

University of Southampton Research Repository ePrints Soton

Copyright © and Moral Rights for this thesis are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given e.g.

AUTHOR (year of submission) "Full thesis title", University of Southampton, name of the University School or Department, PhD Thesis, pagination

UNIVERSITY OF SOUTHAMPTON

Faculty of Engineering, Science and Mathematics

**Adaptive Mesh Refinement for
Computational Aeroacoustics**

by

Xun Huang

Submitted for the degree of
Doctor of Philosophy

October 2006

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE and MATHEMATICS

Doctor of Philosophy

ADAPTIVE MESH REFINEMENT FOR COMPUTATIONAL
AEROACOUSTICS

by Xun Huang

This thesis describes a parallel block-structured adaptive mesh refinement (AMR) method that is employed to solve some computational aeroacoustic problems with the aim of improving the computational efficiency. AMR adaptively refines and coarsens a computational mesh along with sound propagation to increase grid resolution only in the area of interest.

While sharing many of the same features, there is a marked difference between the current and the established AMR approaches. Rather than low-order schemes generally used in the previous approaches, a high-order spatial difference scheme is employed to improve numerical dispersion and dissipation qualities. To use a high-order scheme with AMR, a number of numerical issues associated with fine-coarse block interfaces on an adaptively refined mesh, such as interpolations, filter and artificial selective damping techniques and accuracy are addressed. In addition, the asymptotic stability and the transient behaviour of a high-order spatial scheme on an adaptively refined mesh are also studied with eigenvalue analysis and pseudospectra analysis respectively. In addition, the fundamental AMR algorithm is simplified in order to make the work of implementation more manageable.

Particular emphasis has been placed on solving sound radiation from generic aero-engine bypass geometry with mean flow. The approach of AMR is extended to support a body-fitted multi-block mesh. The radiation from an intake duct is modelled by the linearised Euler equations, while the radiation from an exhaust duct is modelled by the extended acoustic perturbation equations to suppress hydrodynamic instabilities generated in a sheared mean flow. After solving the near-field sound solution, the associated far-field sound directivity is estimated by solving the Ffowcs Williams–Hawkings equation. The overall results demonstrate the accuracy and the efficiency of the presented AMR method, but also reveal some limitations. The possible methods to avoid these limitations are given at the end of this thesis.

Contents

1	Introduction	1
1.1	Overview	1
1.2	Literature Review	3
1.2.1	Adaptive Mesh Refinement	3
1.2.2	Computational Aeroacoustics	9
1.2.3	Sound Radiation from Ducts	11
1.3	Thesis Structure	12
2	The AMR Algorithm	13
2.1	Introduction	13
2.2	Parallel AMR	21
2.2.1	Parallel Methodologies	21
2.2.2	Dynamic Load Balancing	22
2.2.3	Parallel Communications	26
2.3	Summary	31
3	Numerical Issues	32
3.1	Governing Equations	32
3.1.1	One-dimensional Advection Equation	32
3.1.2	Two-dimensional Wave Equations	33
3.1.3	Spinning Mode Radiation Equations	34
3.2	Spatial Discretization at Fine-Coarse Interfaces	37
3.3	Stability Analysis	39
3.4	Artificial Selective Damping and Filters	43
3.5	Convergence Rate	46
3.6	Summary	47
4	Results of Benchmark Problems	49
4.1	Introduction	49

4.2	Two-dimensional Acoustic Wave Propagation	50
4.3	Acoustic Wave Scattering from Cylinder	53
4.4	Spinning Mode Duct Radiation	57
4.5	Summary	64
5	Acoustic Radiation from Engine Intake Duct	66
5.1	Introduction	66
5.2	Problem Setup	68
5.2.1	Numerical Issues	68
5.2.2	Curvilinear Coordinate System	69
5.2.3	Far-field Directivity Prediction Method	70
5.2.4	Absorbing Numerical Noise	72
5.3	Results	74
5.3.1	Near-field Propagation and Far-field Directivity	74
5.3.2	Parallel Performance	80
5.4	Summary	82
6	Acoustic Radiation from Engine Exhaust Duct	84
6.1	Introduction	84
6.2	Problem Setup	86
6.2.1	Extended Acoustic Perturbation Equations	86
6.2.2	Numerical Issues	88
6.3	Results	90
6.3.1	AMR Working Procedure	90
6.3.2	Near-Field Propagation	91
6.3.3	Far-Field Directivity	93
6.4	Summary	94
7	Conclusions and Future Work	97
7.1	Concluding Remarks	97
7.2	Future Work	101
A	Coefficients of Numerical Schemes	102
A.1	Numerical Schemes	102
A.1.1	The DRP Scheme	102
A.1.1.1	The Central Stencil	103
A.1.1.2	The Single-side Stencil at Boundaries	103
A.1.2	The Standard Explicit Scheme	104

A.1.2.1	The Central Stencil	104
A.1.2.2	The Single-side Stencil at Boundaries	104
A.1.3	Interpolations	104
A.2	Operator Matrix in Eigenvalue Analysis	105
B	Asymptotic Solution of Sound Out of Jet	107
B.1	Introduction	107
B.2	Equations	108
B.3	Results and Discussion	109
C	Fourier Based Pseudospectral Method	111
C.1	Introduction	112
C.2	Governing Equations and Algorithm	114
C.2.1	Governing Equations	114
C.2.2	Governing Equations in The Pseudospectral Domain	115
C.2.3	Performance Analysis	116
C.3	Issues and Solutions	116
C.3.1	Points-per-wavelength Requirement	117
C.3.2	Hard-wall Boundary Condition	117
C.3.3	Absorbing Condition for The Rewrap Waves	119
C.4	Applications to Benchmark Problems	119
C.4.1	Wave Propagation and Reflection	120
C.4.2	Open Rotor	123
C.4.3	Summary	125
	Bibliography	128

List of Figures

1.1	The refinement operation of: (a-b) cell-based AMR; (c-d) patch-based AMR; (e-f) block-based AMR, where the thin lines denote meshes while the bold lines denote block boundaries.	4
2.1	Regridding operation of AMR.	14
2.2	Prolongation and restriction operations of AMR.	15
2.3	Quadtree represents the hierarchical relation of AMR.	15
2.4	The flowchart of AMR.	17
2.5	Ghost construction operation of AMR, where the neighbouring blocks have: (a) the same refinement level; (b) different refinement levels. . .	18
2.6	The compact memory model of the AMR ghost construction operation.	19
2.7	A CAA application of block-based AMR.	20
2.8	A dynamic load balancing example: (a)(c)(e) before balancing; (b)(d)(f) after balancing, where ($--\rightarrow$) represents space-filling curves.	24
2.9	The procedure of load migration: (a) initial imbalance state; (b) balancing by max-min load balancing; (c) balancing by space-filling load balancing.	25
2.10	The reconstruction of the quadtree with: (a) max-min load balancing; (b) space-filling load balancing.	25
2.11	A space-filling curve of a jet flow case solved with the AMR code: (a) vorticity contours; (b) space-filling curve.	27
2.12	Computing Morton number for an engine intake case: (a) the computational domain with sound solutions; (b) the corresponding abstract blocks on which the top number is the Morton number and the bottom pair is the block coordinate indices.	28
2.13	The parallel communication of AMR operations.	28
2.14	The simplified parallel communication of AMR operations.	29
2.15	The flowchart of parallel AMR, where two parts correspond to: (a) steps (3)–(4) and (b) steps (5)–(8) of Figure 2.4.	30

3.1	Benchmark applications: (a) advection; (b) wave propagation; (c) duct acoustic radiation.	33
3.2	Two approaches to compute spatial differences around a fine-coarse interface: (a) a multi-size DRP stencil; (b) a stencil with interpolation, where the shadowed side is the ghost cells area; (o) the original solutions on the block; (◆) the interpolated solutions of ghost cells. .	37
3.3	Gaussian pulse propagation through a fine-coarse block interface at $x=250$	38
3.4	Eigenvalue spectra of 4^{th} -order: (a) DRP and (b) standard explicit schemes, working on a uniform mesh with single-side stencils near domain boundaries, (o) $N=50$, (x) $N=100$, (+) $N=200$	40
3.5	Eigenvalue spectra of 4^{th} -order: (a) DRP and (b) standard explicit schemes, working on a hierarchical mesh with single-side stencils near domain boundaries, using a 4^{th} -order interpolation method around the fine-coarse interface, $N=50$	41
3.6	Spectrum (black dots) and ϵ -pseudospectra distribution boundaries (coloured lines) of an operator matrix obtained from a 4^{th} -order standard explicit scheme on: (a) a uniform mesh; (b) a hierarchical mesh with a fine-coarse interface around which a 4^{th} -order interpolation method is employed.	42
3.7	Spectrum (black dots) and ϵ -pseudospectra distribution boundaries (coloured lines) of an operator matrix obtained from a 4^{th} -order DRP scheme on: (a) a uniform mesh; (b) a hierarchical mesh with a fine-coarse interface around which a 4^{th} -order interpolation method is employed.	42
3.8	One-dimensional Gaussian pulse propagation with a 10^{th} -order explicit filter and a 4^{th} -order Compact scheme.	44
3.9	Two-dimensional Gaussian pulse propagation with an artificial selective damping method and a 4^{th} -order DRP scheme, where pressure contours with 6 levels between ± 0.05 are displayed.	45
3.10	Pressure contours: (a) $t=0.1$; (b) $t=4$, where there are sixteen contour levels between ± 0.05 . Gray lines are blocks' borders. Solutions on the centre dark line in (b) are compared to the analytical solution. . . .	46

