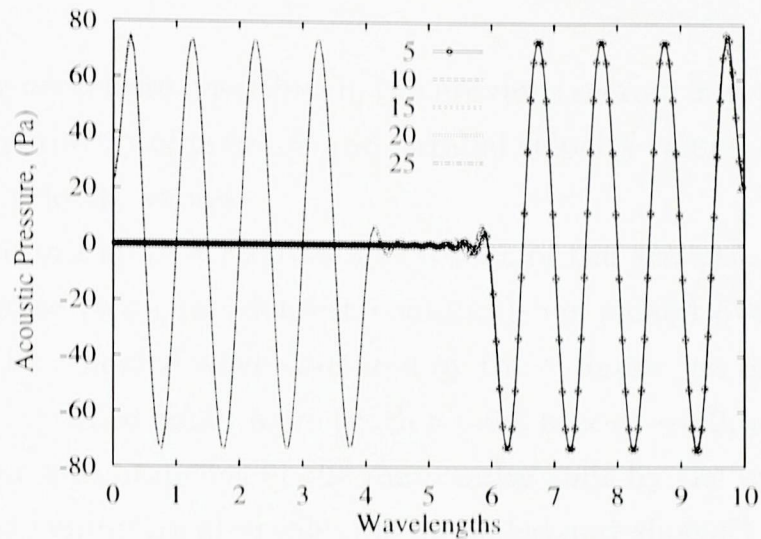


Figure 4.5: Multiple reflections in 1D domain

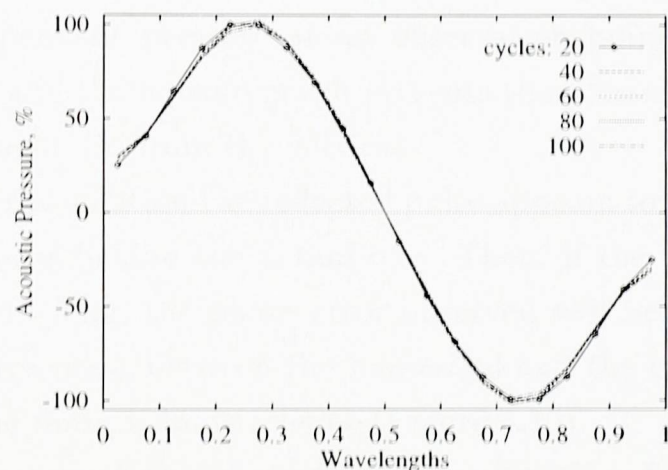


Figure 4.6: Resonant reflections of sine pulse

Figure 4.6 shows one of the most difficult tests for numerical wave simulation: resonance over 1 wavelength of an initial sinusoidal wave. It can be seen that after 100 reflections the solution is quite acceptable.

Curved solid boundary representation was tested on one of the benchmark problems from a Computational Aeroacoustics Workshop (Category 1, Problem 2) (Tam and Hardin, 1997). The finite volume solution of this problem (Figure 3.10) showed some deviations from the exact solution. Here the problem is tackled again with the finite difference acoustic module.

A two-dimensional initial pressure pulse originating at  $(x = 4, y = 0)$  is scattered by a solid cylinder as it propagates outwards. The cylinder is discretised in a stepwise manner (see top part of Figure 4.7), its internal cells are declared blocked and are

not processed.

The mirroring procedure described in the previous section is applied to every solid boundary face: symmetry of pressure and parallel velocity values, and antisymmetry of perpendicular velocity values.

The top graph in Figure 4.7 shows a snapshot of the acoustic pressure field just after the main pulse (with the densest contours) has passed over the cylinder (at  $x = 0, y = 0$ ). The reflected wave scattered by the cylinder can be seen on the right side of the graph. A third small wave (with a peak at  $x = -0.9, y = 1$ ) results from the collision of the two branches of the main pulse split by the cylinder. (Only the upper part of this symmetrical problem is modelled and shown.)

The tabulated analytical solution (Kurbatskii, 1997) (courtesy of K. Kurbatskii) was used to produce the two graphs at the bottom of Figure 4.7. The middle graph shows the time dependent pressure at an observation point located 5 diameters above the cylinder, and the bottom graph presents the pressure history at the same distance along a line at  $45^\circ$  from the vertical.

With the numerical solution the reflected pulse appears to be less than 0.02 non-dimensional time units behind the actual one. Then, if the ‘cycle’ of this pulse is assumed to be 1 unit long, the phase error observed will be less than 2%. It can be seen that the agreement between the numerical and the exact solution is much better than with the finite volume scheme (Figure 3.10).

All problems presented here have been solved with a fixed set of staggered-scheme coefficients based on the values for 20 points from Figure 4.1 and for 50 cycles from Figure 4.2. If the frequency of the signal studied is known, fine tuning of the coefficients can be done using the values from the above graphs.

For the convection terms the DRP coefficients (Tam and Webb, 1993) are also constant, and the artificial viscosity coefficient was set to zero relying on the stabilising property of the staggered terms (this should not be done in the supersonic case where the convection terms prevail).

The development of the present acoustic module was described in a paper (Djambazov et al., 1998c) prepared for a joint Aeroacoustics Conference of the American and European Aeronautical Societies.

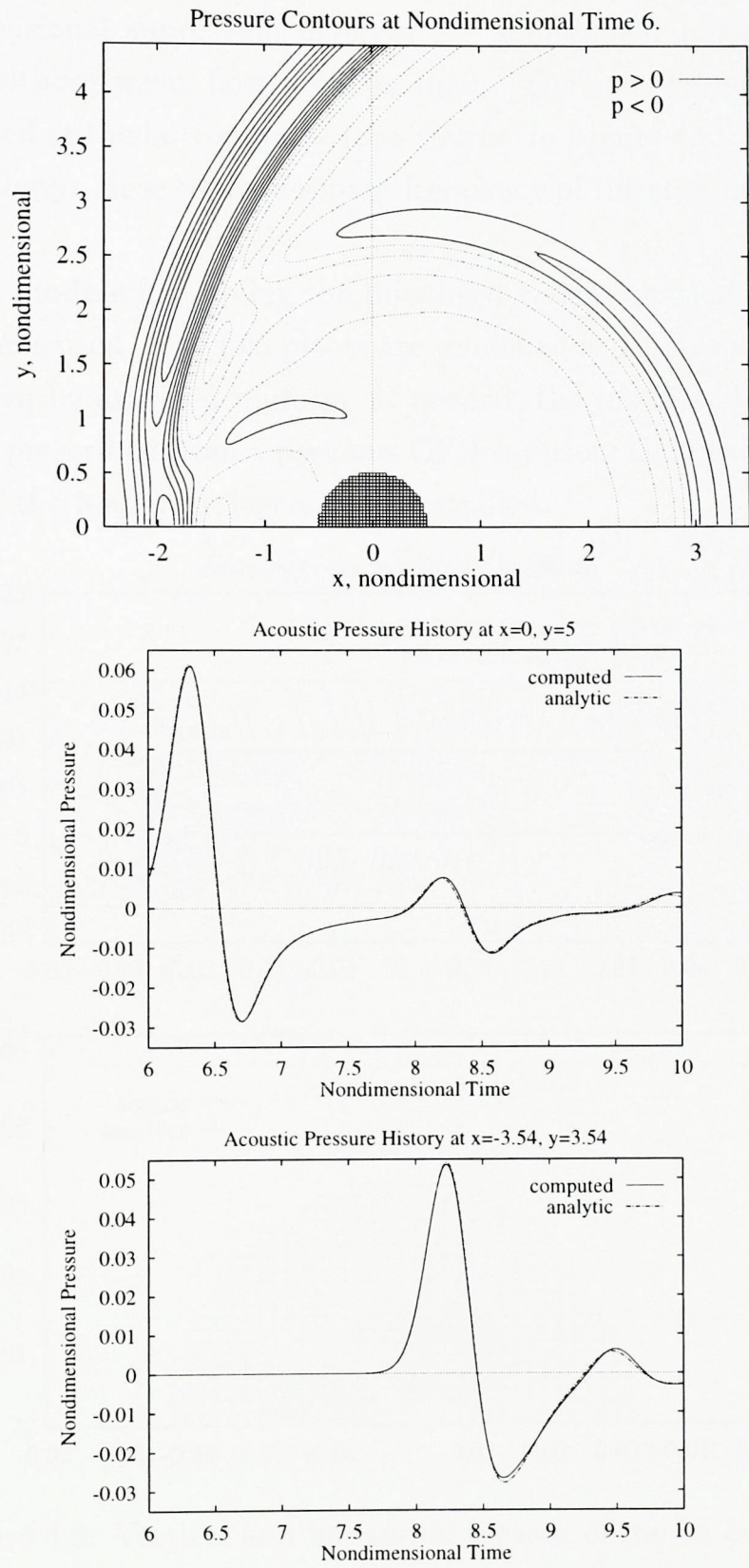


Figure 4.7: Acoustic scattering test

## 4.4 A 3D Application: Resonant Cavity

This three-dimensional simulation involves the sound field between two circular plates with or without mean flow between them. The sound source is a vibrating diaphragm located at the bottom plate (see ‘source’ in Figure 4.8), and its frequency can vary but is always close to the resonant frequency of the standing waves between the two plates.

The acoustic module for solving the linearised Euler equations described in the previous sections is used. The two plates are modelled as thin solid surfaces so that the mean flow can be assumed uniform. If needed, the module allows non-uniform mean flow to be prescribed from a previous CFD solution; then the effect of the real geometry and of the boundary layers can be studied.

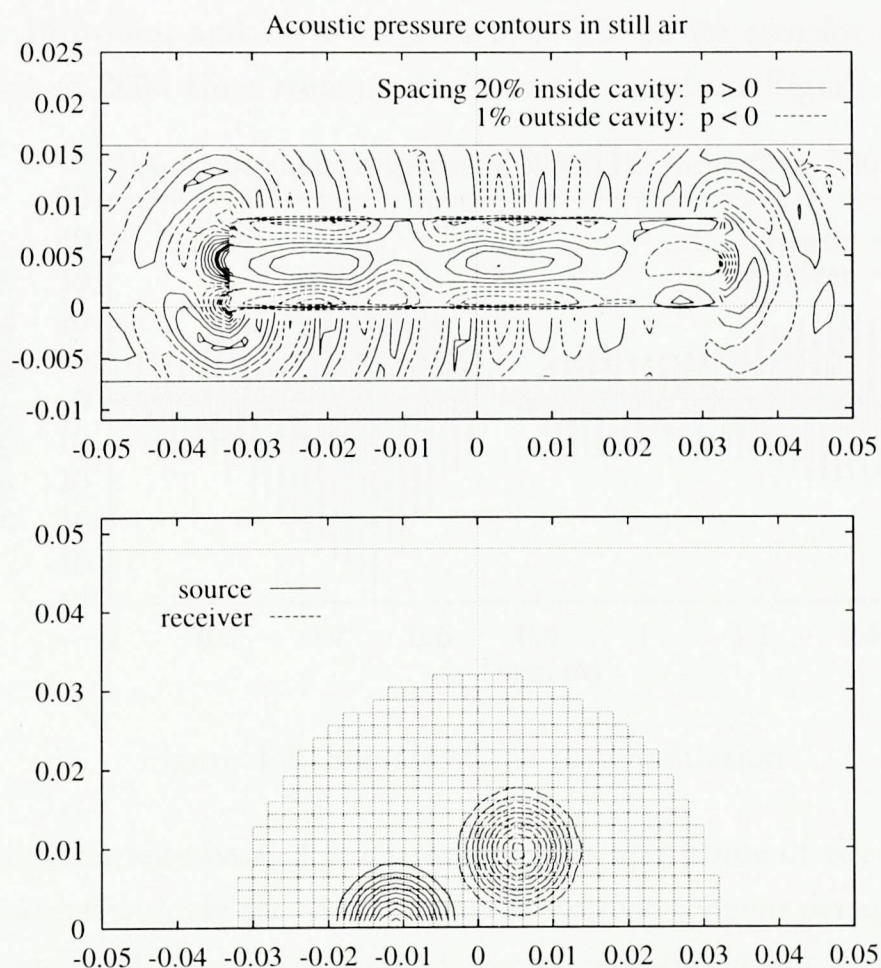


Figure 4.8: Vertical and horizontal section of model domain

The geometry of the resonant cavity and the outer boundaries of the computational domain are shown in Figure 4.8. The ‘south’ boundary (lower boundary on bottom graph) is a symmetry boundary (zero flux, equivalent to a solid wall with



the linearised Euler equations), and the rest of the domain boundaries are radiating acoustic boundaries.

In the direction perpendicular to the plates the minimum possible number of 12 cells per wavelength is prescribed which can ensure correct representation of the multiple reflected waves; in the other two directions the mesh is coarser (about 6 points per wavelength, see dotted lines in the bottom part of Figure 4.8). The variable intensity of the source membrane is shown by solid-line contours, and the variable sensitivity of the receiver region is plotted in broken lines. The amplitude and the phase of the received signal (weighted integral of pressure over the marked region) has to be compared to the amplitude and the phase of the source signal for different values of the source frequency.

To test the numerical accuracy and stability of this model, the source is turned on for about 16 cycles, and then the system is left on its own for about 48 more cycles (a total of 2000 time steps). As it can be seen on Figure 4.9, the initial

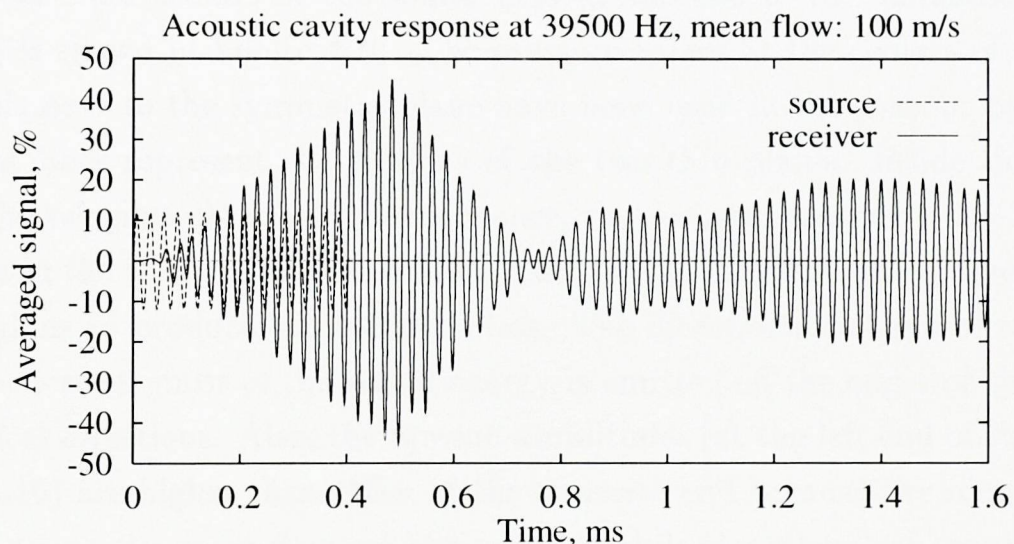


Figure 4.9: Signals of the test simulation

linear increase (characteristic of resonance) in the amplitude of the received signal is captured correctly. After the source signal is stopped a linear decay of the receiver pressure follows until most of the sound energy is radiated out of the domain (at about 0.75 ms). However, some higher frequency oscillations can be observed that are superimposed on the main signal. During the second half of this test simulation (at time greater than 0.8 ms) what has been recorded is, in the author's opinion, self noise of the numerical schemes. Artificial damping was specified (using a fifth order upwind scheme rather than the central DRP scheme for the convection terms)

in order to obtain a stable solution with nonzero mean flow.