3.11	L_2 -norm error of the pressure solution, where \mathbf{N} is the number of cells on each block border. Several schemes employed are: (\triangle) the 2^{nd} -order standard explicit scheme and the 2^{nd} -order interpolation; (X) the 4^{th} -order standard explicit scheme and the 2^{nd} -order interpolation; (\diamond) the 4^{th} -order standard explicit scheme and the 6^{th} -order interpolation; (+) the 4^{th} -order DRP scheme and the 6^{th} -order interpolation; (-.-) the 2^{nd} -order ideal slope; (- -) the 4^{th} -order ideal slope.	47
4.1	Parallel computational results of two-dimensional acoustic propagation: (a) pressure perturbation contours; (b) the load distribution among six CPUs.	51
4.2	The computational time percentage of the code in the case study of two-dimensional acoustic propagation working on a single CPU. . . .	51
4.3	Time percentage of every subroutine in the case study of two-dimensional acoustic propagation.	52
4.4	Performance speedup in the case study of two-dimensional acoustic propagation.	53
4.5	A schematic of the immersed boundary method.	54
4.6	The two-dimensional acoustic scattering off a cylinder.	55
4.7	Sound pressure contours and dynamic load balancing in the case study of two-dimensional acoustic scattering off a cylinder, where (a-b) $t = 6.125$, (c-d) $t = 8.75$	56
4.8	The AMR regridding flag of the two-dimensional acoustic scattering off a cylinder.	57
4.9	The pressure history at three observer points demonstrates the performance of several cut-cell finite-difference approaches of the immersed boundary method.	58
4.10	Schematic of the computation domain for the case study of sound radiation from an axial symmetrical duct.	59
4.11	The spinning mode sound radiation from an unflanged duct solved on a uniform mesh: (a) contours of pressure perturbation; (b) contours of radial velocity perturbation, where $m = 4$, $n = 1$, $k = 10$	60
4.12	Contours of pressure perturbation of the spinning mode sound radiation from an unflanged duct on an adaptively refined mesh, where $m = 4$, $n = 1$, $k = 10$	62
4.13	Pressure perturbation of the spinning mode sound radiation from an unflanged duct with two regridding thresholds: (a) $\tau_r = 0.15$, $\tau_c = 0.1$; (b) $\tau_r = 0.015$, $\tau_c = 0.01$, where $m = 4$, $n = 1$, $k = 10$, $t = 14.67$	63

4.14	Instantaneous pressure perturbation of the spinning mode sound radiation from an unflanged duct: (.-) solution computed by the big threshold; (o) solution computed by the small threshold; (–) solution computed in the finest uniform mesh, where $m = 4$, $n = 1$, $k = 10$, $t = 14.67$	63
4.15	Comparison of sound directivity patterns between AMR prediction and analytical solution for the case of the spinning mode sound radiation from an unflanged duct, where $m = 4$, $n = 1$, $k = 10$	64
5.1	Schematic of noise radiation off an aero-engine bypass duct, where: 1 is rotor and 2 is stator.	67
5.2	Mach number contours of the mean flow around an aero-engine intake: free stream Mach number is 0.25; ambient pressure is 94250 Pa; intake Mach number is set to 0.55 and intake pressure is 79687 Pa.	68
5.3	Three-dimensional FW–H integral surface around an engine duct. . .	70
5.4	Two integral surfaces are placed in the computational domain for the sensitivity study of the FW–H solver.	71
5.5	Directivity plot for different FW–H surface placements and azimuthal grids number (Z_{num}) in the case of the general intake: (\triangle) surface 1 and $Z_{num} = 11$; (–) surface 1 and $Z_{num} = 41$; (– –) surface 1 and $Z_{num} = 61$; (\diamond) surface 2 and $Z_{num} = 11$; (.-) surface 2 and $Z_{num} = 41$; (.-.-) surface 2 and $Z_{num} = 61$, where $m = 4$, $n = 1$, $k = 20$ with the stationary mean flow.	72
5.6	Effect of spurious wave treatment methods for a spinning mode sound radiation from an aero-engine intake duct, where perturbation pressure contours are displayed around the lip of the intake, gray lines denote the boundaries of blocks on the adaptively refined mesh; $m = 12$, $n = 1$, $k = 20$	73
5.7	The pressure perturbation contours around the aero-engine intake duct, computed on either an adaptively refined mesh or a uniformly fine mesh, where: (a) shows the whole domain; (b) is an enlarged part; (.-) the AMR result; (–) the result computed on the uniform mesh, $m = 12$, $n = 1$, $k = 20$, $t = 2.8$. The bottom bold line displayed in (b) is for Figure 5.8.	75
5.8	Comparing instantaneous pressure perturbations on a selected line (as shown in Figure 5.7(b)) for the intake case, where: (.-) denotes the AMR result; (–) is the result on a uniformly fine mesh.	76

5.9	Perturbation pressure contours around the aero-engine intake duct, where $m = 26, n = 1, k = 41.9$, with the stationary medium and (a) LEE on a fixed mesh [153]; (b) LEE on an adaptively refined mesh.	77
5.10	Perturbation pressure contours, where $m = 13, n = 1, k = 16.7$, with the sideline mean flow and (a) LEE on a fixed mesh [153]; (b) LEE on an adaptively refined mesh.	78
5.11	Far-field directivities for the aero-engine intake radiation. (a): $m = 26, n = 1, k = 41.9$ with the stationary medium and (b) $m = 13, n = 1, k = 16.7$ with the sideline mean flow.	79
5.12	Perturbation pressure contours, $m = 12, n = 2, k = 20$. The computing load is distributed over 4 processors, which are denoted by different coloured areas.	80
5.13	The parallel speedup of the case of spinning mode sound radiation from an aero-engine intake with the sideline condition: (–) ideal speedup; (\triangle) the result of the SotonLEE code; (\diamond) the result of the AMR code.	81
6.1	The schematics of APE: (a) two-dimensional domain; (b) axisymmetrical three-dimensional domain.	87
6.2	Mean Mach number distribution of the aero-engine exhaust test case.	88
6.3	The problem setup of the aero-engine exhaust geometry that is displayed with thick lines.	89
6.4	The evolution of the adaptively refined mesh with the propagation of acoustic waves from the engine exhaust, where gray lines represent the block boundaries of the adaptively refined mesh, (a) $t=1$, (b) $t=2$, (c) $t=4$, and (d) $t=6$	90
6.5	Perturbation pressure contours computed by LEE and APE, where $m=13, n=1, k=28.3179$. (a) LEE with AMR, $t=12$, (b) APE with AMR, $t=20$	92
6.6	SPL contours computed by LEE and APE, where $m=13, n=1, k=28.3179$, (a) LEE, $9.5 < t < 10$, (b) APE, $15 < t < 15.5$	93
6.7	The APE prediction of perturbation pressures and SPL contours of several single spinning mode waves, where $m=13, n=2-4, k=28.3179, 12.4 < t < 12.9$ for SPL; (a) perturbation pressures, $n=2$; (b) SPL, $n=2$; (c) perturbation pressures, $n=3$; (d) SPL, $n=3$; (e) perturbation pressures, $n=4$; (f) SPL, $n=4$	95
6.8	Far-field directivity of the engine exhaust duct radiation, $m=13, k=28.3179$, (a) $n=1$, (b) $n=2$, (c) $n=3$, and (d) $n=4$	96

A.1	A stencil in the one-dimensional coordinate.	102
B.1	Setup of an idealised jet case.	107
B.2	Comparison of asymptotic and numerical solutions, where $(m, n) =$ $(4, 1), M_j = 0.14, k = 11.7$, (a) $M_0 = 0$, (b) $M_0 = 0.14$	109
C.1	A schematic of scaling of computation counts with grid size.	113
C.2	One-dimensional Gaussian pulse propagation with low points-per-wavelength. (a) PPW=4, $\Delta t/\Delta x = 0.1$, steps=200; (b) PPW=2, $\Delta t/\Delta x = 0.2$, steps=100.	117
C.3	One-dimensional Gaussian pulse reflected by a left hard-wall and with- out using buffer zone: (a) steps=10; (b) steps=150.	118
C.4	The propagation of a two-dimensional Gaussian pulse: (a) $t = 0.25$, without buffer zone; (b) $t = 0.4$, without buffer zone; (c) $t = 0.25$, with buffer zone; (d) $t = 0.4$, with buffer zone.	120
C.5	A comparison of two-dimensional Gaussian pulse propagation pre- dicted by the Fourier based pseudospectral method and the prefactored compact scheme, where pressures are along the $x = 0$ axis, $t = 0.4$: (a) without buffer zone; (b) with buffer zone.	121
C.6	A comparison of two-dimensional Gaussian pulse propagation after 110 time steps, where the pressure contours of (a) the Fourier PSTD method and (b) the compact scheme are displayed.	122
C.7	Pressure contours generated by an open rotor.	124
C.8	Pressure time history generated by an open rotor at $(x = -0.1, r = 1.9)$.124	
C.9	Prediction and analytical solution of the directivity of an open rotor acoustic radiation, where $r = 3.0$ and θ is the polar angle.	125

List of Tables

5.1	Summary of the incoming waves for the case of an intake duct.	69
5.2	The parallel communication and computation costs of the AMR computation of the aero-engine intake radiation.	82
6.1	Summary of the incoming spinning mode waves for the exhaust duct radiation.	89
6.2	Computation time comparison for the exhaust duct radiation.	91
A.1	The coefficients of a 4 th -order DRP scheme.	103
A.2	The coefficients of a 4 th -order standard explicit scheme.	104
A.3	The coefficients of several explicit interpolations.	105
C.1	Computing time t and L_2 error of the first benchmark problem. . . .	122
C.2	Computing time t and L_1 error of the second benchmark problem. . .	125

Acknowledgements

Foremost, thanks to my parents, Tuying Gao and Xiaolu Huang, for their constant support throughout my time at university and also to my wife, Yangyang Li, for her encouragement and support throughout the duration of my PhD studies.

I am very grateful to my supervisor, Professor Xin Zhang for his invaluable guidance, encouragement, support and enthusiasm throughout the course of this research.

I would like to thank Dr. Xiaoxian Chen for helpful discussion in this work. I would like to thank Professor Jeremy Astley for introducing the spinning mode sound source generated in rotor-stator interactions thoroughly to me. I would also like to thank Professor Geoffrey Lilley for introducing the fundamental problems in the area of computational aeroacoustics when I just joined the Aeronautics and Astronautics Department.

Thanks to my friends, including Zhaokai Ma, Sammie Chan and Ed Peers, for both providing precious suggestions to my research and sharing living experiences in the University of Southampton.

Nomenclature

Alphanumeric Symbols

c	Speed of sound
e	Internal energy
f	Frequency
J	Jacobian metric
J_m	m^{th} -order of first kind Bessel function
k	Wavenumber
k_a	Axial wavenumber
k_r	Radial wavenumber
k_x	x -direction wavenumber
k_y	y -direction wavenumber
L	Characteristic length scale/Buffer length
m	Azimuthal mode order
M	Mach number
n	Radial mode order
p	Pressure
t	Time variable
u, v, w	Velocity components
\mathbf{v}'	Perturbation velocity vector
\mathbf{V}_0	Mean velocity vector
x, y, z	Cartesian coordinates
x, r, θ	Cylindrical coordinates
Y_m	m^{th} -order of second kind Bessel function

Greek Symbols

γ	Ratio of specific heats, $c_p/c_v = 1.4$ for air
λ	Acoustic wavelength
ω	Angular frequency

ρ	Density
ξ, η	Curvilinear coordinates

Subscript or Superscript

$()_{\infty}$	Free stream/reference value
$()_0$	Mean flow value
$()'$	Perturbation value
$()_t$	Derivative with respect to time

Abbreviations

2.5D	Two-and-a-half Dimensional
APE	Acoustic Perturbation Equations
BEM	Boundary Element Method
CAA	Computational Aeroacoustics
CFL	Courant-Friedrichs-Levy
DRP	Dispersion Relation Preserving
FE	Finite Element
FEM	Finite Element Method
FW-H	Ffowcs Williams Hawking
IE	Infinite Element
LEE	Linearised Euler Equations
MPI	Message Passing Interface
PPW	Grid Points Per Wavelength
PSTD	Pseudospectral time-domain
RMS	Root Mean Square
SPL	Sound Pressure Level

Chapter 1

Introduction

1.1 Overview

The high level of nuisance noise generated by the take-off and landing of aircrafts has a significant impact on the communities near airports [1]. With a reduction target of perceived noise level of 50% by 2020 [2], computational aeroacoustics (CAA) is being used with increasing frequency in studying the physics of aerodynamically generated noise. Various attempts have been made to apply CAA methods to airframe/engine noise study both in the EU countries and in the US [3, 4]. The general objectives of CAA focus on the prediction of aerodynamic sound sources and the propagation of the generated sound. It has progressed considerably during the last decade and offers greater advantages over traditional methods such as computational fluid dynamics methods, in terms of accuracy and scope. However, a CAA method includes the use of long time accurate temporal integration strategies and high-order spatial schemes, which leads to high demands on computational resources [5]. More often than not its applications are still restricted to idealised geometries of aircraft components due to limited computational resource. The development of a cost effective and efficient computational method is essential.

Many aeroacoustic problems involving near-field sound generation and sound propagation include multiple spatial and temporal scales [6]. A generic problem is high-order spinning mode sound radiation from an aero-engine intake, where high level short wavelength sound pressures hug the wall posing a severe challenge to the grid resolution requirement. Preliminary studies have found that the size of the near wall grid must be an order of magnitude smaller than other areas of the duct depending on the local radial wavenumber [7, 8]. It requires a sufficient resolution of the computational grid in the near-field. The costs of such a computation could be pro-

hibitive using a uniformly fine grid. To meet the problem imposed by the disparate scales, one could use a highly stretched grid. However, the dispersion and dissipation characters of numerical schemes may be affected and it is highly likely that spurious waves will be generated by the numeric procedure [9]. Highly orthogonal and weakly stretched grids are therefore preferred to meet the grid requirements in the acoustic near- and far-fields [10].

The method of adaptive mesh refinement (AMR) attracts interests mainly due to its potential of reducing both the computational and storage cost over an equivalent static uniform mesh, as it increases grid resolution only in the area of interest [11]. A given spatial error tolerance is achieved by recursively refining meshes. Subsequently a localized mesh of high grid resolution is distributed within an otherwise coarse mesh. AMR has been actively applied to the computation of many different research areas, such as aerodynamics [11, 12, 13], plasma [14, 15, 16], combustion [17, 18], cosmology [19, 20], image processing [21], material analysis [22], and so forth. At present these large multi-physical problems are typically solved on distributed memory computers [23]. It requires the computational domain to be decomposed amongst parallel processors in a load-balanced manner. The strongly inhomogeneous grids of AMR pose a serious challenge to the design of a parallel load balancing algorithm, which has been studied and partially solved using either parallel software abstractions [24, 25, 26] with distributed dynamic data-structures [27] or a global memory model [28, 29, 30]. However, a general solution is still not available.

In this work AMR is employed with the aim of improving computational and memory efficiency in solving a class of CAA applications. The focus is particularly placed on the problem of sound propagation and radiation from an aero-engine nacelle. To achieve the aim, there are still gaps in both the established AMR programs and numerical issues which call for further work. Therefore the overall objectives are:

- to develop a parallel AMR framework to adaptively refine and coarsen a mesh and to balance the relevant computational loads dynamically;
- to design an appropriate approach that is capable of implementing high-order spatial schemes straightforwardly on an adaptively refined mesh;
- to test the proposed AMR method against CAA benchmark problems;
- to solve the practical problem of sound radiation from a generic aero-engine bypass duct by using the proposed AMR method.

The whole design is subjected to the principles of simplicity and portability.

1.2 Literature Review

AMR and CAA methods relevant to this work are surveyed below. With respect to each topic there are different methods that are compared through the review, which helps to identify the most suitable methods to be employed in the rest of this work.

1.2.1 Adaptive Mesh Refinement

AMR method can be performed on either unstructured or structured meshes. The technique of AMR working on unstructured meshes is especially suitable for applications with complex geometries. It has been used extensively in solving biomedical flow [31], aerodynamics [32, 33] and nonlinear dynamic systems [34]. By contrast, the technique of AMR working on structured meshes represents the corresponding computational domain as hierarchical refinement levels and increases mesh resolution mainly according to solution gradients [35] or immersed boundary curvatures [36, 37]. In addition, some other problem-specific adaptation indicators for the corresponding physical problems have been reported in [38, 39, 40]. Truncation error is also employed in [11]. Generally speaking, structured AMR has several distinctive advantages over unstructured AMR. Firstly, structured AMR is capable of treating problems not only in multiple spatial scales but also in multiple temporal scales [11]. Secondly, a structured mesh can be adaptively refined and coarsened without leading to skewness problems [10]. Thirdly, and the most important to this work, structured AMR can support high-order spatial schemes straightforwardly [18, 41, 42]. Hence the method of structured AMR is used here. For the sake of convenience, unless otherwise explicitly indicated, the name of AMR only applies to structured meshes in the rest of this work.

The AMR algorithm was firstly introduced by Berger and Oliger to solve unsteady Euler equations [11]. With this method there was an extensive improvement of computational efficiency for solving shock hydrodynamic problems [13]. Combined this AMR method with an adaptive mesh-moving algorithm a time-dependent problem of moving shock along a flat wall was simulated efficiently too [43]. It was also extended to solving applications governed by the Navier-Stokes equations [44]. An overall summary of AMR for both the Euler and Navier-Stokes Equations was presented in several publications [45, 46, 47]. In addition to studying numerical schemes on an adaptively refined mesh, the most recent interests were focused on improving AMR parallel efficiency [48, 49, 50, 51, 52], applying AMR to practical problems [53, 54, 55, 56] and presenting new AMR frameworks [57, 58, 59].

Along with the evolution of software and hardware techniques, the AMR algorithm

has developed many variants, which can be categorized into: cell-based [60], patch-based [11] and block-based AMR [57] algorithms. Although the fundamental idea behind these algorithms is roughly the same, there are marked distinctions between their implementations particularly in terms of the granularity. In general granularity is a measure of computation power to communication overheads [61]. In this work it specifically implies the required memory size and the complexity of the relevant algorithm. A coarser model granularity denotes a smaller memory size and a simpler AMR operation and vice versa.