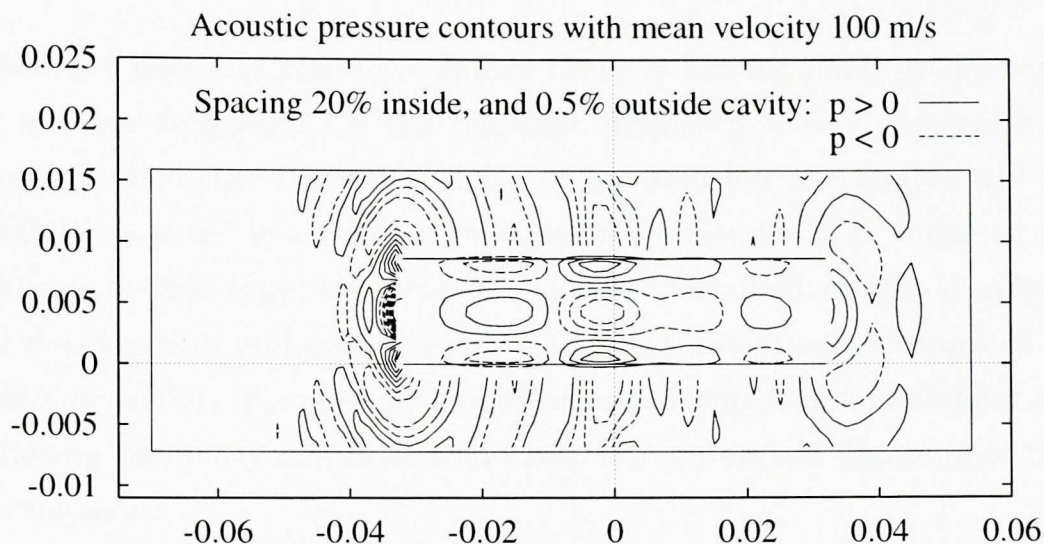


Figure 4.10: Pressure field at symmetry plane

A vertical cross section of the sound field at the end of the simulation (5000 time steps) is shown in Figure 4.10. The pressure values at the centres of the first layer of cells next to the symmetry plane have been used in the contour plot. The two straight lines represent the sections of the two thin plates. Inside the cavity (between the two plates), due to the resonance, the acoustic pressure is 10–20 times higher than in the radiated waves outside. That is why different scales were used in the two regions to produce the contour plots. The directional pattern of radiation can be clearly seen: most of the sound energy is emitted off the edges of the plates in the vertical directions. Also, the upwind amplitudes (at the left end of the plates in Figure 4.10) are higher than those at the opposite end because the sound waves travelling against the mean flow get compressed, while the others are expanded.

The wave pattern in and around the cavity without mean flow can be seen in the top part of Figure 4.8. This is a snapshot of the pressure field after the first 500 time steps with the source signal switched on. Since the source membrane is closer to the left edges of the plates, the amplitude of the radiated sound there is higher as indicated by the denser contours.

A complex three-dimensional pattern of oscillation inside the cavity is produced by the simulation. When an acoustic (compression) wave reaches the end of a duct and starts radiating into the open space, a reflected (depression) wave is formed which will eventually reach the opposite end of the cavity. This process goes on in all directions with all waves interfering. Obviously the simulation captures some of

these effects; the question is what part of the result is induced by false numerical noise?

In closed (or partially closed) volumes there is always likely to be resonance at one or another frequency. If the resonant frequency is not physically present in the signal studied, the numerical noise of the solution procedures will provide it. (By definition, noise is a random mixture of frequencies.) In order to be used for simulations of this type, the acoustic module described in this chapter needs additional development and comparisons with exact analytical solutions of simpler cases of *duct acoustics*. Possible directions of improvement include higher order of stable radiating boundary conditions and selective numerical damping of the non-physical resonance.

# Chapter 5

## Coupling CFD with Acoustics

As it was already explained in Chapter 1, the flow needs to be computed separately from its acoustic perturbations because of the differences in the numerical algorithms. In this chapter some possibilities are discussed, and two examples are given of using *time dependent* flow simulation results to calculate the aerodynamic sources of sound needed for the acoustic simulation.

### 5.1 Aeroacoustic Sources on Solid Surfaces

Aerodynamic sound is generated as a result of the interaction of vortex structures that arise in viscous flows. These vortex structures are most often associated with either a shear layer or a solid surface. Once the sound is generated it propagates in the surrounding non-uniformly moving medium and travels to the ‘far field’.

Sound propagation is hardly affected by viscosity (that is why noise is so difficult to suppress). Also, sound perturbations are so small that their contribution to the convection velocity of the flow is negligible in many cases. These two facts mean that sound propagation is, in essence, described by the linearised Euler equations (2.11) and (2.12).

The simulation of the flow that generates sound, however, requires time accurate solutions of the Navier-Stokes equations. Two approaches exist here: Reynolds Averaging and Large Eddy Simulation. Both of them require adequate turbulence models and fine meshes to capture the small structures in the flow that oscillate and generate sound.

Most commercially available Computational Fluid Dynamics (CFD) codes have

implementations of Reynolds-Average Navier-Stokes solvers (RANS). The new alternative, Large Eddy Simulation (LES), which requires more computational power has become available only in the recent years. In the author's opinion, the future of Computational Aeroacoustics (CAA) is closely related to LES. For the time being, however, one should try to make the most of RANS.

The basic idea of software coupling between CFD and CAA (decomposition of the variables into flow and acoustic parts) as well as the Domain Decomposition into near field and far field was presented in Chapter 2. The CFD code is used to solve the time-dependent RANS equations in the near field while the CAA deals with the linearised Euler equations in the whole domain.

Here, the study concentrates on the use of the source term  $S$  (2.11) to transfer the information about the generation of sound from the CFD code to the acoustic solver.

In Chapter 1 test solutions of 1D sound waves were presented, illustrating the poor propagation properties of the CFD algorithms (see Figure 1.1). A closer examination of the time history of this solution (PHOENICS, 1995) pictured in Figure 5.1 reveals that the pressure at the *first* node next to the source of sound (at the left end of the 1D domain) has been resolved accurately.

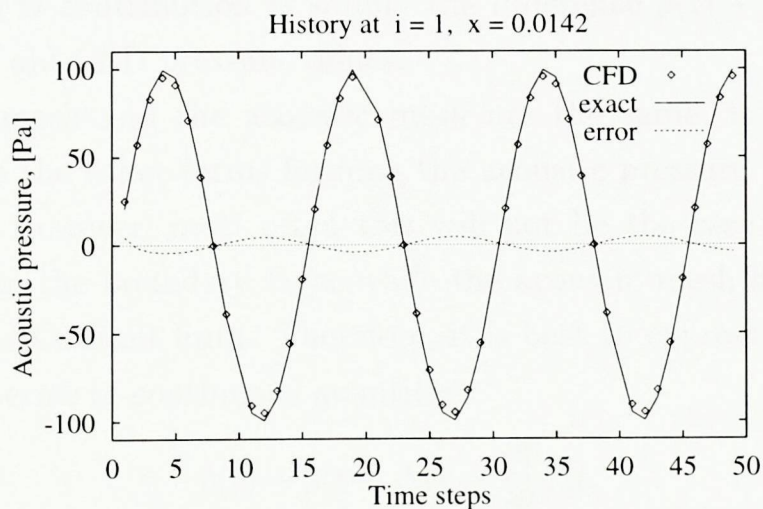


Figure 5.1: CFD solution at source of sound

This suggests that when the source nodes are known, the time dependent pressure at these nodes may be used to calculate the necessary source term. The following *assumption* has to be made: the CFD code resolves the *full* physical pressure (comprised of mean flow and acoustic components) in the first layer of computational

cells next to a solid surface or in any other cells that have been identified as sound sources. The term ‘resolves’ is used here to denote that the CFD pressure is a good approximation of the true pressure signal in these selected cells.

Since the CFD pressure signal contains a mean-flow component, it cannot be fed directly into the acoustic code; the time dependent component has to be separated first. This can be done if a preliminary *steady* CFD solution is obtained in the same geometry, and the time dependent simulation is started with this initial condition. Then the difference between the time dependent and the steady pressure is the signal that has to enter the acoustic simulation at the prescribed source nodes.

One way of inserting this signal into the linearised Euler solver is to specify it as a fixed-value internal condition at the selected nodes. However, this will preclude the possibility of other acoustic waves (reflected from solid boundaries or generated by neighbouring source nodes) to propagate through the prescribed source layers.

The other option is to calculate the contribution of the CFD source to the local increment of pressure

$$\Delta p = \frac{\partial p}{\partial t} \Delta t$$

at the selected nodes over each time step. Since any transients associated with the establishment of the mean flow have been eliminated by starting from a steady solution, this CFD contribution is simply the difference  $\bar{p}(t) - \bar{p}(t - \Delta t)$  between the new and the old CFD pressure values.

If the CFD mesh and the acoustic mesh are the same, it is enough to add this difference to the other terms forming the acoustic pressure increment  $\Delta p$  (see equation 2.11). However, most often this will not be the case because the CFD mesh is refined in the boundary layer while the acoustic mesh has to be coarse in order to obey the Courant limit. Therefore, it is best to express the CFD pressure contribution in terms of continuous quantities:

$$\bar{p}(t) - \bar{p}(t - \Delta t) = \frac{\partial \bar{p}}{\partial t} \Delta t. \quad (5.1)$$

Then the temporal derivative of the local pressure at the source nodes, calculated from the CFD solution, can be added to the source term  $S$  of the acoustic continuity equation (2.11):

$$S = \frac{\partial \bar{p}}{\partial t} + S_{vib}. \quad (5.2)$$

Here  $S_{vib}$  denotes sources external to the flow like vibrating solid objects. Thus the following combined algorithm can be outlined:

1. Obtain a steady CFD solution of the flow problem.
2. Start the time-dependent CFD simulation with these initial conditions.
3. Impose the calculated temporal derivative of the pressure at selected nodes within the flow region as part of the source term of the acoustic simulation.
4. Solve the linearised Euler equations in the acoustic domain applying any external sources of mass (vibrating solids).

Since CFD codes use finite volume discretisation, the value of the temporal derivative  $\partial\bar{p}/\partial t$  has to be recovered from the same type of discretisation. Then, if phase accuracy of the calculated acoustic signal is essential (like with resonance), the time-dependent outflow from the control volume with increasing  $\bar{p}$  in the CFD solution or the inflow perturbation if  $\bar{p}$  is decreasing should also be taken into account in order to calculate the correct amount of mass that is assumed to enter or exit the acoustic simulation at each time step.

The finite volume form of the RANS continuity equation (1.1) in isentropic conditions ( $\frac{\partial\bar{p}}{\partial\rho} = c^2$ ) suggests the following pressure source:

$$S = \bar{\rho}c^2(\bar{v}_j - \overline{v_{j,average}}) \frac{A_j}{\Delta V} + S_{vib} \quad (5.3)$$

$$j = \begin{cases} Influx, \frac{\partial\bar{p}}{\partial t} > 0 \\ Outflow, \frac{\partial\bar{p}}{\partial t} < 0 \end{cases}$$

where  $A_{Influx}$  is the area of the faces of the cell with volume  $\Delta V$  across which there is inflow during the time step  $\Delta t$ , and the repeated index denotes summation over all such faces. This formulation also assumes that the CFD solution is fully converged, since otherwise cell continuity errors would be perceived as sources of sound.

At a solid surface in the region of an aerodynamic sound source the time-dependent velocity ( $\bar{v}_j - \overline{v_{j,average}}$ ) is most likely to have the same sign for all non-blocked faces of a given cell, i.e. there is only inflow or only outflow. Then, according to the continuity equation (1.1) in isentropic conditions

$$\frac{\partial\bar{p}}{\partial t} + c^2 \frac{\partial}{\partial x_j} (\bar{\rho} \bar{v}_j) = 0, \quad (5.4)$$

the sum of the fluxes specified in (5.3) which determines the sound source is equal to the temporal derivative of the RANS pressure. For the cells outside the source region

this cannot be considered true because with propagating waves, the perturbation fluxes through opposite faces have opposite signs. Therefore, it is essential to know in advance which are the source cells.

The computational cells of the first layer next to a solid surface can be considered as prospective sources of sound because the inertial forces from vortex structures (that may be oscillating as they are carried along the surface by the flow) hitting the wall cause pressure fluctuations.

In some cases (separated flows, jets) the sources of sound cannot be localised and are instead distributed across the computational domain. These cases will be discussed in the next chapter with the generation of sound away from solid bodies.

The two codes (CFD and CAA) have separate meshes in overlapping domains. The RANS mesh must be body fitted to represent smooth solid boundaries. The acoustic mesh can be regular Cartesian, and the CAA domain can be larger — extending to the mid field if Kirchhoff's method is used (Lyrantzis, 1993) or to the far field if high-order optimised numerical schemes are employed (Tam and Webb, 1993). Uniform mean flow has to be assumed outside the region of the CFD simulation.

Prior to the introduction into the acoustic simulation the flow quantities ( $\bar{v}_j, \bar{\rho}$  and  $\bar{p}$ , in air  $c^2 = 1.4 \bar{p}/\bar{\rho}$ ), which multiply the terms of the linearised Euler equations (2.11) and (2.12), are interpolated with piecewise constant functions (choosing the nearest neighbouring point from the irregular CFD mesh). This can be done because typically the flow mesh is finer than the acoustic mesh (see 'Direct simulation of sound' in Chapter 1).

In the following two sections the above algorithm is applied to the sound generation due to vortex-blade interactions using two different CFD codes.

## 5.2 Coupling with the Structured Finite Volume Code PHOENICS

One of the first CFD codes to find industrial application was the PHOENICS package (PHOENICS, 1995). Its name is an acronym of 'Parabolic, Hyperbolic Or Elliptic Numerical Integration Code Series'. Its finite volume algorithms are based on staggered *structured* meshes which can be body-fitted (abutting curved solid boundaries). The implementation includes an input language for specifying the geometry



and the parameters of the solution procedures for a given problem.

In aerodynamic sound problems vortex structures of relatively small size have to be resolved since they are responsible for the generation of much longer wavelengths of sound. In order to avoid excessive refinement of the mesh (and higher computational cost), as well as to reduce the false numerical diffusion of the vortices, *second* order numerical schemes are considered rather than the default first order discretisation. Higher order schemes can be selected by specific input commands with the latest versions of PHOENICS.

Before applying to more realistic problems the *coupling* algorithm described in the previous section is tested with the same 1D problem shown in Figure 1.1. The pressure, velocity and density fields provided by PHOENICS at each time step are input into a finite volume acoustic solver with the simplest numerical scheme (3.4) of those presented in Chapter 3. (In one dimension, this simple scheme can be made very accurate by specifying a Courant number  $\sigma_x$  closer to the limit of 1.)

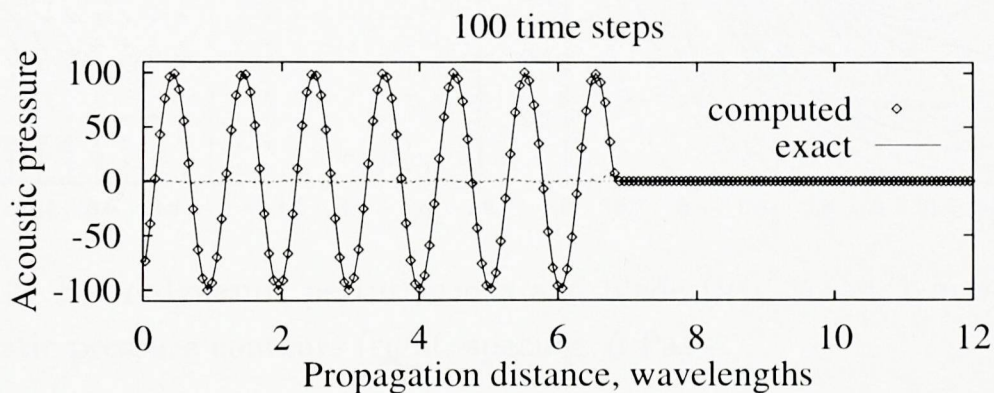


Figure 5.2: Combined solution of test problem

The discretised source term was calculated according to (5.3) with  $S_{vib} = 0$  only for the leftmost cell of the computational domain. The actual source of sound was defined as a time dependent boundary patch in PHOENICS that represents an oscillating piston. (In real simulations such vibrating solid sources should be introduced directly into the acoustic solver rather than passing them through the CFD code; here this is done only for the purpose of testing.)

It can be seen in Figure 5.2 that the actual acoustic signal was recovered perfectly well.

As a **2D example**, generation of sound by vortices impinging on a flat plate is considered. PHOENICS is used to compute the airflow on a mesh that is two

times finer in the direction perpendicular to the plate than the corresponding grid for adequate acoustic simulation.