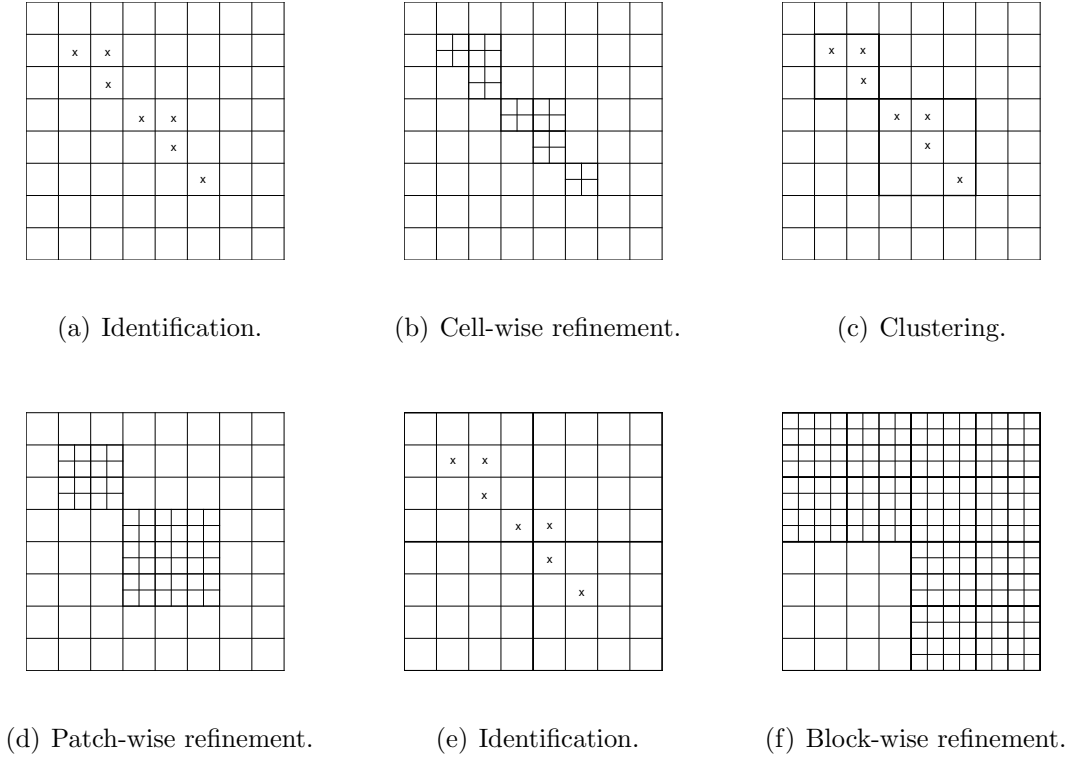


Figure 1.1: The refinement operation of: (a-b) cell-based AMR; (c-d) patch-based AMR; (e-f) block-based AMR, where the thin lines denote meshes while the bold lines denote block boundaries.

The conceptual overview is indicated by several sketches in Figure 1.1, where the refinement operation is displayed as an example. Firstly, in the cell-based AMR refinement operation, the identified cells are refined directly. It is easy to see that its granularity is fine-grained, *i.e.* a detailed data structure has to be maintained for each cell [60]. Secondly, in the patch-based AMR refinement operation, the identified cells are clustered together to be several big patches with a clustering operation [11]. The refinement operation is operated on a relatively high level, on which a simpler data

structure is maintained for each patch. Finally, in the block-based AMR refinement operation, a computational domain consists of blocks with a predefined number of cells, *e.g.* 4×4 cells in each block for the example. If any cell in one block is identified, the whole block is refined [57]. As a result the data structure is only maintained for blocks. Comparing to cell-based and patch-based algorithms, it shows that the granularity of block-based AMR is the coarsest.

In some works [27, 58] patch-based AMR algorithm was also regarded as a generalization of block-based AMR algorithm considering the similar organisation of data structures. In spite of that these two algorithms have been described here respectively in order to illustrate the differences between both AMR ideas more exactly.

It is easy to see that theoretically the highest efficiency and the most flexibility can be gained through a cell-based AMR algorithm. Nevertheless, its memory and communication cost is also the highest and its fine-grained algorithm is error-prone and difficult to implement. By contrast, in patch-based AMR algorithm some of irregularities in the algorithm have been removed by using a clustering operation to organize the identified cells in patches. The required cost is reduced accordingly. Block-based AMR algorithm is certainly the most homogeneous and its data structure and the corresponding algorithm are the simplest. It allows the most feasible parallel implementation while still increases computational efficiency by reducing the required cells count. The price to pay for the simplicity is the loss of some efficiencies comparing to cell-based and patch-based AMR algorithms [57].

Although the mentioned block-based AMR algorithm is conservative and includes a tradeoff between programming complexity and computational cost, it has been found that the efficiency of block-based AMR is still quite satisfactory in solving practical applications [62, 63]. Therefore, in this work the algorithm of block-based AMR is employed for its simplicity, which allows the main effort to be focused on implementing CAA schemes and studying practical CAA problems. The most part of the following survey is exclusively about block-based AMR algorithm, unless otherwise stated.

Fundamental ideas of AMR data structures and the corresponding implementations are scattered in several publications [26, 57, 64, 65]. No matter for cell-based, for patch-based or for block-based AMR, data structures can be implemented in the form of a tree [66], *i.e.* a binary tree for one-dimensional, a quadtree for two-dimensional and an octree for three-dimensional problems [67, 68, 69]. Two approaches have been developed. The first approach used traditional programming methodology, where the AMR data structure was described with the structure of C or was organized within modules of Fortran [57]. The other approach employed object-oriented design

methodology [70] because the highly object-oriented idea of AMR could be fairly represented with Java [29] or C++ [71, 72]. It was especially popular in the implementation of parallel AMR [26, 65]. Specific design details within object-oriented models could be found in [73] where the data structures and the related operations of AMR had been encapsulated in several classes. To combine with numerical computation efficiently, it has been implied that a mixed-language model could be a better choice to hide the details of AMR from numerical parts [74] using C++ to manage higher-level data structures while using Fortran for mathematic calculations. There were also some efforts trying to implement AMR with a parallel language [75], such as using the recently emerging High Performance Fortran [76]. In this work, although the relatively advanced object-oriented design methodology attracts more interest, the traditional approach is followed in order to be consistent with the previous work of the group.

Other than the mentioned two approaches for designing AMR data structure, there are several issues of AMR that have been studied by many researchers holding with three different point of views. The details are discussed below.

Firstly, from the perspective of high performance computing, the parallel implementation of AMR algorithm leads to a significant challenge on distributed memory machines. The dynamic nature of AMR operations, *e.g.* parallel mesh construction and runtime management, require more extensive and flexible memory operations. The whole computational domain of solutions has to be decomposed periodically amongst parallel processors to achieve load balancing dynamically [77, 78]. Fundamentally it is a problem of distributing AMR data structures evenly to parallel machines [57, 79]. One algorithm was presented to make load distribution evenly using grid-splitting and direct grid movements [80]. In the other algorithm, a suit of space-filling curves were employed to form an application-centric partitioning method that could select a proper partitioning strategy at runtime depending on the running application and system states [81]. Both algorithms are problem-specific and can not be applied to this work directly.

In addition, the improvement of the parallel AMR computational efficiency is also a major concern. Several algorithm issues affecting parallel scaling have been discussed, where the particular attention was paid to the cost and efficiency of communication libraries [82]. It was proved the AMR method behaved well when scaling up to thousands of processors [83]. Nevertheless, the test case employed was idealised. The parallel performance for practical applications is still problem-specific and relies heavily on the code optimization. In this work the parallel algorithm is to be designed specifically for simplifying the code, which parallel performance associated with CAA

applications is also to be discussed.

Secondly, from the perspective of numerical analysis, the correct treatment around fine-coarse interfaces associated with an adaptively refined mesh has been actively studied for both the finite volume [11, 44, 84] and finite difference methods [85, 86].

Within finite volume methodology, a flow solver consisting of a linear reconstruction method and an approximate Riemann solver was described for applications on an adaptively refined mesh [66]. Another numerical scheme was described to model chemical reaction over overlapped adaptive meshes using 2^{nd} -order Godunov method for convective fluxes and working in a predictor-corrector fashion [87]. Some efforts have also been done combining AMR with the multigrid method [88]. To keep the divergence-free property of some physical problems, *e.g.* magnetohydrodynamics, several methods were presented in [89] and references therein. Special treatments including revised interpolations and flux updating between fine and coarse grids were employed to enforce conservation laws at fine-coarse interfaces between blocks of different refinement levels [57].

Within finite difference methodology, a 2^{nd} -order centered spatial differential scheme was employed to solve a wave propagation problem [85]. The numerical reflection coefficient with regard to the wavenumber at fine-coarse interfaces was studied. It was indicated that wave with smaller wavelength had higher reflection coefficient, *i.e.* high wavenumber part was more susceptible to be unstable. Short wavelength spurious reflection wave was discovered as wave propagated across fine-coarse interfaces with a 2^{nd} -order linear interpolation method. To mitigate the problem, a 3^{rd} -order quadratic interpolation method was used around a fine-coarse interface. Conservative smoothing method was also used in the other work [86] to allow the Lax-Wendroff scheme freeing of spurious oscillations around interfaces.

Although both finite volume and finite difference methodologies have been employed successfully on an adaptively refined mesh, the latter is preferred herein in that it is relatively easy to handle.

Thirdly, from the perspective of grid generation, AMR algorithm has potential to generate a structured hierarchical mesh around solid boundaries [60, 66, 90] working with an immersed boundary method [91]. The number of grids surrounding a solid boundary is increased to simulate the immersed boundary to an expected exactness, while the number of grids in the region away from that boundary is reduced to save computation resource. Subsequently the tedious task of producing a body-fitted mesh can be saved [36, 92]. Several approaches have been presented to compute spatial differentials of grids near immersed boundaries [91], including: reshaping finite volumes near the body to a mosaic of body-fitted trapezoidal cells [62], imposing a

feedback forcing [36] or a direct forcing [93, 94] and using a ghost-cell (cut-cell finite-difference) method [94] to simulate the existence of boundaries.

However, the advantage of automatic grid generation with AMR was mainly manifested in solving the Euler equations due to its difficulty in solving problems with viscous boundary layers. One method approached the problem with an anisotropic mesh adaptation managing the Cartesian grid cells and faces with an unstructured data structure [92]. Nevertheless, the parallel implementation of this approach is difficult due to its highly inhomogeneous memory allocation.

Some others proposed using AMR on a body-fitted mesh [12, 62]. This approach, by refining and coarsening a body-fitted multi-block mesh, gives up much of the simplicity of the immersed boundary method and the elegance of the Cartesian grid method. By contrast, a method of hybrid mesh was presented using overlapping grids to solve viscous boundary layers on body-fitted grids and otherwise to solve flow fields on Cartesian grids [95]. The method has evolved to a huge framework, OVERTURE [96], including: grid generation, AMR, solvers, data base, graphical display and so forth. Other than reducing the required cell number besides boundaries, both methods also supported high-order spatial schemes straightforwardly [97, 98]. In this work the former method, body-fitted multi-block AMR, is employed for its simplicity of the code.

The above paragraphs summarise several design aspects associated with AMR. The attention here is restricted to solving CAA applications with AMR. Two specific issues appear.

Firstly, with respect to numerical issues, high-order finite difference schemes are generally preferred to ensure accurate performance [5]. Potential methods consist of two options: compact (implicit) and explicit schemes. A detailed survey of their numerical properties is to be given in the next section. The following discussion only concerns their potential usage on an adaptively refined mesh.

Normally, a compact scheme is more accurate than its explicit counterpart of the same order [99]. However, a special treatment is required around the interfaces between blocks for the proper working of a compact scheme across the whole domain. Several approaches have been presented. The first approach was based on a mesh with overlapped blocks, on which solutions of grids near one side of a block interface are exchanged with solutions on the other side after each computational step [9, 100]. The other approach designed a high-order explicit scheme specifically for the grids around block interfaces [101, 102]. For the reason that both approaches can not be applied on an adaptively refined mesh straightforwardly, compact scheme is not recommended in this work in spite of its numerical merit.

High-order explicit schemes are hold more attentions here. A primitive method of high-order mesh refinement multigrid computation with explicit schemes was given in [103]. Recently work was done at Sandia National Laboratories combining high-order methods and the AMR algorithm together for simulating reacting flow [18, 42, 63, 104]. An initial work investigated several centered spatial stencils (2^{nd} - and 4^{th} -order), the corresponding interpolations and their overall effect on the convergence order [104]. It was demonstrated that the order of the interpolation method employed in the computation should be higher than the order of the selected spatial scheme to keep the order of accuracy on an adaptively refined mesh. Up to 8^{th} -order spatial schemes were tested and the same result was still kept [18, 63]. A complete description of high-order discretization schemes, interpolation and filters for AMR has been given [42]. It was discovered that the computational efficiency of 4^{th} -order approaches was higher on an adaptively refined mesh comparing to 2^{nd} -order schemes [104]. More efficiency may be achieved if higher order spatial schemes were to be used. However, the corresponding high-order interpolation methods may become cumbersome and difficult to code. As a result of performing a tradeoff between computational efficiency and programming effort, 4^{th} -order schemes gains more interests in the work.

Other than the numerical issues, developing an AMR code with reasonable efforts poses a challenge. Several frameworks implementing AMR algorithm have been established, involving: Gerris [60] for cell-based AMR; AMRCLAW [105], BoxLib [106], DAGH [71] and SAMRAI [58] for patch-based AMR; and Chombo [74] and PARAMESH [57] for block-based AMR. These tools were generally designed for a particular research community, most in the area of computer science [27, 79, 107, 108]. When applying AMR to CAA applications, two disadvantages prevent the direct using of one established framework. First, a public accepted mature flexible AMR framework on a distributed memory machine is still not available. Second, those existing frameworks are always too big to be accessible and not convenient enough to accommodate a group of CAA schemes while the overall efficiency is still kept. Hence a simplified AMR framework is constructed in this work, although the code from some existing frameworks, especially PARAMESH, is extensively referred.

1.2.2 Computational Aeroacoustics

Generally speaking, CAA applications fall into three stages: sound generation, sound propagation and sound radiation. In this work the main concern is sound propagation. Some of the relevant computational methods are surveyed below. The technique used to predict far-field sound radiation is also discussed briefly at the end of this section.

High-order methods are generally preferred to ensure accurate performance when

simulating aeroacoustic propagation problems [5]. For finite difference schemes, either explicit or implicit schemes have been studied and employed. A commonly used explicit dispersion-relation-preserving (DRP) scheme was developed by Tam *et al*, in which the coefficients of the scheme were optimized to reduce the dispersion error [109]. The scheme was also extended to a predefined multi-size mesh [110]. In addition to the DRP scheme, standard explicit schemes whose coefficients were given by matching to the corresponding Taylor series could be used on an adaptively refined mesh too [42]. One of these schemes was employed in solving a problem of duct propagation [97] for the reason that its single-side stencil was simpler to implement than the DRP counterpart was.

Other than the relatively simple explicit schemes, implicit type high-order compact schemes are studied and employed in CAA applications as well. A complete description of compact finite difference schemes with spectral-like resolution was given in [99]. Such schemes allow the amplitude and phase of radiating waves to be accurately determined using a smaller points per wavelength grid resolution, yielding a marked improvement in accuracy and efficiency compared to the explicit counterparts. However, a tridiagonal matrix has to be solved to apply the method. To mitigate the difficulty, a prefactored compact scheme was presented in [101], where the tridiagonal matrix was partitioned to a lower triangular and an upper triangular matrix that could be solved more efficiently. To minimize dispersion errors an optimization process was introduced in prefactored compact scheme adjusting its coefficients [102].

With respect to temporal integration method, a multi-step low dispersion Adams-Bashforth method has been employed successfully in some CAA applications [110]. However, multi-stage Runge-Kutta methods gain more preference mainly due to their simplicity. A low-dissipation and low-dispersion Runge-Kutta method was developed particularly for CAA applications [111]. In case development stage the choice between multi-step and multi-stage method depends to a large extent on memory allocation costs and implementation efforts. Generally the former Adams-Bashforth method requires more memory costs and design efforts than the latter Runge-Kutta method. Moreover, Runge-Kutta method has been proved to behave well with both the compact and DRP schemes and it is self-starting. Hence a low-dissipation and low-dispersion Runge-Kutta method is to be employed in applications throughout this work.

A high-order method does provide the attractive properties of less dissipation and less dispersion than its low-order counterpart. However, it is more susceptible to numerical errors [85]. A range of methods, including: an explicit filter [101], a group of implicit filters [9, 112] and an artificial selective damping method [109, 110], were used

in CAA applications to remove the accompanied high frequency numerical nuisance. Generally speaking, an explicit filter is easy to implement. However its performance is worse than an implicit filter of the same order. Meanwhile, artificial selective damping method differs a little from its predecessor: artificial dissipation model [113, 114]. It was optimized to eliminate unresolved high wavenumber components as well as to keep resolved parts [115]. In a computation with a multi-stage temporal integration method, filtering should only be employed after the whole step integration is finished [42]. By contrast, artificial selective damping can be used in each stage.

It was found that sound generated by aerodynamic flows could be described by a wave equation that was an exact rearrangement of the Euler or the Navier-Stokes equations [116, 117]. This so-called acoustic analog theory allows solving sound source mechanisms and sound propagation separately. One common form of the acoustic analogy is Ffowcs Williams–Hawkins (FW–H) equation [118]. Therefore, once a sound solution of the near-field is obtained the corresponding far-field directivity can be estimated via an integral surface solution of FW–H equation. The details relevant to the numerical implementation of FW–H equation can be found in [119].

1.2.3 Sound Radiation from Ducts

In order to learn more about the advantages of the AMR method for CAA applications and about techniques could be used to mitigate the possible disadvantages, several aeroacoustic problems are to be solved in this work. The propagation and radiation of discrete frequency tones generated by fan rotor-stator interactions in aircraft engine ducts are of particular concern in the work. The source model is assumed to be independent of the propagation and radiation so that it is usually considered to be a known input. It has been described in the classical work of Tyler and Sofrin [120], where an analytical relationship determined the sound frequency and the circumferential mode number in terms of the blade passing frequency and the number of rotor blades and stator blades is presented.

The majority of research applying theoretical analysis to acoustic radiation from turbofan engine has modelled the engine duct as a straight duct. A relative simple asymptotic equation simulated a high frequency sound radiation out of a jet pipe based on Kirchhoff approximation [121]. As a benchmark problem, it is used in the work to testing against numerical solutions solved on an adaptively refined mesh. More details of the asymptotic equation are presented in Appendix B.

In practical numerical computations, the duct is generally simplified to be axisymmetric. The source is decomposed to separate components with different pair of circumferential mode and frequency. For the reason that the magnitude order of

sound is much smaller than the mean flow field in the duct, each component of sound can be described by a group of two-dimensional linearised Euler equations (LEE) [122], which contain terms with complex number. To avoid the operation of complex number and reduce required memory, a new variable: the temporal difference of the circumferential sound velocity, was introduced to reorganize the original governing equations to so-called two-and-a-half dimensional linearised Euler equations (2.5D LEE) without complex terms [7, 123]. The method has been applied to generic aero-engine intake duct [8, 124, 125] and exhaust duct [126] problems and proved its efficiency and convenience.

1.3 Thesis Structure

A code applying the simplified AMR algorithm to CAA applications is developed as a part of this PhD work. It consists three parts: AMR library, CAA library and applications code. The AMR library defines the overall data structures and provides subroutines to support AMR operations. The high-order computational solver, including spatial and temporal schemes, explicit filters and artificial selective damping, are organized together in the CAA library. Applications code contains several CAA benchmark problems to verify the AMR and CAA libraries.