The middle section of the inlet patch is defined as time dependent sinusoidal source of momentum in the vertical direction. The background flow is uniform at 160 m/s. This creates a vortex that hits the flat plate situated in the middle of the computational domain. Ordinary zero pressure outflow boundary conditions are specified. The top and bottom boundary pressures are also set to zero in order to simulate the vortex convection in open space.

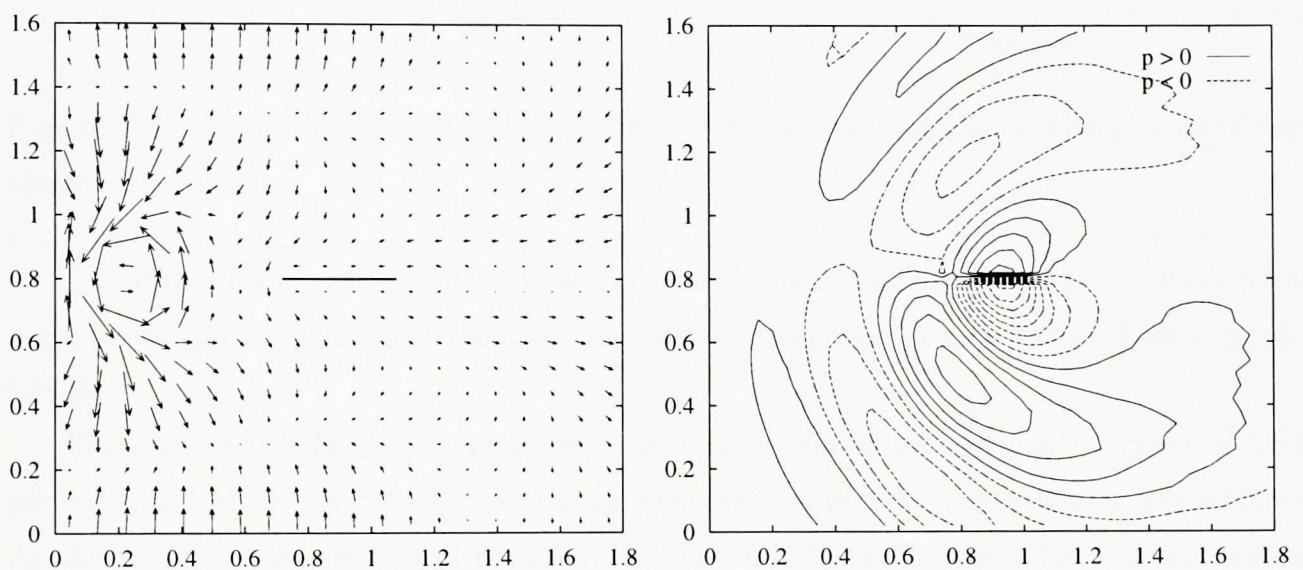


Figure 5.3: Hydrodynamic perturbations and blade (left, scale: 1 m/s to 0.1 m), and acoustic pressure contours (right, spacing: 6 Pa)

The geometry of the problem and the hydrodynamic perturbation velocity field (with the uniform background flow subtracted) are on the left of Figure 5.3.

The pressure fluctuations (temporal derivatives) next to the solid surface are imposed as source terms on the linearised Euler equations which are solved separately as described in Chapter 3. The size of the computational domain is small enough so that the finite volume solver (Djambazov et al., 1997b) can predict accurately the sound field.

A snapshot of the pressure perturbations can be seen on the right of Figure 5.3. A graph was made of the acoustic pressure as a function of time at different locations above and below the solid blade. (The numbers of the cells monitored above and below the centre of the blade are printed next to each line key.) As it can be seen in Figure 5.4, the amplitude of the sound waves generated at the blade decreases away from it as expected.

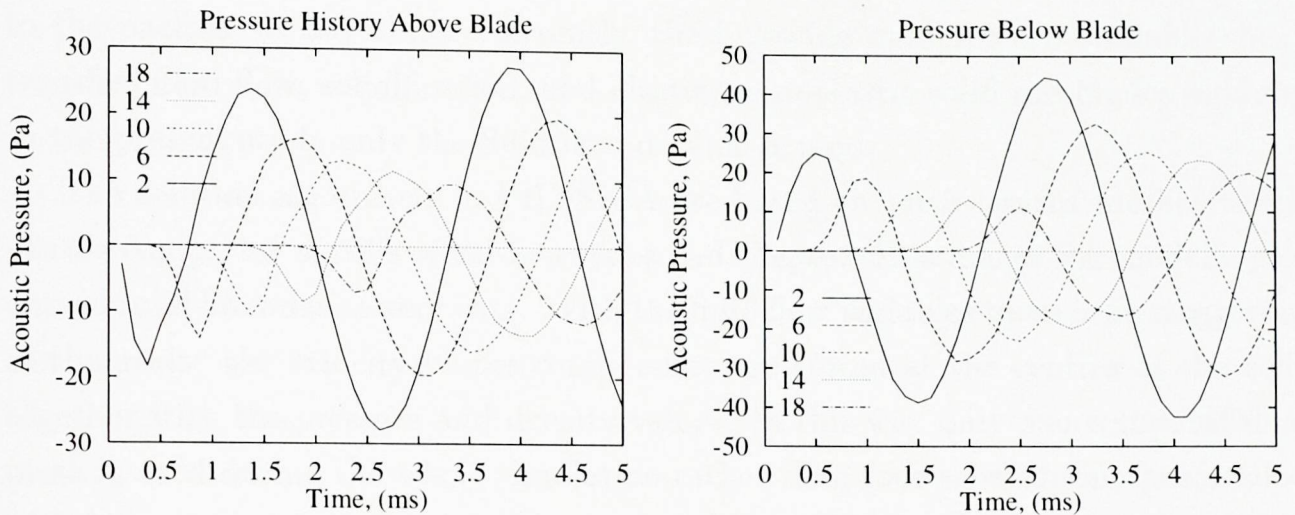


Figure 5.4: Acoustic signal in the specified cells above and below the centre of the blade.

These results together with the sound source formulation in the previous section were presented at the 10th Domain Decomposition Conference (Djambazov et al., 1998b).

The coupled CFD-CAA technique developed so far allows general-purpose RANS solvers like PHOENICS to be used with problems of aerodynamic sound generation. At the locations where sound is generated, the CFD pressure oscillations are passed to a linearised Euler solver that can simulate accurately the acoustic waves propagating away from their source. Also, the local values of the flow quantities which appear in the acoustic equations are provided by the CFD solution.

### 5.3 Coupling with the Unstructured Mesh Finite Volume Code PHYSICA

The coupling between a CFD software package and an acoustic software module will be used in this section for the simulation of aerodynamic sound generated by vortex-blade interactions in *realistic* geometries. PHYSICA (Croft et al., 1995) was selected for its flexibility with complex shapes and different numerical algorithms, and also, because it is being developed at the University of Greenwich.

The PHYSICA package has a flexible modular structure which allows various modelling procedures of various physical phenomena to be accessed in a single numerical simulation. New modules or new features of existing modules can be added

to the package at any time. Currently, the following modules are available: heat transfer, fluid flow, solidification, and elastic/visco-plastic solid mechanics module. In the present study only the fluid flow module is used.

The solution algorithms in PHYSICA are based on *unstructured* meshes which can be comprised of cells of various types and shapes. This makes the modelling of curved solid boundaries very easy. With the fluid flow variables there is no staggering of the grids: the velocity vector components are stored at the centres of the cells together with the pressure and density values. In this way only one computational mesh is used during the whole simulation rather than four separate meshes needed with the staggered approach. (The trade-off is a little decrease in the stability in some cases and a more complex interpolation procedure at cell faces.)

### 5.3.1 Second order schemes

In most CFD implementations the discretisation algorithms are based on the stable upwind scheme (Versteeg and Malalasekera, 1995). It ensures that during the iterative solution, an increase of a quantity at a given location will *always* be followed by an increase and not by a decrease at the neighbouring points. Unfortunately, this restricts the approximation of the variables to piecewise constant in both space and time.

For the *flow perturbations* which generate aerodynamic sound better accuracy is needed. This can be illustrated by comparing the solutions of a vortex pair carried by a uniform mean flow obtained separately with the upwind scheme and with a second order scheme.

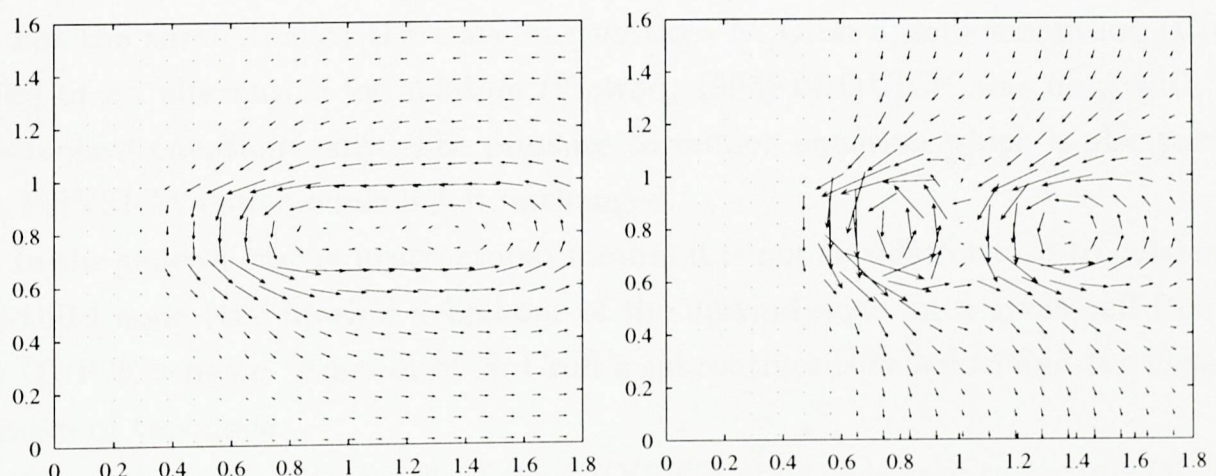


Figure 5.5: Vortex convection with upwind (left) and QUICK (right) schemes

The two results in Figure 5.5 were computed by PHYSICA before (left) and after (right) the implementation of the QUICK scheme which will be examined in greater detail in this section. Very similar results were produced also by PHOENICS (a CFD code described in the previous section). The vortex series in both cases is generated by sinusoidal time dependent boundary conditions at the left boundary of the domain. The mean velocity of 160 m/s (from left to right) is subtracted from the solutions before making the graphs. It can be seen that at a distance of only about three vortex diameters downstream, the upwind solution is not correct: false (numerical) diffusion is observed which is much stronger than with the second order scheme.

QUICK stands for Quadratic Upwind Interpolation for Convection Kinematics. With this numerical scheme the cell face values of fluxes are calculated by second order interpolation between the two neighbouring nodes and an upstream node (Leonard, 1979). The scheme can be formulated in a standard way and in several alternative ways (Versteeg and Malalasekera, 1995).

The standard (straightforward) formulation can be unstable due to the appearance of negative coefficients in the discretised equations. Then during the solution iterations, an increase of a given value will lead to a decrease of a neighbouring value, and false (non-physical) oscillations will be initiated.

The alternative formulations involve placing troublesome negative coefficients in the source term (right-hand side) of the discretised equations. Since the source terms are updated less frequently than the nodal values, the solution procedure is likely to converge before an instability starts developing. All variations should give the same solution upon convergence.

For the simulation of the travelling vortices N. Croft's implementation (Croft, 1998) of an alternative formulation (Flow3d, 1995) of QUICK was used with the *momentum* equations only. The pressure correction equation which is also part of the PHYSICA flow module is left unchanged.

In the general case of unstructured meshes it is not straightforward to determine the third node (the upwind neighbour of the upwind node for a given cell face) of the QUICK scheme. Another of N. Croft's subroutines is called to find the element number of this node.

The changes to version 2.00 of the PHYSICA code necessary to include QUICK (and two more higher order schemes) are the following:

1. In the General Equation Module the call to ‘differencing schemes’ is modified to pass the array pointers along with the other arguments.
2. In the Fluid Flow Module the slot SCHEME UPWIND in subroutine ‘differencing schemes’ is used to call a new subroutine ‘QUICKS\_LINK’. This slot is activated by a special command in the PHYSICA input file.
3. According to parameters initialised in the User Module, the new subroutine ‘QUICKS\_LINK’ can activate one of the second order schemes after a specified number of iteration sweeps. The various first order schemes are also retained as options. For each face the subroutine performs finding of the neighbouring element numbers, a call to find the third upwind element, and calls to calculate the contributions to the system matrix and to the source terms vector.

This implementation was tested with the inviscid example of travelling vortices shown on the right of Figure 5.5.

### 5.3.2 Aerofoil geometry specification

The PHYSICA package is provided with an interface to a finite element grid generation package (FEMGEN, 1992) for the specification of the geometry and for the construction of the unstructured computational mesh. This is very important and convenient for the creation of triangular and tetrahedral meshes.

For the external 2D flow around an aerofoil another approach was chosen which allows automatic meshing of different aerofoils depending on a set of numerical parameters.

The aerofoil is defined by a central curve and a thickness function. The point coordinates on the central curve are calculated using the parametric formula of third order Bezier curves:

$$\begin{aligned}
 x &= x_0v^3 + 3x_1uv^2 + 3x_2u^2v + x_3u^3, \\
 y &= y_0v^3 + 3y_1uv^2 + 3y_2u^2v + y_3u^3, \\
 0 &\leq u \leq 1, \quad v = 1 - u
 \end{aligned}
 \tag{5.5}$$

with the values of the curve parameter  $u$  chosen from the values of a cosine function so that near the leading edge ( $u = 0$ ) and near the trailing edge ( $u = 1$ ) the density of points is the highest. The four governing points of the Bezier curve ( $x_i, y_i$ ,

$i = 0 \dots 3$ ) are defined in a coordinate system with an origin at the aerofoil leading edge in the following way:

$$\begin{aligned} x_0 &= 0 \quad , \quad y_0 = 0 \\ x_1 &= k_{lead} \frac{x_3}{2} \quad , \quad y_1 = 0 \\ x_2 &= x_3 - k_{trail} \frac{x_3}{2} \cos \alpha_{curv} \quad , \quad y_2 = y_3 + k_{trail} \frac{x_3}{2} \sin \alpha_{curv} \\ x_3 &= L \cos(\alpha_{curv}/2) \quad , \quad y_3 = -L \sin(\alpha_{curv}/2). \end{aligned} \quad (5.6)$$

There are four parameters left to specify the shape of the central curve:  $L$  is the chord length,  $\alpha_{curv}$  is the curvature angle (like with a circular arc), and the coefficients  $k_{lead}$  and  $k_{trail}$  with values between 0 and 1 determine the rate of deviation from the tangent at the leading and trailing edge respectively.

The thickness function of the aerofoil is defined in the following way:

$$\begin{aligned} \delta(s) &= \exp(p_T \ln \frac{t}{m}) \sqrt{(1-t)/m_1} \\ t &= \bar{s} + \frac{m - s_M}{\sin \pi s_M} \sin \pi \bar{s} \\ \bar{s} &= \frac{s_T - s}{s_T} \\ m &= \frac{2p_T}{2p_T + 1} \quad , \quad m_1 = 1 - m. \end{aligned} \quad (5.7)$$

Here  $s$  is the distance measured along the central curve of the aerofoil from the leading edge towards the trailing edge, and  $s_T$  is its final value. Only two parameters are enough to define the thickness function  $\delta(s)$ : the exponent  $p_T \leq 1$  determines the sharpness of the trailing edge, and the non-dimensional  $s_M$  determines the position of the maximum thickness.

The points on both sides of the aerofoil are calculated by symmetrically distributing the thickness function along the central curve  $(x_C, y_C)$ :

$$x(s) = x_C(s) \mp \frac{\delta_{max}}{2} \delta(s) \sin \beta \quad , \quad y(s) = y_C(s) \pm \frac{\delta_{max}}{2} \delta(s) \cos \beta$$

with  $\beta$  being the local tangent angle of the central curve.

In total 7 parameters are enough to approximate virtually any aerofoil: 4 for the central curve, 2 for the thickness function, and  $\delta_{max}$  for the maximum thickness.