The rest of the paper is organized as follows. Firstly in Chapter 2, various AMR algorithms and their essential ideas are introduced, followed by the details of the algorithm employed in this work. After that, in Chapter 3 several numerical issues associated with the AMR method are addressed. In Chapter 4 preliminary results of some benchmark problems are presented, where a problem of wave propagation demonstrates the working of the distributed parallel AMR framework, a benchmark problem of sound scattering from a cylinder shows the working of the immersed boundary method and a benchmark problem of a spinning modal sound radiation out of an unflanged duct is computed and tested against the asymptotic solution [121], to show the accuracy and efficiency of the AMR method. After that, in Chapters 5–6 the approach developed in this work is applied to practical problems of sound radiation from an aero-engine intake duct and exhaust duct respectively. A summary of the overall PhD work on AMR and recommendations for future research are made in Chapter 7. Finally, a number of appendices have been written to provide the relevant information at the end of this work, including: the coefficients of the employed schemes; several demonstration codes; and a Fourier pseudospectral time-domain method applying to some CAA benchmark problems.

Chapter 2

The AMR Algorithm

This chapter presents the fundamental features of AMR. The first section gives an overall description of AMR by introducing its basic principles. The second section outlines the block-based AMR algorithm that is employed in this work. It then discusses its parallel implementation in more detail. At the end, the algorithm complexity of the block-based AMR method is analysed qualitatively.

2.1 Introduction

AMR improves the storage and computational costs by refining cells only in places where a high grid resolution is desired as well as coarsening cells in places with an unnecessary high resolution. The fundamental algorithm has been presented for more than twenty years and developed to various algorithms. It was firstly presented in the form of patch-based AMR to solve the Euler equations, where the Berger and Oliger algorithm operated on each cell of the relevant computational domain [11]. The basic procedure of the Berger and Oliger algorithm, from initiating the computation on a rectangular coarse mesh, is:

- estimating local truncation errors at all grid points with either Richardson's extrapolation [11] or using a problem specific criterion to identify areas with excessive and unnecessary resolutions;
- organizing these areas into rectangular patches by using a clustering algorithm to set up connection information and improve communication efficiency;
- regriding (refining and coarsening) these clustered patches by superimposing or removing sub-grids to accommodate changes in flow dynamics.

The procedure is operated recursively until either a given regriding level is reached or a predefined local truncation error level has been met.

```
//Pseudo code for regridding operation.
for(cell in the domain){
  if(RefineFlag(cell)==TRUE)
    Refine(cell);                                //Refine the cell.
  elseif(CoarseFlag(cell)==TRUE){ //Need coarsen operation?
    if(CoarseFlag(cell->siblings)==TRUE) //Siblings status.
      Coarse(cell); }}                      //Coarse the cell.
```

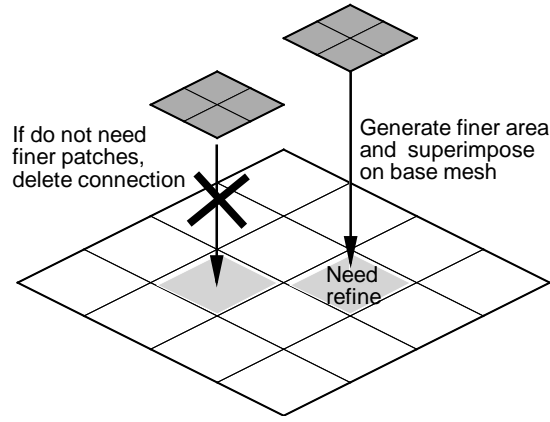


Figure 2.1: Regridding operation of AMR.

The regridding operation is illustrated in Figure 2.1. After this operation, the computational domain consists of a set of nested patches. Initial solutions on the newly generated patches are inherited from the base mesh as illustrated in Figure 2.2. This operation is referred to as prolongation. Conversely, solutions on the sub-grids update solutions of the corresponding base grids to keep accuracy consistently between refinement levees. This is known as restriction. It is easy to see that both prolongation and restriction operations in AMR are similar to the corresponding operations in multigrid methods [127].

In short, the Berger and Oliger algorithm contains: clustering, regridding, prolong and restriction operations. It is quite flexible: the superimposed rectangular sub-grid is allowed to rotated relative to the coordinates' axes or merged with other sub-grid to generate a bigger patch to save the relevant communication cost; and the refinement ratio between hierarchical levels can be adjusted, generally, but not limited by, from two to four [11]. It asks for extra efforts in the code implementation, especially on a parallel machine. To decrease the programming complexity, several simplifications were presented in the literature.

```
//Pseudo code for prolongation and restriction operations.
for (cell in the domain){
  if (after regridding operation){
    if(NewGenerated(cell)==TRUE) //Newly generated?
      Prolong(cell); } //Initialize its solutions.
  if (after each computing step){
    if(HaveParent(cell)==TRUE) //Have parent?
      Restriction(cell); }} //Update parent's solutions.
```

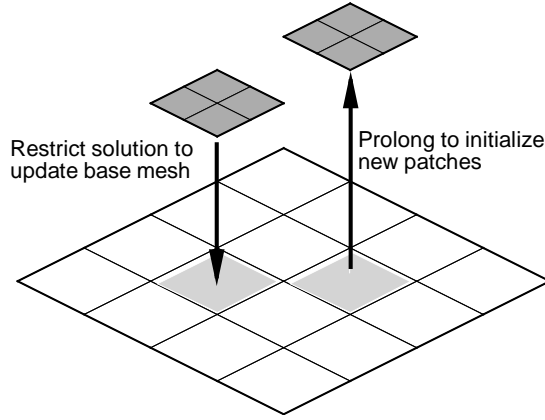


Figure 2.2: Prolongation and restriction operations of AMR.

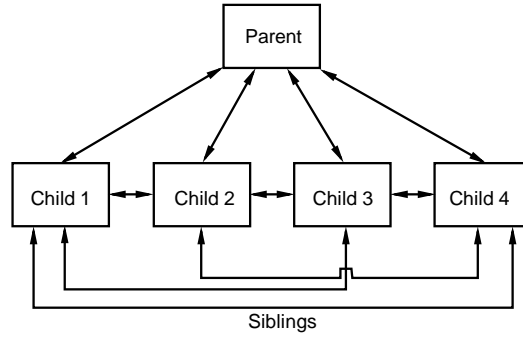


Figure 2.3: Quadtree represents the hierarchical relation of AMR.

The first simplification used a fixed refinement ratio of two by only bisecting required cells and maintained a quadtree data structure for two-dimensional hierarchical adaptive meshes [66]. A simple example of the data structure is displayed in Figure 2.3, where the relationships between refinement levels are named as parent and

children. Cells that have the same parent are described as siblings. The quadtree data structure stores the connection information and locates cells during computation. From any cells the whole tree can be accessed.

The second simplification employed block-based AMR method [57]. In block-based AMR method, the previously discussed AMR operations of the Berger and Oliger algorithm were operated on blocks that contained a predefined number of cells, *e.g.* $N_x \times N_y$ cells in a block for two-dimensional problems. There are some favorable properties of this method, *e.g.* tree-type data structure is maintained for regular hierarchical blocks only, parallel communications become simple and clustering operation is saved.

Both simplifications are employed in this work. A typical workflow involves the following steps:

- (1) setting up the computational parameters according to an input file and constructing the mesh with the finest grids;
- (2) regridding the computational domain according to a selected regridding criterion;
- (3) assigning solutions to the newly generated meshes with the AMR prolongation operation;
- (4) preparing solutions to compute spatial difference with the AMR ghost construction operation;
- (5) starting the general computation procedure;
- (6) updating solution on the coarse refinement level with the AMR restriction operation;
- (7) repeating steps (4) to (6) if necessary;
- (8) computing the regridding criterion according to the solutions on the adaptively refined mesh and going back to step (2).

All AMR operations have been discussed before except a new one, the AMR ghost construction operation, appearing in step (4). Its details are introduced in the next paragraph. Figure 2.4 shows these steps in a flow diagram.

An extra area surrounding each block is prepared in step (4) to solve partial differences of those cells located near a block boundary. That extra area was called ghost area or guard cells. Subsequently the operation of preparing solutions for the extra area was called ghost construction or guard cell filling interchangeably. To be consistent in this work, the names of ghost area and ghost construction are used. Figure 2.5 demonstrates the ghost construction operation on a vertex-centered mesh as an example. In this figure the deep gray area represents a ghost area,

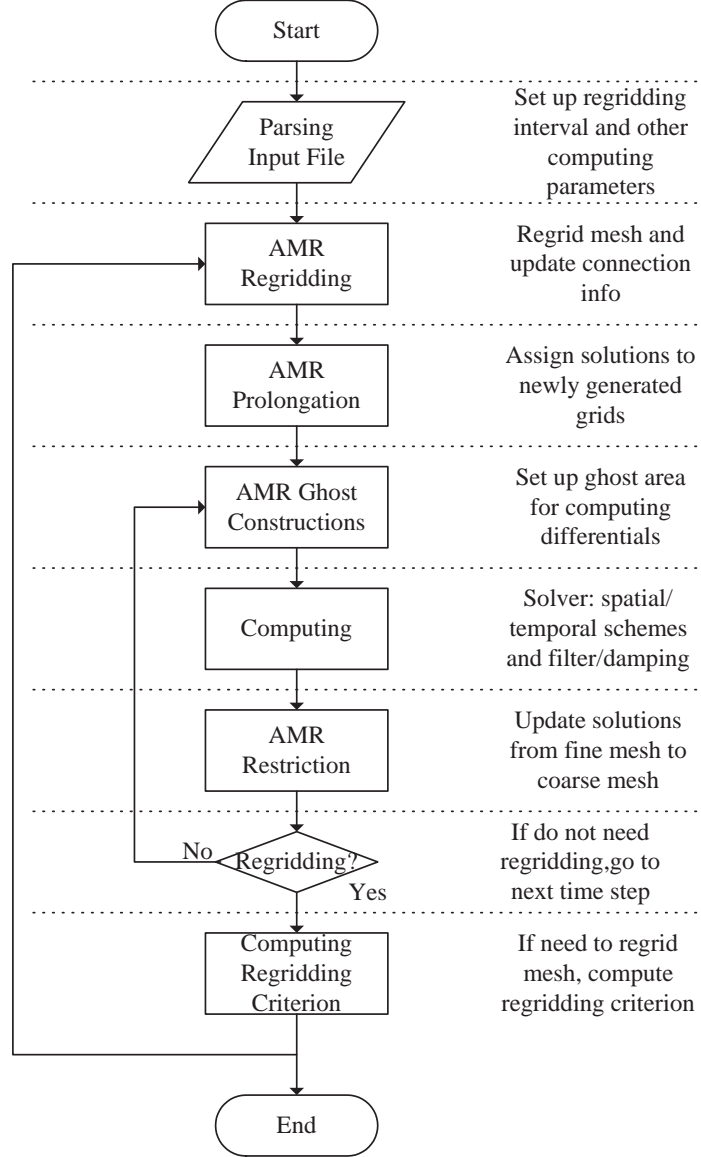


Figure 2.4: The flowchart of AMR.

where solutions are provided from a shallow gray area of the neighbouring block in two different ways depending on the refinement level difference. If the neighbouring blocks have the same refinement level, solutions are copied directly. Otherwise, either a restriction or an interpolation operation is executed before copying solutions.

In addition to the mentioned fundamental algorithm, several issues affecting the performance are considered in designing the code for this work. They are summarised below in more detail.

Firstly, a compromise between the computational load balancing and the communication efficiency is made to determine the number of cells, N_c (e.g. $N_x \times N_y$ in a

```

//Pseudo code for ghost construction operation.
//Copy_solution(src,dst)    //Protocol of the function.
for (block in the domain){
    neigh=neighbour(block);    //Get neighbour block id.
    if (level(block)==level(neigh))
        Copy_solution(neigh,block);
    elseif (level(block)<level(neigh)) //Block is coarser?
        Restriction(neigh);
        Copy_solution(neigh,block);
    elseif (level(block)>level(neigh)) //Block is finer?
        Interpolation(neigh);
        Copy_solution(neigh,block); }

```

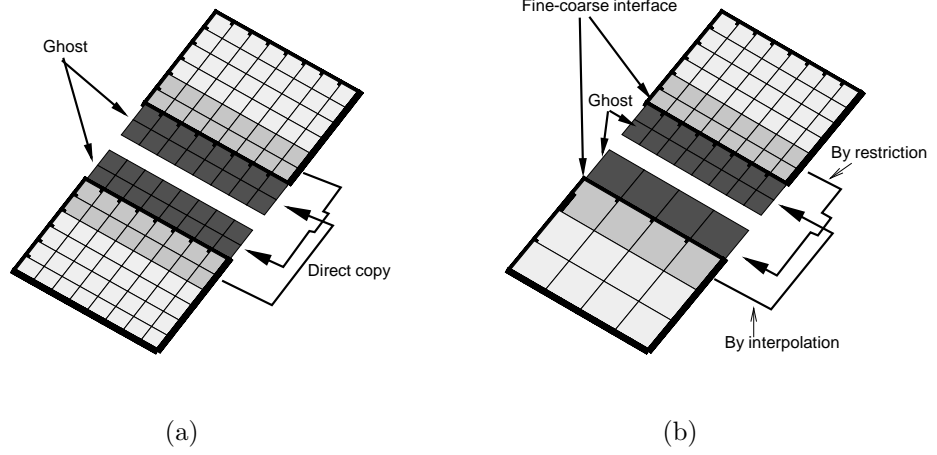


Figure 2.5: Ghost construction operation of AMR, where the neighbouring blocks have: (a) the same refinement level; (b) different refinement levels.

two-dimensional block), of each block. Generally, the difference of computational load between processors is proportional to the size of one block. A smaller N_c , therefore, gives a better balanced computational load amongst parallel processors. However, it also introduces more communications between processors according to several numerical experiments that have been done in this work. In addition, for the purpose of reducing the potential code complexity, every block in the computational domain have the same cell number. The value is problem specific. For some two-dimensional benchmark problems appearing in Chapter 4 each block contains 8×8 cells to achieve good load balancing. For other practical problems appearing in Chapters 5–6 each block contains 20×40 cells to simplify the task of grids generation.

Secondly, an appropriate time interval, T_{amr} , between the consecutive operations of regridding has to be set to assure the successful capturing of instantaneous physics on an adaptively refined mesh. In short, a small T_{amr} has to be used for applications with a rapid physical movement and vice versa.

Thirdly, the numerical scheme employed in CAA applications affects the width, W_g , of a ghost area: W_g is at least 3 for a 4th-order DRP scheme; at least 5 for a 4th-order prefactored compact scheme. The value of W_g also affects the value of T_{amr} . It was discovered that with a wider width (W_g), a bigger regridding interval (T_{amr}) could be used without losing capturing ongoing physics. An empirical relation of $T_{amr} \leq W_g \Delta x / (|v| \Delta t)$ based on several numerical experiments should be satisfied to solve wave propagation problems, where Δx is the size of spatial discretization, Δt the integration temporal step, v the propagation speed, $| |$ the amplitude. Otherwise the instantaneous physical phenomena may be lost on the finest level of blocks.

```
//Pseudo code for ghost construction operation.
//Copy_solution(src,dst)           //Copy solutions.
//Copy_difference(src,dst)         //Copy spatial differences.
for (block in the domain){
    Copy_solution(block,tmp_block);
    Ghost_construction(tmp_block); //Do ghost construction.
    Compute_difference(tmp_block); //Get d/dx.
    Copy_difference(tmp_block,block);}

```

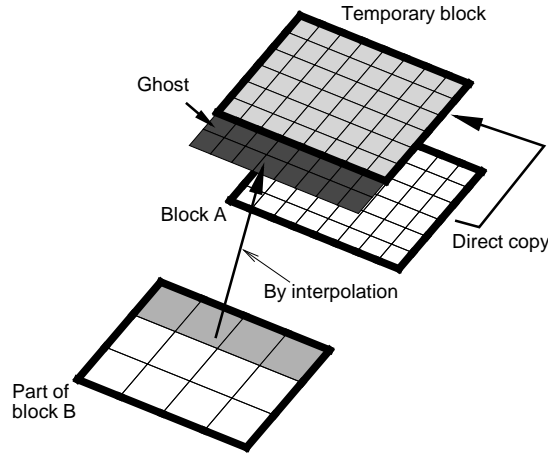


Figure 2.6: The compact memory model of the AMR ghost construction operation.

Finally, a compact memory model is chosen to reduce the storage burden at the cost of some computational efficiency. Actually, the underlying memory model of

the previous example (Figure 2.5) is not compact. For example, in two-dimensional cases its overall storage cost is proportional to $[N_{block} \times (N_x + W_g) \times (N_y + W_g)]$, where N_{block} is the block number in the domain. In contrast, the storage cost of the compact memory model, which is much smaller than the previous one, is proportional to $[N_{block} \times N_x \times N_y + (N_x + W_g) \times (N_y + W_g)]$. An extra temporary block (located at the top level in Figure 2.6) of size $[(N_x + W_g) \times (N_y + W_g)]$ is introduced to computing spatial differences. The whole procedure is indicated in the pseudocode listed in Figure 2.6. With this compact memory model, the storage cost is reduced whereas some extra computational costs managing memory movements between a smaller block and the bigger temporary block have to be incurred. It reflects a tradeoff between the storage cost and the computational cost. The quantitative analysis of the increased computational cost is performed in a benchmark case study that is given in the next chapter.

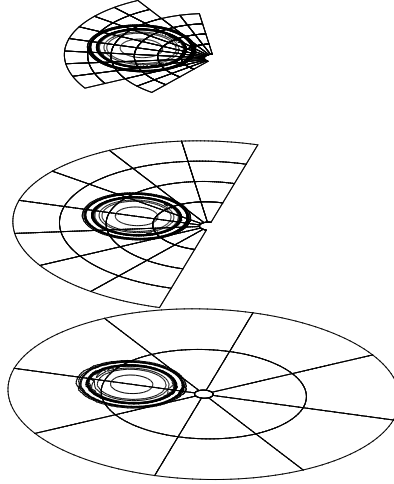


Figure 2.7: A CAA application of block-based AMR.

Figure 2.7 provides an overall view of the AMR method employed in this work. It shows a problem of sound scattering off a cylinder that is solved on a body-fitted multi-block mesh, where each block contains 20×20 cells. The overall mesh consisting of three refinement levels is created at the start of the computation. The refinement ratio between two consecutive coarse and fine levels is 2. The AMR re-gridding operation defines the relationships between blocks as parents/children or sibling according to the means they are connected. It stores the hierarchy information in the data structure of quadtree and refines and coarsens the hierarchy mesh based on the gradient of the velocity perturbation. As the simulation progresses, the mesh is dynamically updated to reflect the evolving physics. Meanwhile, the prolon-

gation operation provides initial solutions to newly generated blocks. The restriction operation updates solutions of coarse blocks. The diagram is displayed with the aim of showing the hierarchy of the adaptively refined mesh clearly. In reality fine levels are superimposed on coarse meshes directly. In addition, block boundaries are displayed whereas cells are not visible in order to display the figure clearly. The same figure style is followed throughout this work.