The mesh around the aerofoil is constructed in stages using again third order Bezier curves (5.5). At first the three aerofoil lines (central line and two sides) are

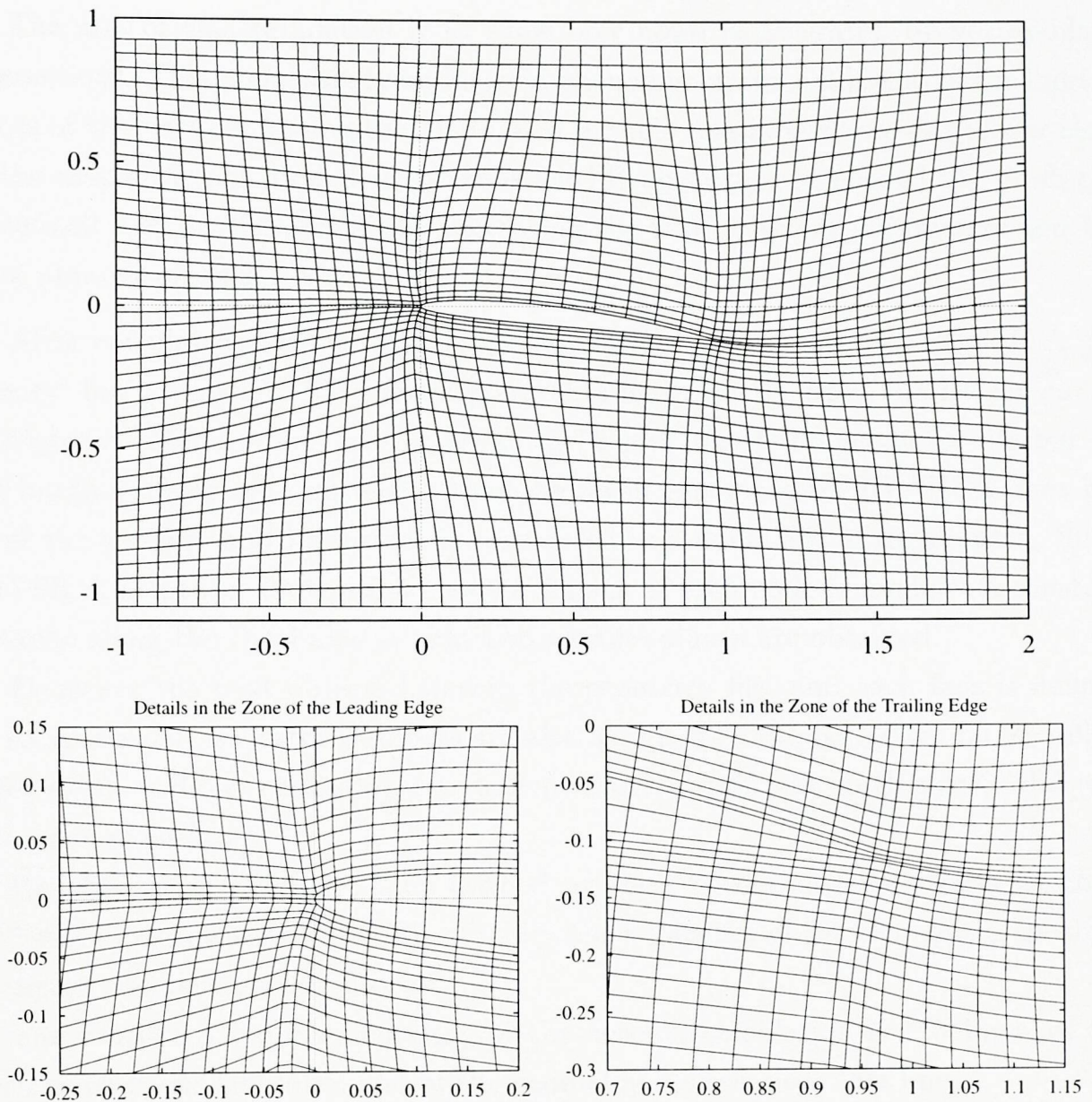


Figure 5.6: Aerofoil and mesh for inviscid flow computation

extended smoothly to the west and to the east side of the rectangular computational domain. Then the cross sections of the aerofoil are extended smoothly to the south and to the north side of the domain. Finally cross curves are created retaining regular spacing at all sides of the domain. The resulting mesh is shown in Figure 5.6 with the regions of the leading and trailing edges magnified.

Nine more parameters are necessary to define the whole mesh: one for the angle of attack, two for the total number of cells along each axis, two for the total length along the axes, two for the positions of the south and west sides of the domain relative to the origin at the leading edge of the aerofoil, and two exponent parameters specifying the density of the mesh close to the aerofoil in both directions.



The aim of this simulation is to show how noise is generated by vortex-blade interactions. This is essentially an inviscid phenomenon since it is due to the inertial forces of the vortices hitting the blade, and for this reason the boundary layer close to the aerofoil is not modelled. If the exact lift and drag are needed the mesh can be refined next to the aerofoil by increasing the two exponent parameters and the total number of cells.

After calculating the coordinates of the vertices of the mesh the PHYSICA 'geometry' file is created. The vertices are obtained directly from the Bezier curves as described above. They are listed at the beginning of the geometry file but all the internal nodes are omitted so that the mesh generated in PHYSICA does not cover the aerofoil, and its surface is considered as a boundary of the domain. Since PHYSICA is a three dimensional code a replica of each node is made at a constant distance along the third axis so that two parallel planes are obtained.

Faces are the next objects listed in the geometry file, and each face is defined by its four vertices. Patch numbers are also assigned to the boundary faces: inflow (west side), outflow (east side), constant pressure (south and north side), solid wall (the aerofoil surface).

Hexahedral elements are then defined by their bordering faces whose numbers are listed next in the geometry file. Finally, the adjacency of the elements is specified by means special element records.

The FORTRAN implementation of the aerofoil calculation is 115 lines, of the mesh generation: 300 lines, and of the geometry file creation: 200 lines.

### 5.3.3 Simulation results

According to the algorithm outlined in the first section of this chapter, first a **steady** solution of the the airflow around the aerofoil was obtained. The variable density was defined in the user module of PHYSICA for the isentropic conditions (1.3). The free-stream velocity is 160 m/s which in air produces a Reynolds number of about  $8.6 \times 10^6$  based on the aerofoil chord length of 1 m. No turbulence model was used, and inviscid flow was assumed instead because the purpose of this simulation is to study the flow perturbations rather than the mean flow. The aerofoil chord forms an angle of attack  $7^\circ$  with the free-stream velocity vector.

Upwind differencing scheme was used to compute the steady mean flow (Figure 5.7) since the QUICK scheme becomes unstable with so many iterations (about

200) that are necessary for convergence.

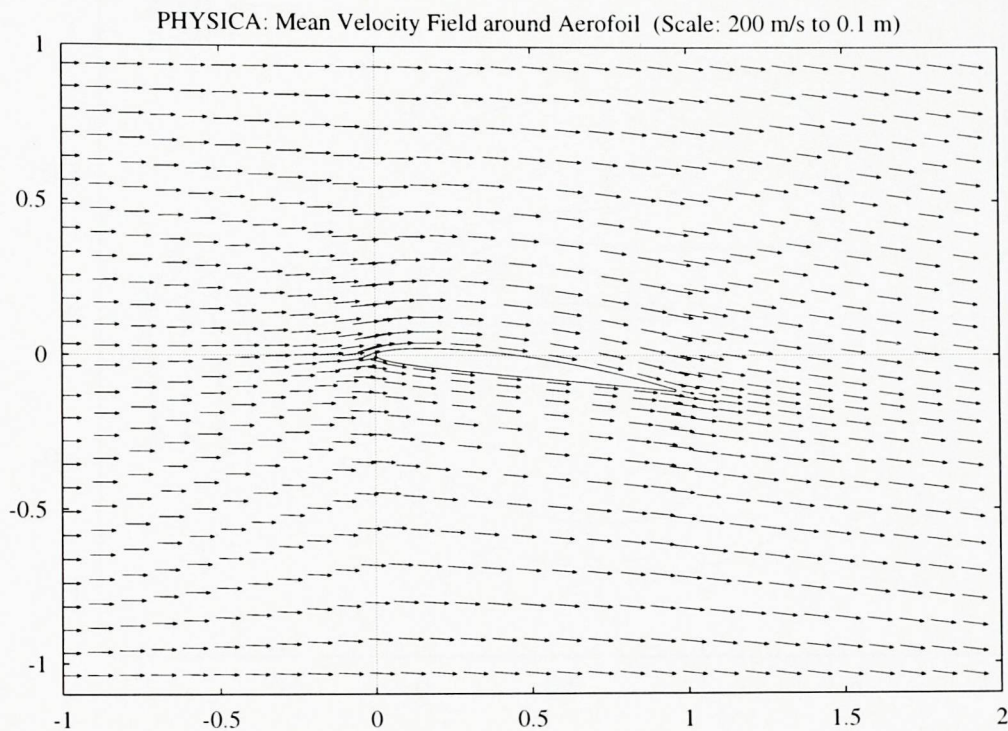


Figure 5.7: Steady mean flow solution

At the second stage of the aeroacoustic computation, the **time dependent** simulation of the flow is initialised with the steady solution, and with the inflow boundary conditions at the left end of the domain, a time dependent perturbation of the mean flow is specified in the perpendicular direction. It is sinusoidal with amplitude 7.5% of the mean velocity and is applied to the inflow momentum in the middle part of the inflow boundary. In this way a series of vortex perturbations of the mean flow is generated.

In a real aeroacoustic computation the flow perturbations (vortex structures) should not be prescribed as it is done here. Instead they should be resolved within the CFD code. The turbulence models used in RANS solvers are generally tuned for steady flow and are likely to be unsuitable for this purpose, especially at the high Reynolds numbers that are characteristic of aerodynamic noise problems. The most promising CFD technique, in the author's opinion at the time of writing, is Large Eddy Simulation (LES). Hopefully, PHYSICA will be equipped with LES capabilities in future.

The implementation of the QUICK scheme described in the beginning of this section is used with the flow perturbation solution. Before switching on the time dependent boundary patch the solution is left to settle with the second order scheme

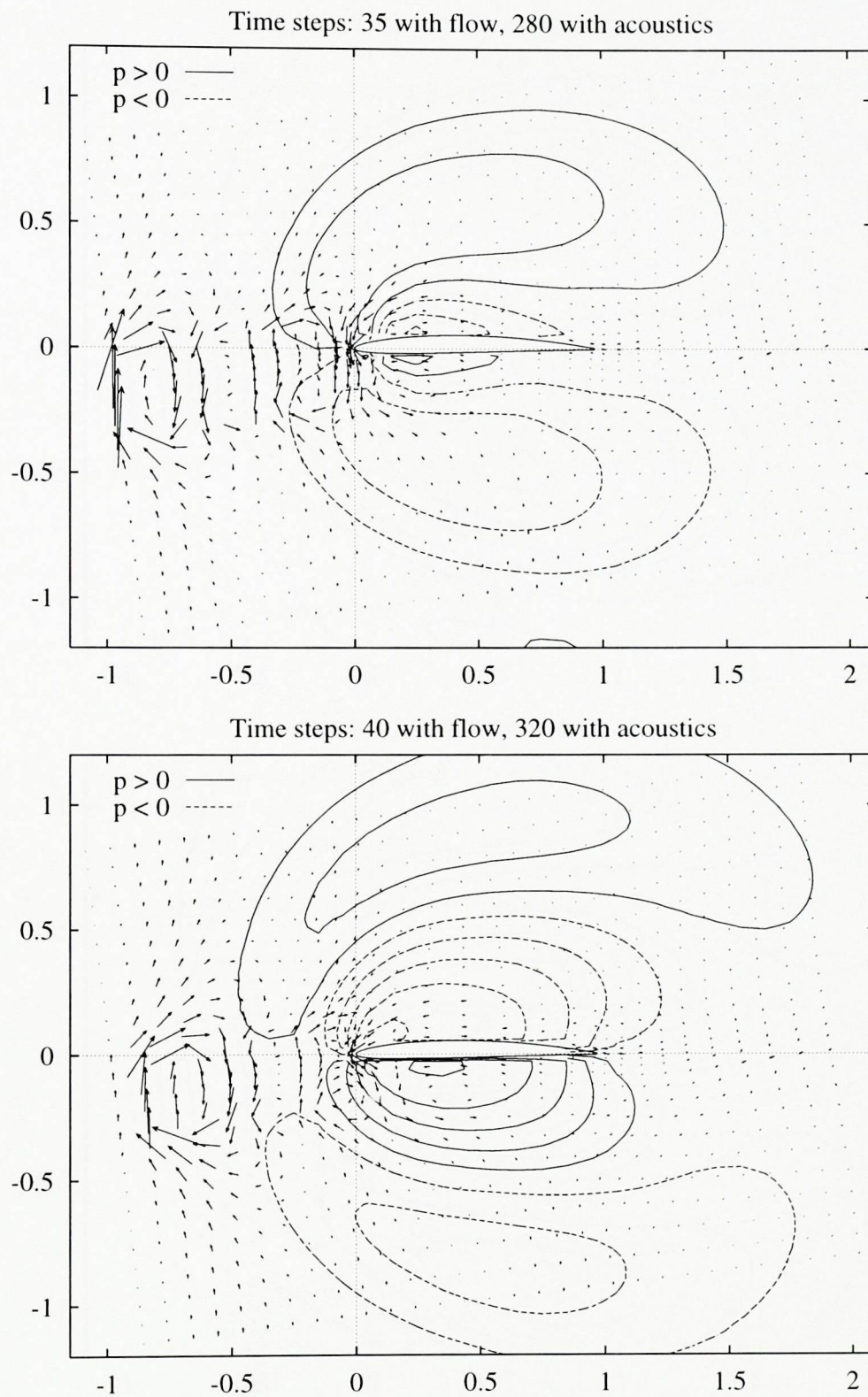


Figure 5.8: Flow and acoustic perturbation fields (showing superposition of mean-flow and acoustic domains)

for 20 time steps. When the resulting velocity field is compared to the upwind solution, only minor differences are observed in the wake region behind the aerofoil.

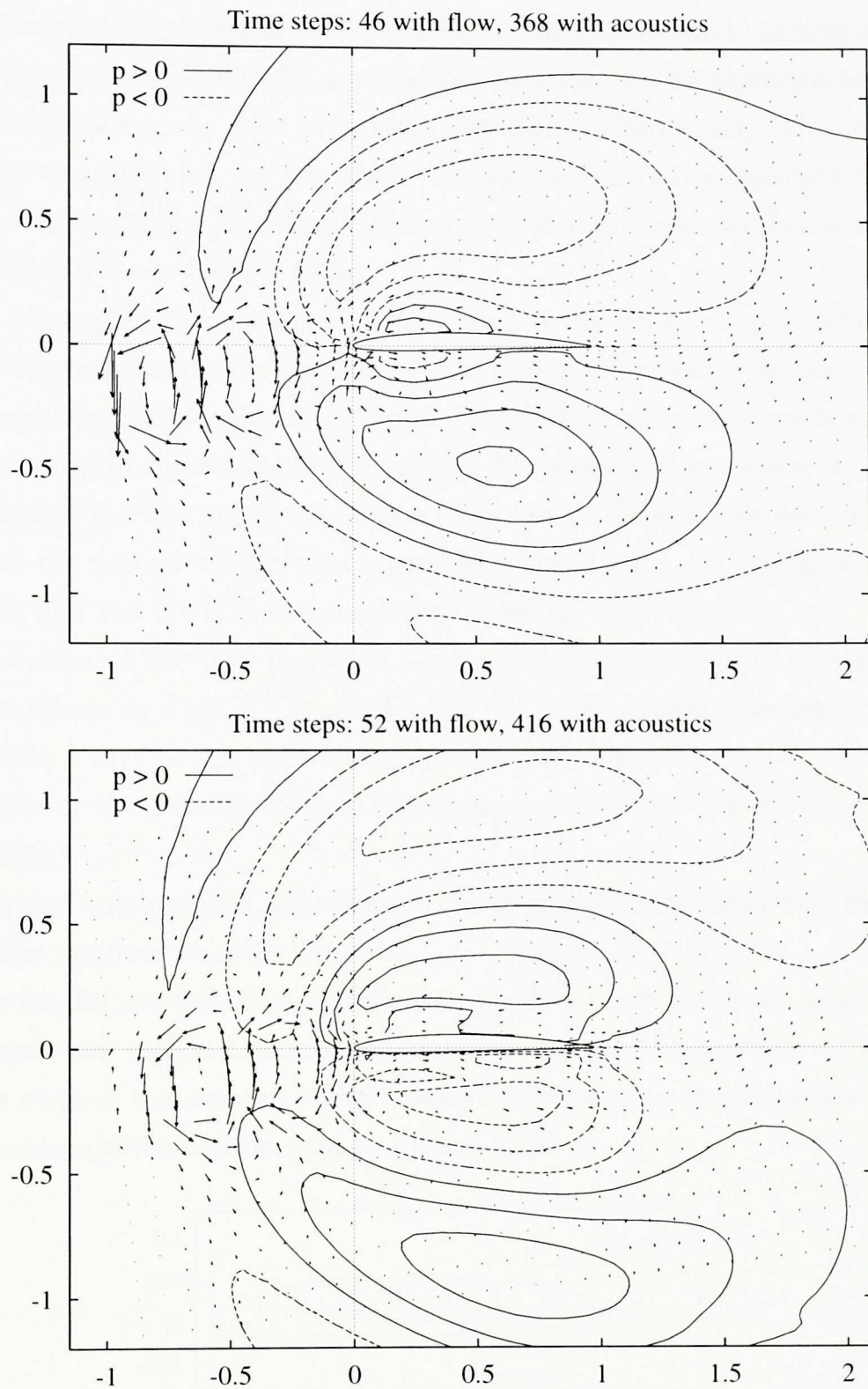


Figure 5.9: Flow and acoustic perturbation fields, continued (showing superposition of flow and acoustic perturbations)

An instability develops in PHYSICA with variable density and with the small time steps needed to trace the passing vortices. This happens even with the robust

upwind scheme for the momentum equations. With unreasonably strong relaxation of the density variations (0.05 %) a converged solution can be obtained but it takes too many iterations to be part of a time dependent simulation. The most likely reason is the way in which the transient term of the continuity equation is handled in the pressure correction equation. Alternative formulations are being considered by the PHYSICA team (Croft, 1998).

A semi-incompressible assumption was used to by-pass this problem. For the purpose of this test simulation it was assumed that the flow perturbations (vortices) do not change the local density of the fluid. Physically, this is not true because there is a pressure drop in the centre of the vortex. The density fluctuation in the vortex is neglected, but the the density field due to the mean flow is retained. Technically speaking, at the restart of the time dependent simulation, the variable density is switched off, and the QUICK scheme is switched on.

The evolution of the flow field due to the vortex convection is illustrated by a series of snapshots in Figure 5.8 and Figure 5.9 with arrows representing velocity vectors (scale: 4 m/s to 0.1 m). These vectors depict only the perturbation of the mean flow due to the passing vortices (the mean velocity vector has been subtracted before plotting).

At the third stage of this aerodynamic noise problem the **acoustic** module described in the previous chapter is used.

In order for the aerofoil to be better discretised on the rectangular acoustic mesh, the CFD mesh and velocity vectors have been rotated to the angle of attack around the leading edge of the aerofoil. The blocked cells forming the solid boundaries in the acoustic simulation can be seen in Figure 5.10.

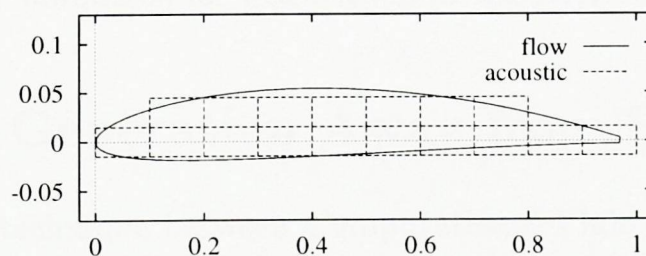


Figure 5.10: Representation of solid boundaries

Although the flow solution does not contain acoustic waves, the sources of sound can be calculated from the pressure variations on the surface of the aerofoil. A finite volume formulation is used for the source term that is compatible with the finite

volume formulation within the CFD code. The pressure fluctuations (temporal derivatives) of the flow solution next to the solid surface are averaged over the rectangular cells which are neighbouring the blocked cells (see Figure 5.10), and are imposed as the source term  $S$  of the linearised Euler equations (2.11) using the simplest formulation (5.2) of those described in the first section of this chapter. The aerofoil surface is assumed stationary ( $S_{vib} = 0$ ).

Since the linearised Euler solver uses explicit schemes, several acoustic time steps are needed to cover one flow time step.

The flow quantities  $\bar{v}_j, \bar{\rho}$  and  $\bar{p}$  in (2.11) and (2.12) are interpolated with piecewise constant functions (choosing the nearest neighbouring point from the irregular mesh). The acoustic mesh is much coarser than the CFD mesh because the acoustic wavelength is longer than the vortex diameter.

The acoustic domain is made larger than the flow domain as this will be the case in most real aeroacoustic computations (see Chapter 2). Uniform flow is assumed outside the flow computational domain.

In order to compare the mean convection and the sound propagation times, each of the instantaneous plots (Figures 5.8 and 5.9) shows the flow perturbation field and, superimposed on it, the resulting acoustic waves propagating away from the aerofoil. It can be seen that the mechanism of inviscid sound generation by perturbations in the flow has been captured by the combined simulation. The present implementation of the coupling technique does not account for sound generated in the wake downstream of the aerofoil.

The results obtained by this combined simulation using PHYSICA and the linearised Euler solver (Chapter 4) were presented in a paper (Djambazov et al., 1998c) which has also been submitted for publication in the *AIAA Journal*.

## 5.4 Sound Generation Away from Solid Bodies

With the coupling technique between Computational Fluid Dynamics (CFD) and acoustics, described in the previous sections, the finite volume cells which become sources of sound have to be specified prior to the simulation. Inside the flow, the source regions are likely to move and cannot be localised before the simulation. This is especially true for jets, wakes and other similar types of airflow.

The general case of aerodynamic sound generation will be considered below.

Ways will be suggested of extending the coupling techniques between CFD and acoustic codes to include simulations of the sound produced inside the fluid volume.

Several approaches can be followed in this more general case. **First**, a fixed mesh of source cells can be defined with a regular spacing (say, every tenth cell) which has to be determined experimentally. Considering every computational cell in the CFD domain as a source cell is not correct, because the propagation of acoustic waves close to the source is partly resolved by the CFD code (see Figure 1.1), and false signals will enter the acoustic simulation due to CFD pressure oscillations that have already been input into the acoustics. It is possible to set up numerical experiments for the tuning of this spacing parameter for each couple of CFD and acoustic codes. Analytical solutions of source distributions in three dimensions have to be used as a basis.

The **second** approach is to try to detect the cells that are currently sources of sound during the simulation. The acoustic motion is always characterised by non-zero divergence (see 2.11) of the velocity vector (the outflow of mass is slightly different from the inflow, and this difference varies in a periodic manner). If *local* 4-dimensional averaging (see 2.1) with suitable bounds is applied currently to every cell in the computational CFD domain, the mean velocity vector can be determined and subtracted to reveal the perturbation fluxes across the cell faces. Then, if all *perturbation* fluxes for a given cell are of the same sign (all inflow, or all outflow), the cell can be assumed to be a source cell, otherwise, there is only propagation across this cell. Unlike the first approach, this one introduces runtime overhead which slows down the simulation. There is also some empirical knowledge involved about the width of the averaging region around each cell.

The **third** approach is based on a different interpretation of the variables that are solved for within the acoustic module. Rather than solving for the full acoustic field, it is possible to use the acoustic (linearised Euler) module to solve for the *differences* between the true physical fields of pressure, velocity and density and the ones computed by the CFD code. In this way all the information about the sound sources contained in the CFD solution will be used automatically, and no special assumptions are needed. The acoustic signal of interest has to be extracted by post-processing the *sum* of the CFD and linearised Euler solutions which will involve averaging on scales larger than the longest wavelength studied.

A simple example with this approach follows in the next section.

## 5.5 Acoustic Expansion Technique

The complex pressure, density and velocity fields that are observed in aerodynamic noise problems can be considered as comprised of three types of physical fields as shown in Figure 5.11. The magnitude of the quantities characterising each of these

Physical field		
Mean flow	Flow perturbations	Aerodynamic sources of sound
Computational Fluid Dynamics (CFD)		Acoustic solvers

Figure 5.11: Analysis of unsteady airflow

fields decreases from left to right, but the smallest length scales are associated with the flow perturbations (vortices, eddies) and not with the acoustic waves.

As shown in Figure 5.11, the CFD solution contains part of the acoustic waves. This suggests that it can be ‘expanded’ to the full physical field with the help of a linearised Euler (acoustic) code which will solve for the missing part.

In Chapter 2, a perturbation analysis of the Navier-Stokes equations (2.2) and (2.3) was made assuming that the averaged quantities satisfy *exactly* their equations of continuity (2.4) and momentum (2.5). Here it will be taken into account that the CFD solution satisfies the *discrete* analogues of these equations rather than the differential equations themselves. The difference is very small with a converged CFD solution but it matters for the acoustic waves that are even smaller in magnitude.

Equations (2.2) and (2.3) can be written with the following notation:  $\bar{\rho}$ ,  $\bar{v}_j$  ( $j = 1 \dots 3$ ),  $\bar{p}$  define the compressible time dependent CFD solution, and  $(\rho, v_j, p)$  is the missing wave propagation part of the physical fields that have to be determined.

The following is assumed:

1. The missing perturbations  $(\rho, v_j, p)$  can be resolved on a regular Cartesian grid with stepwise solid boundaries. (This means that they should *not* contain flow perturbations which require smooth solid boundaries.)
2. The time dependent forces  $(\bar{f}_i, i = 1 \dots 3)$  which include viscous and external forces (2.3) can be obtained from the CFD code together with  $(\bar{\rho}, \bar{v}_j, \bar{p})$ . (This should not be difficult with an ‘open’ computational environment like PHYSICA (Croft et al., 1995) but may not be available in a commercial CFD package.)



3. The viscous forces in the missing perturbations that are part of the  $f_i$  term in (2.3) can be neglected (this is easy to accept bearing in mind that ‘missing’ is only part of the acoustic waves which are not affected by viscosity because of the small particle velocity), and the external forces in  $f_i$  are known.

As this was done in Chapter 2, the Navier-Stokes equations (2.2) and (2.3) are rearranged so that only the terms containing derivatives of the perturbations remain on the left-hand side. The CFD equations for  $\bar{\rho}$  and  $\bar{v}_j$ , however, are not subtracted and appear on the right:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + (\bar{v}_j + v_j) \frac{\partial \rho}{\partial x_j} + (\bar{\rho} + \rho) \frac{\partial v_j}{\partial x_j} &= Q_{ext} + \bar{R}'_C \\ \frac{\partial v_i}{\partial t} + (\bar{v}_j + v_j) \frac{\partial v_i}{\partial x_j} + \frac{1}{\bar{\rho} + \rho} \frac{\partial p}{\partial x_i} &= \frac{f_{i,ext}}{\bar{\rho} + \rho} + \bar{R}'_M \\ \frac{\partial p}{\partial \rho} = c^2 = \gamma \frac{\bar{p} + p}{\bar{\rho} + \rho} &, \quad \text{in air } \gamma = 1.4 \end{aligned} \quad (5.8)$$

where the CFD residuals are

$$\bar{R}'_C = - \left[ \frac{\partial \bar{\rho}}{\partial t} + (\bar{v}_j + v_j) \frac{\partial \bar{\rho}}{\partial x_j} + (\bar{\rho} + \rho) \frac{\partial \bar{v}_j}{\partial x_j} \right] \quad (5.9)$$

$$\bar{R}'_M = \frac{\bar{f}_i}{\bar{\rho} + \rho} - \left[ \frac{\partial \bar{v}_i}{\partial t} + (\bar{v}_j + v_j) \frac{\partial \bar{v}_i}{\partial x_j} + \frac{1}{\bar{\rho} + \rho} \frac{\partial \bar{p}}{\partial x_i} \right]. \quad (5.10)$$

Since the perturbations have very small magnitudes compared to the CFD quantities, in most cases it will be possible to assume:

$$\begin{aligned} \bar{\rho} + \rho &\approx \bar{\rho} \\ \bar{v}_j + v_j &\approx \bar{v}_j \\ \bar{p} + p &\approx \bar{p}, \end{aligned} \quad (5.11)$$

and then the CFD residuals are expressed only in terms of the CFD quantities:

$$\bar{R}_C = - \left[ \frac{\partial \bar{\rho}}{\partial t} + \bar{v}_j \frac{\partial \bar{\rho}}{\partial x_j} + \bar{\rho} \frac{\partial \bar{v}_j}{\partial x_j} \right] \quad (5.12)$$

$$\bar{R}_M = \frac{1}{\bar{\rho}} \bar{f}_i - \left[ \frac{\partial \bar{v}_i}{\partial t} + \bar{v}_j \frac{\partial \bar{v}_i}{\partial x_j} + \frac{1}{\bar{\rho}} \frac{\partial \bar{p}}{\partial x_i} \right]. \quad (5.13)$$

The isentropic pressure-density relation (2.10) can also be used because the perturbations considered here are part of the acoustic waves propagating in the fluid

medium. After substitution, the perturbation density  $\rho$  can be excluded from (5.8), and *linearised Euler* equations with source terms are obtained:

$$\frac{\partial p}{\partial t} + \bar{v}_j \frac{\partial p}{\partial x_j} + \bar{\rho} c^2 \frac{\partial v_j}{\partial x_j} = c^2 (Q_{ext} + \bar{R}_C) \quad (5.14)$$

$$\frac{\partial v_i}{\partial t} + \bar{v}_j \frac{\partial v_i}{\partial x_j} + \frac{1}{\bar{\rho}} \frac{\partial p}{\partial x_i} = \frac{1}{\bar{\rho}} f_{i,ext} + \bar{R}_M \quad (5.15)$$

where  $Q_{ext}$  and  $f_{i,ext}$  denote acoustic external sources of mass and momentum respectively which are most naturally introduced in the perturbation part of the simulation rather than in the CFD part. It can be seen that the small residual of the CFD continuity equation is multiplied by the large factor  $c^2$  and, therefore, cannot be ignored.

In the CFD fully-implicit discretisation the residuals (5.12) and (5.13) are nearly zero at convergence but their magnitude is enough to damp out the acoustic waves as it was shown in Figure 1.1. Assuming that this is due mainly to the piecewise constant approximation along the temporal axis, it is suggested here that these terms are *discretised again* (after the CFD solution) using higher order of approximation. Since the CFD mesh may be unstructured, it is unlikely that approximations of order higher than two will be preferred. For example, the scheme (3.14) that is second order in space and first order in time can be very useful for evaluating the fluxes at the faces of the finite volume cells on a cell-centred mesh.

These CFD cells are typically smaller than the Cartesian blocks of the acoustic mesh which overlaps the CFD mesh. Then, the evaluated residuals will be averaged within each Cartesian cell prior to their introduction into the source terms of the linearised Euler solver described in Chapter 4.

The acoustic expansion algorithm can be summarised in the following steps:

1. Solve the time dependent RANS equations (2.4) and (2.5) using a CFD code to obtain  $\bar{\rho}$ ,  $\bar{v}_i$ , and  $\bar{p}$ .
2. Evaluate higher order approximations of the CFD residuals (5.12) and (5.13) on the CFD mesh at each time step.
3. Map the calculated residuals onto the regular acoustic mesh and add them to the corresponding external sound sources.
4. Solve the linearised Euler equations (5.14) and (5.15) using an acoustic code of high accuracy to obtain the missing perturbations  $p$  and  $v_i$ .

5. Post-process (if necessary) the full time dependent fields  $(\bar{p} + p)$  and  $(\bar{v}_i + v_i)$  to determine the acoustic parameters of interest (amplitude, phase, direction of propagation, etc.)

The regular Cartesian mesh is essential for the accurate representation of the acoustic waves (as shown in Chapters 3 and 4). Therefore, it is important that the CFD code does not leave any *flow perturbations* unresolved, and that only acoustic motion is left out. In particular, this requirement means that the time dependent CFD code should be compressible so that, for example, pressure drops at vortex centres do not enter the acoustic simulation via the residual sources.

The acoustic expansion is similar to the acoustic/viscous splitting (Hardin and Pope, 1994). The main difference is that the CFD here is required to solve the compressible Navier-Stokes equations rather than relying on the incompressible solvers which were common several years ago. Also, the new fluctuations  $p$  and  $v_i$  do not contain a hydrodynamic component (1.10), and this allows them to be resolved on regular Cartesian meshes.

To test the feasibility of this approach the simple one-dimensional example will be considered of an initial-value wave propagation problem with exact solution

$$\begin{aligned}
 p &= f(x - ct) + f(x + ct) \\
 \bar{\rho}cv_1 &= f(x - ct) - f(x + ct) \\
 f(x) &= \begin{cases} \frac{A}{2} \left(1 + \cos 2\pi \frac{x}{\lambda}\right), & |x| < \frac{\lambda}{2} \\ 0, & |x| \geq \frac{\lambda}{2} \end{cases}
 \end{aligned} \tag{5.16}$$

where  $A$  is the amplitude and  $\lambda$  is the wavelength of the two pressure pulses that start from the origin ( $x = 0$ ) at  $t = 0$ .

The initial conditions  $p = 2f(x)$  and  $u = 0$  were prescribed for the CFD solution which uses a structured finite volume code (PHOENICS, 1995) with QUICK differencing scheme for the momentum equations. The time dependent result pictured in Figure 5.12 is similar to that of Figure 1.1 because the problem is similar and the software is the same. Since the problem is symmetrical with respect to the origin, only the right part ( $x > 0$ ) is shown (and solved for).

Uniform mesh was used to avoid averaging of the residual sources with this test. The time step of the CFD simulation can be several times larger than the time step of the explicit Euler solver (which has to obey the Courant limit). In this example the CFD makes 12.5 time steps per cycle with 20 points per wavelength. In fact, time

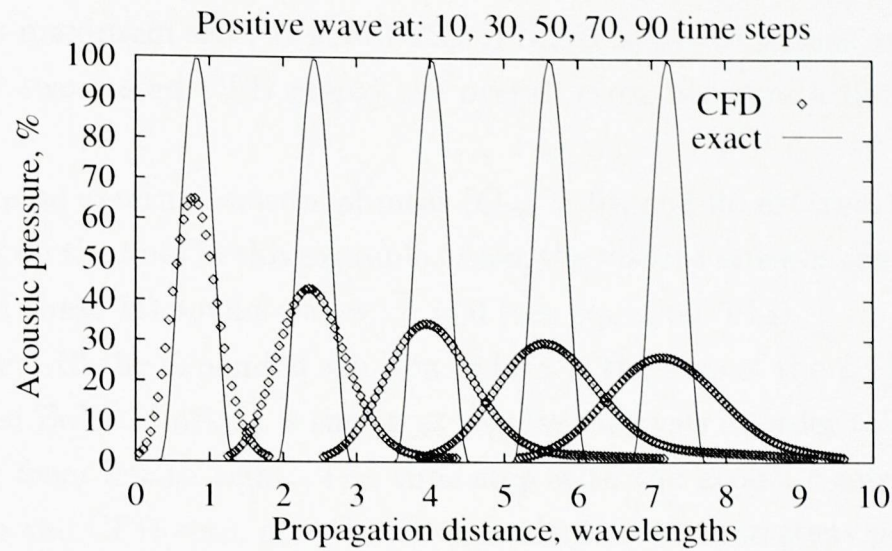


Figure 5.12: Analytic and preliminary CFD solutions of test problem

steps smaller than this produce greater numerical errors over the same propagation distance. This is most probably due to the false diffusion of the CFD schemes which accumulates with every time step.

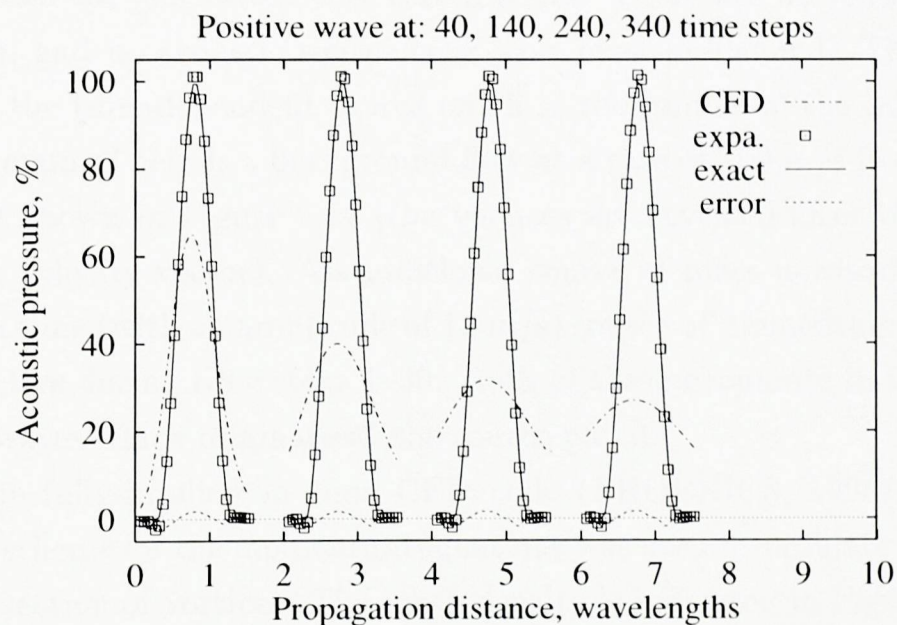


Figure 5.13: Acoustic expansion solution of test problem

The acoustic module starts with zero initial conditions, and gradually accumulates the differences between the real pressure and velocity fields and their CFD representations. This process is driven by the source terms of (5.14) and (5.15) which are discretised in a time-accurate way. The solution in Figure 5.13 is obtained with second order approximation of the CFD quantities along the temporal

axis, and its maximum error is about 2%. If linear approximations are used (which require only two stored CFD steps) the overall error becomes a little higher than 6%.

There are no external sources of mass ( $Q_{ext} = 0$ ), and no external forces ( $f_{i,ext} = 0$ ) are acting on the fluid in this example. Also, the viscous stresses can be completely ignored with these 1D sound waves:  $\overline{f}_i = 0$  (see equation 5.13).

In Figure 5.13 the expanded solution (which is the sum of the CFD solution and the linearised Euler solution) is shown at regular intervals in order to trace the wave propagating from left to right. The time step with the acoustic module is 4 times smaller than the CFD step, and this is equivalent to 50 time steps per cycle. Since the acoustic procedure is fully explicit (see Chapter 4) these expansion steps are computationally very inexpensive (the acoustic module needs less than 10 s to do the expansion of this example including the input and output of disk files). It can be seen that the result of this one-dimensional test is very encouraging.

A two-dimensional example is presented in Figure 5.14 showing the sound waves associated with the generation of a vortex series. There are no solid bodies in the flow domain, and no acoustic source cells have been predefined. The vortices are initiated by the time-dependent source patch in the middle of the left boundary of the CFD domain. There is a background flow at a rate of 160 m/s from left to right which is not shown in Figure 5.14 (the vortices are revealed after the subtraction of the mean velocity vector). An additional source of mass is associated with the sinusoidal in time (with an amplitude of 12 m/s) source of momentum in the vertical direction, active during time steps 1–30. Both of these cooperate in the production of acoustic waves which originate at the source patch.

The same fully-implicit in time CFD code (PHOENICS, 1995) with QUICK differencing scheme for the momentum equations was used to simulate the generation and the convection of vortices. The mesh density is indicated in Figure 5.14 by the density of the arrows representing velocity vectors. As expected, no acoustic waves can be identified in the resulting CFD pressure field. After the expansion steps (re-discretisation, mapping of residuals, and linearised Euler solution), the missing part of the pressure field is obtained, and it is shown in Figure 5.14 by contours. In this case of regular meshes with no solid objects, the mapping procedure is simple: two CFD cells in the vertical direction constitute one acoustic cell, and simple arithmetic averaging is used for the mapping.

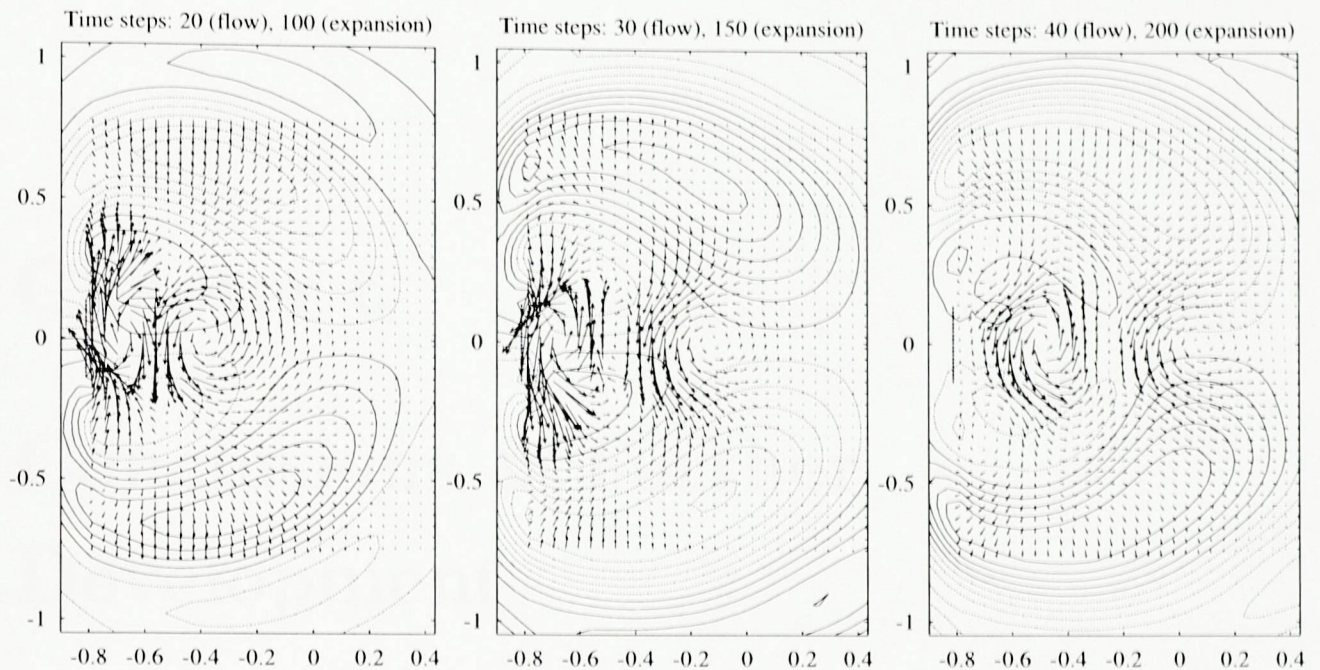


Figure 5.14: Vortex generation and acoustic expansion pressure contours (positive - solid lines, negative - dashed lines, spacing: 20 Pa). Velocity vector scale: 3 m/s to 0.2 m. Vertical dashed line marks vortex generation source patch.

Figure 5.14 shows clearly the acoustic waves that have been produced at the vortex generation patch propagating upwards, downwards, and out of the domain. There is no analytical validation for this example, but the results obtained are physically correct.

In real simulations post-processing of the resulting pressure field is needed in order to determine the noise spectrum and directivity. For this purpose, the acoustic component of the pressure field has to be extracted from the sum of the CFD and the expansion solutions. This can be done by means of a ‘running average’ procedure following the definition in (2.1). The limits of the averaging region have to be chosen large enough to contain the longest wavelengths studied. When the resulting average pressure field is subtracted from the full time-dependent signal the necessary acoustic pressure field is obtained. In many cases only the mid-field or the far-field sound is of interest. Then the CFD code can be tested to determine how far any traces of the acoustic waves reach; if fully-implicit dissipative schemes are used within the CFD code, those waves may not be reaching the mid field, not to mention the far field. Therefore, far from the aerodynamic sources, the linearised Euler acoustic expansion solution will contain the whole sound signal, and no post-processing of the result will be necessary.

## Chapter 6

# Conclusions & Further Development

Various aspects of the aerodynamic noise problem have been considered, and stronger emphasis was given to the direct simulation of sound using Computational Aeroacoustics (CAA) algorithms. The approach with separate computations for the air-flow and for the sound was followed in the ‘near’ acoustic field as described in Chapter 2. After considering a finite volume technique of improved accuracy (Chapter 3), preference was given to an optimised higher order finite difference scheme for the linearised Euler equations governing the sound propagation in moving media. It was implemented in three dimensions together with a set of solid-wall and radiation boundary conditions (Chapter 4). The new code was validated against analytical solutions, and was coupled with two different Computational Fluid Dynamics (CFD) codes for the simulation of the process of aerodynamic noise generation (Chapter 5). Possible extensions for the general aeroacoustic case of non-localised sound sources were also presented.

The contribution of this research to the knowledge of the subject of Computational Aeroacoustics can be summarised as follows:

1. A finite volume technique was developed for the linearised Euler equations that is more accurate than the general CFD schemes and can be used in some aerodynamic sound problems.
2. Staggered-mesh modifications of the optimised dispersion relation preserving (DRP) finite difference schemes for the linearised Euler equations were derived

- using a heuristic approach.
3. A solution procedure for the linearised Euler equations was designed which handles easily complex solid boundaries with high accuracy and reasonable computational cost.
  4. A coupling technique of two CFD codes with the acoustic module was demonstrated to capture the mechanism of sound generation by perturbations of the airflow at solid surfaces.
  5. An acoustic 'expansion' technique for revealing the acoustic waves in general-purpose CFD solutions was tested in one and two dimensions proving to be a promising tool for the case of aerodynamic sound generation away from solid bodies.
  6. Better understanding was achieved of the three-level interaction between airflow and sound: mean flow, flow perturbations, and acoustic waves. This can be a good basis for designing aerodynamic noise suppression methods.

**Future** steps can begin with a three-dimensional implementation of the general coupling of CFD with the linearised Euler solver (the acoustic expansion technique). The source terms of the expansion equations will be calculated on the irregular CFD mesh, and after that, together with the mean-flow quantities, they have to be averaged over each of the overlapping rectangular acoustic computational cells. Careful analytical (where possible) or experimental validation of this technique is necessary when using different RANS solvers.

The next step will be towards numerical simulations of the actual process of viscous vortex generation at high Reynolds numbers. The method which is most likely to succeed in doing this, is Large Eddy Simulation (LES). It will probably need an experimentally tuned sub-grid turbulence model. Alternatively, some time-accurate implementations of RANS solvers can be attempted, again incorporating specially tuned turbulence models.

In a few years time, on the basis of detailed numerical simulation, researchers will be able to propose suitable methods for modifying the flow perturbation pattern in a way that will decrease the aerodynamic noise. Eventually, this will lead to more silent propellers, jets, wind turbines, high-speed trains, and even personal computers and wind-streamed buildings.



# Bibliography

- Agarwal, R. K. and Huh, K. S. (1995). A novel formulation of farfield boundary conditions for computational acoustics. In (Lyrintzis et al., 1995), pages 35–40.
- Atkins, H. (1997). Continued development of the discontinuous Galerkin method for computational aeroacoustics applications. In *3rd AIAA/CEAS Aeroacoustics Conference*, number 97-1581 in AIAA Papers.
- Avital, E., Sandham, N., and Luo, K. (1998). Mach wave radiation by time-developing mixing layers. Part II: Analysis of the source field. *Theoretical and Computational Fluid Dynamics*.
- Baysal, O., Kaushik, D., and Idres, M. (1997). Low-dispersion scheme for nonlinear acoustic waves in nonuniform mean flow. In *3rd AIAA/CEAS Aeroacoustics Conference*, number 97-1582 in AIAA Papers, pages 10–18.
- Colonus, T., Lele, S. K., and Moin, P. (1995b). The sound generated by a two-dimensional shear layer: A comparison of direct computations and acoustic analogies. In *1st CEAS/AIAA Aeroacoustics Conference*, number 95-036 in AIAA Papers.
- Colonus, T., Lele, S. K., and Moin, P. (1995a). The sound generated by a two-dimensional shear layer: the far field directivity from computations and acoustic analogies. In (Lyrintzis et al., 1995), pages 47–52.
- Crighton, D. (1993). Computational aeroacoustics for low Mach number flows. In (Hardin and Hussaini, 1993), pages 50–68.
- Croft, N. (1998). *Unstructured Mesh - Finite Volume Algorithms for Swirling, Turbulent, Reacting Flows*. PhD thesis, University of Greenwich.

- Croft, N., Pericleous, K., and Cross, M. (1995). PHYSICA: A multiphysics environment for complex flow processes. In Taylor, C. et al., editors, *Num. Meth. Laminar & Turbulent Flow '95*, volume 9, part 2, page 1269. Pineridge Press, U. K.
- Curle, N. (1955). The influence of solid boundaries upon aerodynamic sound. *Proc. Roy. Soc., A* 231:505.
- Degani, A. and Fox, G. (1996). Parallel multigrid computation of the unsteady incompressible Navier-Stokes equations. *Journal of Computational Physics*, 128:223–236.
- Dempsey, S. and Pericleous, K. (1995). Turbulent flow simulation using a fractal-based LES model. *ASME FED*, 208:61–66.
- Djambazov, G., Lai, C.-H., and Pericleous, K. (1997a). Development of a domain decomposition method for computational aeroacoustics. In *DD9 Proceedings*. John Wiley & Sons.
- Djambazov, G., Lai, C.-H., and Pericleous, K. (1997b). Domain decomposition methods for some aerodynamic noise problems. In *3rd AIAA/CEAS Aeroacoustics Conference*, number 97-1608 in AIAA Papers, pages 191–198.
- Djambazov, G., Lai, C.-H., and Pericleous, K. (1997c). Testing a linear propagation module on some acoustic scattering problems. In (Tam and Hardin, 1997), pages 221–229.
- Djambazov, G., Lai, C.-H., and Pericleous, K. (1998a). Development of numerical techniques for near-field aeroacoustic computations. *International Journal for Numerical Methods in Fluids*.
- Djambazov, G., Lai, C.-H., and Pericleous, K. (1998b). Efficient computation of aerodynamic noise. In *Contemporary Mathematics*, volume 218, pages 506–512. American Mathematical Society.
- Djambazov, G., Lai, C.-H., and Pericleous, K. (1998c). Staggered-mesh computation for aerodynamic sound. In *4th AIAA/CEAS Aeroacoustics Conference, part 1*, number 98-2219 in AIAA Papers, pages 97–103.

- Dong, T., Shih, S., Mankbadi, R., and Povinelli, L. (1997). A numerical study of duct geometry effect on radiation of engine internal noise. In *3rd AIAA/CEAS Aeroacoustics Conference*, number 97-1604 in AIAA Papers.
- Farassat, F. and Myers, M. (1988). Extension of Kirchhoff's formula to radiation from moving surfaces. *Journal of Sound and Vibration*, 123(3):451-460.
- FEMGEN (1992). *FEMGEN/FEMVIEW, Version 2*. Femview Limited, Leicester, UK.
- Ffowcs Williams, J. and Dowling, A. (1996). A. Acoustics, Aerodynamics and Fluid Mechanics. In *University of Cambridge Department of Engineering Annual Report 1995/96*. Cambridge.
- Ffowcs Williams, J. and Hawkings, D. (1969). Sound generation by turbulence and surfaces in arbitrary motion. *Philosophical Transactions of the Royal Society*, 264 A:321-342.
- Flow3d (1995). Flow3d, Version 2.3.2. AEA Technology. Harwell, England.
- Goldstein, M. (1984). Aeroacoustics of turbulent shear flows. *Ann. Rev. Fluid Mech.*, 16:263-285.
- Goldstein, M. (1993). *Aeroacoustics*. McGraw-Hill.
- Goldstein, M. and Mankbadi, R. (1993). Calculation of noise produced by high Mach number jets. In (Hardin and Hussaini, 1993), pages 207-215.
- Goodrich, J. W. (1997). High accuracy finite difference algorithms for computational aeroacoustics. In *3rd AIAA/CEAS Aeroacoustics Conference*, number 97-1584 in AIAA Papers.
- Hardin, J. and Hussaini, M., editors (1993). *Computational Aeroacoustics*. Springer-Verlag New York, Inc.
- Hardin, J. and Pope, D. (1994). An acoustic/viscous splitting technique for computational aeroacoustics. *Theoretical and Computational Fluid Dynamics*, 6:323-340.
- Hardin, J. and Pope, D. (1995). Sound generation by flow over a two-dimensional cavity. *AIAA Journal*, 33:407-412.

- Hardin, J. C. (1993). Recent insights into computational aeroacoustics. In (Mankbadi et al., 1993), page 1.
- Hayder, M. and Hagstrom, T. (1995). An outflow boundary condition for aeroacoustic computations. In (Lyrintzis et al., 1995), pages 41–46.
- Hayder, M., Hu, F., and Hussaini, M. (1997). Towards perfectly absorbing boundary conditions for Euler equations. Technical Report 97-25, ICASE.
- Hesthaven, J. (1997). The analysis and construction of perfectly matched layers for linearized Euler equations. Technical Report 97-49, ICASE.
- Hirsch, C. (1990). *Numerical Computation of Internal and External Flows, Volume 2*. John Wiley & Sons.
- Hixon, D., Shih, S., and Mankbadi, R. (1995). Evaluation of boundary conditions for computational aeroacoustics. *AIAA Journal*, 33(11):2006–12.
- Hixon, R. (1996). On increasing the accuracy of MacCormack schemes for aeroacoustic applications. Contractor Report 202311, NASA.
- Howe, M. (1975). Contributions to the theory of aerodynamic sound with application to excess jet noise and the theory of the flute. *J. Fluid Mech.*, 71(4):625–673.
- Kato, C., Iida, A., and Ikegawa, M. (1995). Numerical simulation of aerodynamic sound radiated from low Mach number turbulent wakes. In (Lyrintzis et al., 1995), pages 53–58.
- Kurbatskii, K. A. (1997). Analytical solutions of the category 1, benchmark problems 1 and 2. In (Tam and Hardin, 1997), pages 9–14.
- Leonard, B. (1979). A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer Methods in Applied Mechanics and Engineering*, 19:59–98.
- Lighthill, J. M. (1952). On sound generated aerodynamically: I. General theory. *Proceedings of the Royal Society*, 211 A:564–587.
- Lighthill, J. M. (1954). On sound generated aerodynamically: II. Turbulence as a source of sound. *Proc. Roy. Soc., A* 222:1–32.

- Lighthill, J. M. (1978). *Waves in Fluids*. Cambridge University Press.
- Lilley, G. (1974). On the noise from jets. In *Noise Mechanisms*, number 131 in CP, pages 13.1–13.12. AGARD.
- Lyrantzis, A. et al., editors (1995). *Computational Aeroacoustics*, volume FED-219. ASME.
- Lyrantzis, A. S. (1993). The use of Kirchhoff's method in computational aeroacoustics. In (Mankbadi et al., 1993), pages 53–61.
- Lyrantzis, A. S. and Mankbadi, R. (1996). Prediction of the far-field jet noise using Kirchhoff's formulation. *AIAA Journal*, 34(2):413–416.
- Mankbadi, R. et al., editors (1993). *Computational Aero- and Hydro-acoustics*, volume FED-147. ASME.
- Morris, P., Wang, Q., Long, L., and Lockard, D. (1997). Numerical predictions of high speed jet noise. In *3rd AIAA/CEAS Aeroacoustics Conference*, number 97-1598 in AIAA Papers.
- Morton, K. and Mayers, D. (1994). *Numerical Solution of Partial Differential Equations*. Cambridge University Press.
- Patankar, S. V. (1980). *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Co.
- Phillips, O. (1960). On the generation of sound by supersonic shear layers. *J. Fluid Mech.*, 9:1–28.
- PHOENICS (1995). *PHOENICS, Version 2.1.3*. CHAM Ltd, Wimbledon, UK.
- Pierce, A. (1981). *Acoustics: an introduction to its physical principles and applications*. McGraw-Hill.
- Pierce, A. (1993). Validation methodology: Review and comments. In (Hardin and Hussaini, 1993), pages 169–173.
- Pilon, A. and Lyrantzis, A. S. (1997). Refraction corrections for the Kirchhoff method. In *3rd AIAA/CEAS Aeroacoustics Conference*, number 97-1654 in AIAA Papers.

- Powell, A. (1964). Theory of vortex sound. *J. Acoust. Soc. America*, 36:177–195.
- Reitsma, S., Manro, V., and Tureaud, T. (1993). Non-reflective boundary conditions for fluid-acoustic simulations using finite volume formulation. In (Mankbadi et al., 1993), pages 71–82.
- Sankar, L., Reddy, N., and Hariharan, N. (1993). A comparative study of numerical schemes for aeroacoustic applications. In (Mankbadi et al., 1993), pages 35–40.
- Sarkar, S. and Hussaini, M. (1993). A hybrid direct numerical simulation of sound radiated from isotropic turbulence. In (Mankbadi et al., 1993), pages 83–89.
- Shih, S., Hixon, D., and Mankbadi, R. (1995). A zonal approach for prediction of jet noise. In *1st CEAS/AIAA Aeroacoustics Conference*, number 95-144 in AIAA Papers.
- Spyropoulos, E. and Holmes, B. (1997). Large-Eddy Simulation of a high Reynolds number flow around a cylinder including aeroacoustic predictions. In (Tam and Hardin, 1997), pages 319–328.
- SYSNOISE (1995). *SYSNOISE, Rev 5.2*. Numerical Integration Technologies, Leuven, Belgium.
- Tam, C. (1995). Computational aeroacoustics: Issues and methods. *AIAA Journal*, 33(10):1788–1796.
- Tam, C. (1997). Numerical methods in computational aeroacoustics. In *Computational Aeroacoustics: Methods and Applications*, Short Course, pages 1–78. AIAA, Atlanta, Georgia.
- Tam, C. and Dong, Z. (1993). Solid wall boundary conditions for computational aeroacoustics. In (Mankbadi et al., 1993), pages 63–70.
- Tam, C. and Hardin, J., editors (1997). *Second Computational Aeroacoustics Workshop on Benchmark Problems*. Number 3352 in Conference Publications. NASA.
- Tam, C. and Webb, J. (1993). Dispersion-relation-preserving finite difference schemes for computational acoustics. *Journal of Computational Physics*, 107:262–281.

- Turner, J. and Pretlove, A. (1991). *Acoustics for Engineers*. MacMillan.
- van Leer, B. (1977). Towards the ultimate conservative difference scheme. VI: A new approach to numerical convection. *Journal of Computational Physics*, 23:276–299.
- Versteeg, H. and Malalasekera, W. (1995). *An Introduction to Computational Fluid Dynamics. The finite volume method*. Longman.
- Viswanathan, K. and Sankar, L. (1995). Numerical simulation of airfoil noise. In (Lyrintzis et al., 1995), pages 65–70.
- Yokono, Y. and Fujita, H. (1995). Interactive steering of supercomputer simulation for aerodynamic noise radiated from square cylinder. In (Lyrintzis et al., 1995), pages 59–64.
- Zhang, X., Rona, A., and Lilley, G. (1995). Far-field noise radiation from an unsteady supersonic cavity flow. In *1st CEAS/AIAA Aeroacoustics Conference*, number 95-040 in AIAA Papers.
- Zingg, D., Lomax, H., and Jurgens, H. (1996). High-accuracy finite-difference schemes for linear wave propagation. *SIAM J. Sci. Comput.*, 17(2):328–346.

# Appendix A

## Pseudo-Code of the Acoustic Module

### Main Program

```
Read model parameters (big_steps, small_steps)
Set domain boundary parameters
Adjust scheme coefficients

Get mean flow data
Initialise model sources
Set radiation boundary parameters

time_step = 0
For big_step = 1 to big_steps do
  For small_step = 1 to small_steps do
    time_step = time_step + 1
    Step with pressure (time_step)
    Fix pressure (time_step)
    Step with velocity (time_step)
    Averaging (time_step)
  end do
  General data output (big_step, time_step)
  Model-specific data output (big_step, time_step)
end do
```



## Read model parameters

Read from model data file:

Integers: `big_steps`, `small_steps`

Logicals: `radiating(side)`, `side` = {East, West, North, South, High, Low}

Reals: `c0`, `Δt`

Integer `nx`; Reals: `Δx`, `x0`, `u0`

Integer `ny`; Reals: `Δy`, `y0`, `v0`

Integer `nz`; Reals: `Δz`, `z0`, `w0`

Reals: `δ1`, `δ0`

Initialise mean-flow data arrays with `u0`, `v0`, `w0`, and `c0`

Initialise to zero solid-face arrays: `wall_u`, `wall_v`, and `wall_w`

## Set domain boundary parameters

For `side` = {East, West} do

`radiating_layer(side)` = 3

    If not `radiating(side)` then

`radiating_layer(side)` = 0

`wall_u(side)` = 1

    end

end do

For `side` = {North, South} do

`radiating_layer(side)` = 3

    If not `radiating(Side)` then

`radiating_layer(side)` = 0

`wall_v(side)` = 1

    end

end do

For `side` = {High, Low} do

`radiating_layer(side)` = 3

    If not `radiating(side)` then

`radiating_layer(side)` = 0

`wall_w(side)` = 1

    end

end do

## Adjust scheme coefficients

Assuming that  $\delta_1$  has been specified according to Figure 4.1:

With 10 points per wavelength:  $\delta_1 = 0.04972569$

With 16 points per wavelength:  $\delta_1 = 0.04796525$

With 20 points per wavelength:  $\delta_1 = 0.04756939$

Set staggered-scheme derivation coefficients according to the restrictions (4.9):

$$a_1 = 9/8 + \delta_1$$

$$a_2 = -0.5(1/12 + \delta_1)$$

$$a_3 = 0.1 \delta_1$$

Assuming that  $\delta_0$  has been specified according to Figure 4.2:

With 25 points per cycle:  $\delta_0 = 0.08230$

With 50 points per cycle:  $\delta_0 = 0.08307$

With 63 points per cycle:  $\delta_0 = 0.08317$

Set staggered-scheme temporal integration coefficients according to (4.13):

$$b_0 = 1 + \delta_0$$

$$b_1 = 1/24 - 3 \delta_0$$

$$b_2 = 3 \delta_0 - 1/12$$

$$b_3 = 1/24 - \delta_0$$

End

## Get mean flow data

This subroutine is called to initialise *non-uniform* model-specific mean flow into the 3-dimensional arrays:

$$cx = \bar{v}_1 \Delta t / \Delta x$$

$$cy = \bar{v}_2 \Delta t / \Delta y$$

$$cz = \bar{v}_3 \Delta t / \Delta z$$

$$c = \gamma \bar{p} / \bar{\rho}.$$

Time-dependent mean-flow adjustment should go in “Averaging” or “Model-specific data output”.

## Initialise model sources

This call is provided for the initialisation of any variables which the model-specific subroutines may need during the simulation.

Also here, the geometry of the model problem has to be defined by setting the appropriate solid-face array elements. As an example, the specification of the cylinder in Figure 4.7 with diameter  $D = 1$  and a centre in the origin is included:

```

 $x_C = 0$ 
 $y_C = 0$ 
For  $j = 1$  to  $ny$  do
     $y = y_0 + (j - 0.5) \Delta y$ 
For  $i = 1$  to  $nx$  do
     $x = x_0 + (i - 0.5) \Delta x$ 
     $r = \sqrt{(x - x_C)^2 + (y - y_C)^2}$ 
    if  $r < 0.5$  then
        wall_u( $i - 1, j, 1$ ) = 1
        wall_u( $i, j, 1$ ) = 1
        wall_v( $i, j - 1, 1$ ) = 1
        wall_v( $i, j, 1$ ) = 1
    end
end do
end do

```

The acoustic radiation direction has to be specified here for every cell in the radiating 3-cell layer at the outer boundaries of the domain. This is achieved by assigning source-point coordinates into the 3-dimensional arrays:  $rad_x$ ,  $rad_y$ ,  $rad_z$ . For the cylinder problem, the peak location of the initial pressure pulse is prescribed as the source point for all cells  $(i, j, k)$ :

```

 $rad_x(i, j, k) = 4.0$ 
 $rad_y(i, j, k) = 0.0$ 
 $rad_z(i, j, k) = z_0 + 0.5 \, nz \, \Delta z$ 

```

Initial conditions of the linearised Euler equations (2.11) and (2.12) can be specified at this stage into the 3D variable arrays  $\mathbf{p}$ ,  $\mathbf{u}$ ,  $\mathbf{v}$ , and  $\mathbf{w}$  which are otherwise initialised to zero.

## Set radiation boundary parameters

For every cell  $(i, j, k)$  in the outer radiating layer do

$$x_C = \text{rad\_x}(i, j, k)$$

$$y_C = \text{rad\_y}(i, j, k)$$

$$z_C = \text{rad\_z}(i, j, k)$$

$$x = x_0 + (i - 0.5)\Delta x$$

$$y = y_0 + (j - 0.5)\Delta y$$

$$z = z_0 + (k - 0.5)\Delta z$$

$$r = \sqrt{(x - x_C)^2 + (y - y_C)^2 + (z - z_C)^2}$$

Implementation of equation (4.16):

$$\cos \alpha = (x - x_C)/r$$

$$\cos \beta = (y - y_C)/r$$

$$\cos \gamma = (z - z_C)/r$$

$$\delta = cx(i, j, k) \Delta x \cos \alpha + cy(i, j, k) \Delta y \cos \beta + cz(i, j, k) \Delta z \cos \gamma + c(i, j, k) \Delta t$$

$$\text{rad\_x}(i, j, k) = \delta / (\cos \alpha \Delta x)$$

$$\text{rad\_y}(i, j, k) = \delta / (\cos \beta \Delta y)$$

$$\text{rad\_z}(i, j, k) = \delta / (\cos \gamma \Delta z)$$

end do

## Step with pressure

Spatial derivation:

For every non-blocked internal cell  $(i, j, k)$  do

(‘Internal cells’ are those which do not belong to the 3-node radiating layer along each side of the domain.)

Calculate the staggered spatial derivatives of the propagation terms of the continuity equation (2.11) using the mirroring procedures for perpendicular velocity components:

```
call uew(i,j,k, 0, ue)
```

```
call uew(i,j,k, 1, uw)
```

```
dux = dersta(uw(3), uw(2), uw(1), ue(1), ue(2), ue(3) )
```

```
call vns(i,j,k, 0, ue)
```

```
call vns(i,j,k, 1, uw)
```

```
dvy = dersta(uw(3), uw(2), uw(1), ue(1), ue(2), ue(3) )
```

```
call whl(i,j,k, 0, ue)
```

```
call whl(i,j,k, 1, uw)
```

```

dwz = dersta(uw(3), uw(2), uw(1), ue(1), ue(2), ue(3) )
dpsta(i,j,k,0) = -c(i,j,k)*(dt_dx*dux + dt_dy*dvy + dt_dz*dwz)

```

Calculate the non-staggered pressure convection derivatives with the basic mirroring procedures:

```

call pEW(p, i,j,k, pp)
dpx = dernon( pp(-3), pp(-2), pp(-1), pp(1), pp(2), pp(3) )
call pNS(p, i,j,k, pp)
dpy = dernon( pp(-3), pp(-2), pp(-1), pp(1), pp(2), pp(3) )
call pHL(p, i,j,k, pp)
dpz = dernon( pp(-3), pp(-2), pp(-1), pp(1), pp(2), pp(3) )
dpnon(i,j,k,0) = -cx(i,j,k)*dpx - cy(i,j,k)*dpy - cz(i,j,k)*dpz

```

end do

Interpolate radiating layer non-staggered pressure (dpnon)

Interpolate radiating layer staggered pressure (dpsta)

Temporal integration:

```
t_source = (time_step - 0.5) Δt
```

For every non-blocked cell (i, j, k) do

( mback = 3 is the number of stored old time levels )

```

For m = 0 to mback do
  d(m) = dpnon(i,j,k, m)
  s(m) = dpsta(i,j,k, m)
end do
p(i,j,k) = p(i,j,k) +
  quanon( d(0), d(1), d(2), d(3) ) +
  quasta( s(0), s(1), s(2), s(3) ) +
  source( i, j, k, time_step, t_source )

For m = 1 to mback do
  dpnon(i,j,k, m) = d(m-1)
  dpsta(i,j,k, m) = s(m-1)
end do

```

end do

## Numerical scheme functions

Non-staggered spatial derivative:

```
function dernon(W3,W2,W1, E1,E2,E3)
parameter (a1 = 0.77088238051822552,
*          a2 = -0.166705904414580469,
*          a3 = 0.02084314277031176 )
dernon = a1*(E1 - W1) + a2*(E2 - W2) + a3*(E3 - W3)
end
```

Staggered spatial derivative:

```
function dersta(w3,w2,w1, e1,e2,e3)
include 'sch.cmn'
dersta = a1*(e1 - w1) + a2*(e2 - w2) + a3*(e3 - w3)
end
```

Non-staggered temporal integral:

```
function quanon(der0, der1, der2, der3)
parameter (b0 = 2.3025580888 ,
*          b1 = -2.4910075998 ,
*          b2 = 1.5743409332 ,
*          b3 = -0.3858914222 )
quanon = b0*der0 + b1*der1 + b2*der2 + b3*der3
end
```

Staggered temporal integral:

```
function quasta(der0, der1, der2, der3)
include 'sch.cmn'
quasta = b0*der0 + b1*der1 + b2*der2 + b3*der3
end
```

## Mirroring procedures

### East-West pressure (pEW)

Given the 3D pressure array  $\mathbf{p}$ , the u-velocity solid face map  $\mathbf{wall\_u}$ , and a centre cell  $(iC, j, k)$ , find and store in 1D array  $\mathbf{pvec}$  the neighbouring values with mirroring at every solid face encountered:

```

pvec(0) = p(iC, j, k)
For direction = -1 to 1 step 2 do
  i = iC
  increment = direction
  For neighbour = 1 to 3 do
    look = i
    if ( increment < 0 ) look = i - 1
    if wall_u(look, j, k) then
      increment = -increment
    else
      i = i + increment
    end if
    pvec(direction*neighbour) = p(i, j, k)
  end do
end do

```

### East-West u-velocity (uew)

Given the 3D array  $\mathbf{u}$ , the u-velocity solid face map  $\mathbf{wall\_u}$ , a centre cell  $(ip, j, k)$ , and **left** key (0 or 1), find and store in 1D array  $\mathbf{uvec}$  the neighbouring u-values with mirroring at every solid face encountered:

```

i = ip - left
uvec(1) = u(i, j, k)
increment = 1 - 2*left
sign = 1
For neighbour = 2 to 3 do
  if wall_u(i, j, k) then
    increment = -increment

```

```

        sign = -sign
    end if
    i = i + inc
    uvec(neighbour) = sign*u(i, j, k)
end do

```

### North-South pressure (pNS)

Given the 3D pressure array **p**, the v-velocity solid face map **wall\_v**, and a centre cell (i, jC, k), find and store in 1D array **pvec** the neighbouring values with mirroring at every solid face encountered:

```

pvec(0) = p(i, jC, k)
For direction = -1 to 1 step 2 do
    j = jC
    increment = direction
    For neighbour = 1 to 3 do
        look = j
        if ( increment < 0 ) look = j - 1
        if wall_v(i, look, k) then
            increment = -increment
        else
            j = j + increment
        end if
        pvec(direction*neighbour) = p(i, j, k)
    end do
end do

```

### North-South v-velocity (vns)

Given the 3D array **v**, the v-velocity solid face map **wall\_v**, a centre cell (i, jp, k), and **left** key (0 or 1), find and store in 1D array **vec** the neighbouring v-values with mirroring at every solid face encountered:

```

j = jp - left
vec(1) = v(i, j, k)
increment = 1 - 2*left

```



```

sign = 1
For neighbour = 2 to 3 do
  if wall_v(i, j, k) then
    increment = -increment
    sign = -sign
  end if
  j = j + inc
  vec(neighbour) = sign*v(i, j, k)
end do

```

The remaining two mirroring procedures for High-Low pressure (pHL) and for High-Low w-velocity (whl) are implemented in the same way as those shown above.

To obtain East-West  $\mathbf{v}$ -velocity, call pEW with  $\mathbf{v}$  instead of  $\mathbf{p}$ ; to obtain High-Low  $\mathbf{u}$ -velocity, call pHL with  $\mathbf{u}$  instead of  $\mathbf{p}$ , etc.

## Step with velocity

For the 3 velocity variables ( $u$ ,  $v$ ,  $w$ ) do the following:

```

For all non-blocked internal cell faces do
  Obtain neighbouring pressure values with mirroring
  Calculate staggered pressure derivatives and store (dusta, dvsta, dwsta)
  Obtain neighbouring velocity values with mirroring
  Calculate and store non-staggered spatial derivatives (dunon, dvnnon, dwnon)
end do

```

```

Interpolate radiating layer non-staggered velocity derivatives (dunon, etc.)
Interpolate radiating layer staggered velocity derivatives (dusta, dvsta, dwsta)

```

Temporal integration:

```

For every non-blocked cell face do
  Calculate non-staggered temporal integrals
  Calculate staggered temporal integrals
  Update velocity variables
  Update back-storage arrays of the velocity derivatives
end do

```

## Radiating boundary interpolation

### Interpolate radiating layer

Second order interpolating function:

$$\text{pInter}(fL, f, fR, x) = f + 0.5 * x * (fR - fL + x * (fL + fR - 2.0 * f))$$

Operating on a 4-dimensional storage array  $a(i, j, k, m)$

with  $m=0$  at the new time step, and  $m=1$  at the old time step:

For every node in the East radiating layer do

$$a(i, j, k, 0) = \text{pInter}(a(i-2, j, k, 1), a(i-1, j, k, 1), a(i, j, k, 1), 1 - \text{rad}_x(i, j, k))$$

end do

For every node in the West radiating layer do

$$a(i, j, k, 0) = \text{pInter}(a(i+2, j, k, 1), a(i+1, j, k, 1), a(i, j, k, 1), 1 + \text{rad}_x(i, j, k))$$

end do

For every node in the North radiating layer do

$$a(i, j, k, 0) = \text{pInter}(a(i, j-2, k, 1), a(i, j-1, k, 1), a(i, j, k, 1), 1 - \text{rad}_y(i, j, k))$$

end do

For every node in the South radiating layer do

$$a(i, j, k, 0) = \text{pInter}(a(i, j+2, k, 1), a(i, j+1, k, 1), a(i, j, k, 1), 1 + \text{rad}_y(i, j, k))$$

end do

For every node in the High radiating layer do

$$a(i, j, k, 0) = \text{pInter}(a(i, j, k-2, 1), a(i, j, k-1, 1), a(i, j, k, 1), 1 - \text{rad}_z(i, j, k))$$

end do

For every node in the Low radiating layer do

$$a(i, j, k, 0) = \text{pInter}(a(i, j, k+2, 1), a(i, j, k+1, 1), a(i, j, k, 1), 1 + \text{rad}_z(i, j, k))$$

end do

## General data output

Write pressure and velocity fields to disk files.

Dump buffered time dependent values in consecutive disk files. The variables to be stored at each time step in the real arrays: *buf1*, *buf2*, *buf3*, and in the integer arrays: *ibuf*, *jbuf*, *kbuf*. This is most conveniently done in the Averaging subroutine which is called at the end of every time step.

Create plot files with the blocked cell faces at the South and at the Low boundary of the domain for displaying and debugging purposes.

## Model-specific coding

### Source function

Given the cell indices (i, j, k), the time step, and the current time, this subroutine has to calculate the right hand side of the continuity equation (2.11) for the cell (i, j, k). With no sources in the domain the code should read:

```
source = 0.0
```

### Fix pressure

This call is provided for overriding, if necessary, the calculated pressure in specific cells.

It can also be used for specifying source terms distributed over larger areas of the domain. For example, the acoustic expansion pressure source term (5.14) which is applied to all computational cells is calculated here from the current CFD values (pbar, ubar, rhobar) and the old time step CFD fields (oldub, oldpb):

```
cop = dt/dtcfid
cou = dt*c0*c0/dxcfd
uw = 0.
do i = 1, nxcfd
  ue = 0.5*( oldub(i) + ubar(i) )
  resC = cop*(pbar(i) - oldpb(i)) + cou*rhobar(i)*(ue - uw)
  p(i, 1, 1) = p(i, 1, 1) - resC
  uw = ue
end do
```

This code is valid if the linearised Euler and the CFD meshes are the same.

### Averaging

This subroutine is called after every completed time step (with the appropriate value of the counter time\_step).

Some quantities may have to be averaged continuously in time during the simulation. Others may need spatial averaging before storing for later output.

An example of temporal averaging is the sound pressure level (APL) in specified locations (i, j, k) with the cylinder scattering problem (Figures 3.8–3.9):

```
ratio = (time_step - 1) / time_step
APL = ratio*APL + (p(i, j, k))2 / time_step
```

An example of spatial averaging is the resonant cavity signal detected by the receiver membrane with distributed sensitivity (Figure 4.9):

```
signal = 0.0
number = 0
For all cells (i, j, k) belonging to receiver do
    ratio = number / (number + 1)
    signal = ratio*signal + local_sensitivity*p(i, j, k) / (number + 1)
    number = number + 1
end do
buf1(time_step) = source_signal
buf2(time_step) = signal
```

The buffer storage arrays buf1 and buf2 will later be saved on disk by the General data output.

This is also the place where adjustment to the current velocity variables can be added. For example, the momentum source term of the acoustic expansion (5.15) is calculated within this subroutine in way similar to the corresponding pressure source (see previous section), and the u-velocity variable is updated.

### **Model-specific data output**

Any disk files can be created at this stage since this subroutine is called at every 'big\_step' rather than at every time step.

For example, in order to produce Figure 4.5, the whole 1-dimensional pressure field was output 5 times.

## Appendix B

# Pseudo-Code of the PHYSICA Coupling

The coupling of the unstructured CFD code PHYSICA with the acoustic module is implemented through the model-specific coding (see Appendix A).

### Initial Settings with the Acoustic Module

Read mesh coordinates and patch information (aerofoil blockage)  
Calculate chord length and angle of attack  
Rotate mesh to the angle of attack around the leading edge  
Define as blocked all cells of the Cartesian acoustic mesh with centres inside aerofoil  
Define as acoustic source cells all immediate neighbours of the blockage  
Build an index of the CFD cells that are outside the aerofoil  
Build an index for averaging the mean flow quantities over the acoustic cells  
Read initial mean flow (CFD) fields (pm, um, vm)  
Do the New CFD time step settings  
Define centre for the radiating boundary conditions: the leading edge of the aerofoil

### New CFD time step settings

Copy (pm, um, vm) into (pold, uold, vold)  
Increment CFD time step counter  
Read current mean flow (CFD) fields (pm, um, vm)

```
For every acoustic cell do
  Local average of (pm, um, vm)
  Calculate isentropic density
  Update acoustic module mean flow arrays (c, cx, cy)
end do
```

## Acoustic Source Function

```
It the current cell is an acoustic source cell then
  Average the CFD pressure (pm) over the current cell
  Source = (pm_average - pold_average) * ( $\Delta t / \Delta t_{CFD}$ )
else
  Source = 0
end
```

## Averaging subroutine

(Called by the acoustic module after every *acoustic* time step)

```
Store acoustic pressure above and below leading edge for time-dependent output
If new CFD time step needed then
  Save acoustic pressure field to disk
  New CFD time step settings
end
```

If not all of the acoustic pressure field files are needed (or if the disk space is limited), the saving can be done by the 'Model-specific data output' subroutine which is called at every 'big\_step' rather than at every time step. The specified number of 'big steps' depends only on the user's needs for displaying and post-processing the solution.

```
For every acoustic cell do
  Local average of (pm, um, vm)
  Calculate isentropic density
  Update acoustic module mean flow arrays (c, cx, cy)
end do
```

## Acoustic Source Function

```
It the current cell is an acoustic source cell then
  Average the CFD pressure (pm) over the current cell
  Source = (pm_average - pold_average) * ( $\Delta t / \Delta t_{CFD}$ )
else
  Source = 0
end
```

## Averaging subroutine

(Called by the acoustic module after every *acoustic* time step)

```
Store acoustic pressure above and below leading edge for time-dependent output
If new CFD time step needed then
  Save acoustic pressure field to disk
  New CFD time step settings
end
```

If not all of the acoustic pressure field files are needed (or if the disk space is limited), the saving can be done by the 'Model-specific data output' subroutine which is called at every 'big\_step' rather than at every time step. The specified number of 'big steps' depends only on the user's needs for displaying and post-processing the solution.