2.2 Parallel AMR

This section introduces the background information and the underlying algorithms for implementing the aforementioned body-fitted AMR on a parallel machine. Firstly it discusses the features of two parallel development methodologies based on shared memory and distributed memory models. The latter one is used in this work for its portability. It then introduces the operation that is dynamically balancing computational load. Several examples are given to explain the principle behind the operation. Finally, it introduces a simplified parallel communication part employed in this work. This section requires previous knowledge of the loosely coupled parallel programming applications [128] and the message passing interface (MPI) [129].

2.2.1 Parallel Methodologies

The distributed memory machines are increasingly adopted as a cost effective alternative to classical supercomputers for running large scale numerical simulations. Either shared memory or distributed memory model can be applied on these machines [130]. From a programmer's point of view, shared memory model is more desirable in that a memory location can be both read from and written to by multiple processors directly and transparently with this model. It allows programmers to focus on algorithm design rather than on managing tedious low-level memory communications. Prevalent machines offering this model, via particular combination of software and shared memory hardware, include the quite expensive SGI Onyx and IBM SMPs.

For economical distributed memory machines, *e.g.* Beowulf cluster [131], which are built directly with commercial off-the-shelf products such as personal computers connected by high-speed network, several software projects are developing to fulfill a distributed shared memory model [130]. Some most promising options, including the Global Arrays toolkit [28], Titanium [29] and Unified Parallel C [30], have demonstrated their functions by providing primitive examples of AMR applications. The first option, Global Arrays toolkit, helps programmers to access distributed global

arrays transparently as if they were residing in a shared memory. The other two options, Titanium and Unified Parallel C, extend Java and C languages respectively by adding features to supply a global address space for the underlying distributed parallel machines. These tools are promising because of their potential to ease programming effort extensively in developing AMR. However, these programs are as yet to be tested, not mature enough to match its pervasive counterpart, MPI that has manifested its success in numerous applications [132].

An application based on MPI generally has multiple threads on multiple address space, distinctively different from that on shared memory model with multiple threads on a single address space. In addition, although the parallel debugging is still quite difficult, an application based on MPI is more portable on both distributed and shared memory machines. MPI is therefore employed herein to implement the AMR algorithm with a distributed memory model.

Most established parallel AMR frameworks employed the block-based AMR algorithm [57, 62, 74, 106]. Following the same way, an AMR framework has been developed in this work. Other than the mentioned AMR operations, it consists of a dynamic load balancing operation to distribute computational load evenly amongst parallel processors [27, 108]. Details are described below.

2.2.2 Dynamic Load Balancing

Dynamic load balancing is the most complex part in the parallel implementation of the AMR method in terms of the required programming efforts. It has to be used after each regridding operation. The operation repartitions and reconstructs a tree data structure on each processor and transfers proper number of blocks from overloaded processors to underloaded processors. Different methodologies have been presented, which generally belong to two types: diffusion-based method and scratch-and-remap method [80, 133]. The diffusion-based method adjusts the computational loads between adjacent processors and the scratch-and-remap method repartitions and distributes workload globally. The latter method is adopted here giving the fact that the computational load changes dynamically throughout the whole computational domain.

Moreover, two algorithms have been developed for the scratch-and-remap method. The first algorithm contains several steps:

- (1) mapping the computational domain to a one-dimensional array by space-filling curves;
- (2) partitioning and migrating the array evenly amongst processors;

- (3) storing new memory addresses and reconstructing the tree data structure to complete the load balancing operation.

Figure 2.8 indicates the procedure of the load balancing with a simple two-dimensional test case solving on two processors. First of all, there are five blocks distributed within two processors. As soon as a regridding operation is finished, four new blocks (block 3–block 6) are generated over their parent block 2 leading to the load imbalance (Figure 2.8(a)). By using a space-filling curve, these blocks are ranked to form a one-dimensional array, in which load balancing can be done straightforwardly. It is easy to see that the final two blocks in the CPU 1 is required to move to CPU 2 to distribute the load more evenly (Figure 2.8(c)–2.8(d)). At the same time the quadtree data structure on each processor is updated accordingly (Figure 2.8(e)–2.8(f)). This algorithm is called space-filling load balancing in this work.

The other algorithm doing the same job is the so-called max-min load balancing that transfers computational load from the most overloaded processor to the most underloaded processor [80]. Theoretically max-min load balancing works more efficiently than space-filling load balancing does because its potential communication cost is minimal. For example, if the computational load shown in Figure 2.8 is redistributed amongst three processors, three migrations are required for max-min load balancing (Figure 2.9(b)), whilst five migrations are required for space-filling load balancing (Figure 2.9(c)).

However, max-min load balancing method has some drawbacks such as increased parallel communication cost in the ghost construction operation. The domain in Figure 2.8 is used as an example again to explain the problem. Two parallel communications are used for constructing ghost areas of block 3–block 6 using space-filling load balancing (Figure 2.9(c)). In contrast, four parallel communications are required for max-min load balancing due to the irregularly distribution of blocks (Figure 2.9(b)). This difference is also evident in the relevant data structures. The quadtree generated by space-filling load balancing is regular and well-ordered amongst processors, whereas the quadtree generated by max-min load balancing is quite messy (Figure 2.10). The same phenomena are also discovered for case studies with more blocks and processors. Space-filling load balancing, therefore, is preferred to improve the overall AMR communication efficiency for the ghost construction operation is operated more frequently than the dynamic load balancing operation does.

The construction of a space-filling curve in N -Dimensional ($N = 2, 3$) space appears to be the core part in space-filling load balancing. The construction procedure is shown in Figure 2.8(a), where a curve passes through the midpoint of each block in the solution domain. The curve is self-similar and recursive, maintaining locality

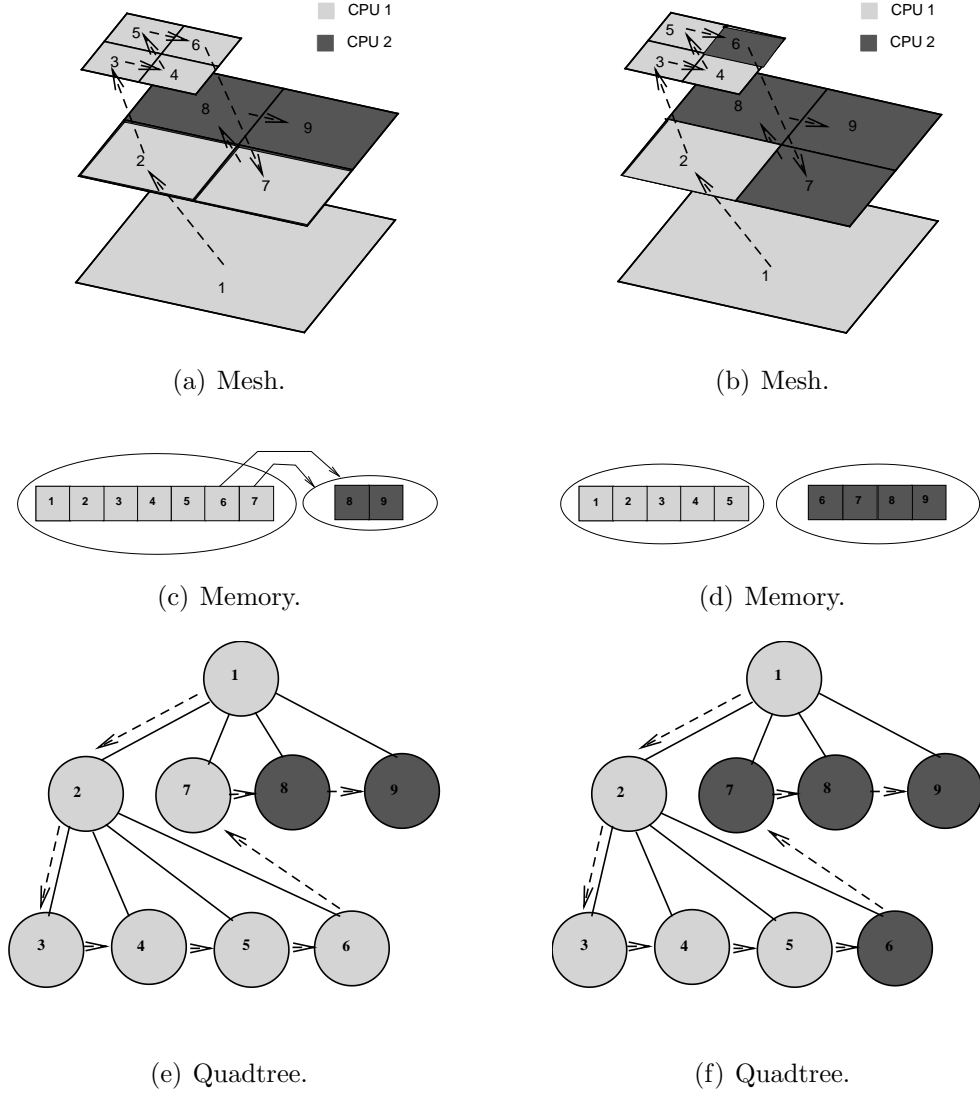


Figure 2.8: A dynamic load balancing example: (a)(c)(e) before balancing; (b)(d)(f) after balancing, where (\dashrightarrow) represents space-filling curves.

of the original domain. In other words, the neighbouring blocks on the original N -Dimensional ($N = 2, 3$) domain are still located closely in the relevant space-filling curve.

The curve index can be obtained by computing either Peano-Hilbert order (U-Order) [134] or Morton-order (N-order) [135]. In this work both computational methods have little difference in terms of computational efficiency. Morton-order is selected herein. Figure 2.8(b) has illustrated the working procedure of a Morton-order curve that maps a hierarchical two-dimensional domain to a one-dimensional array. The

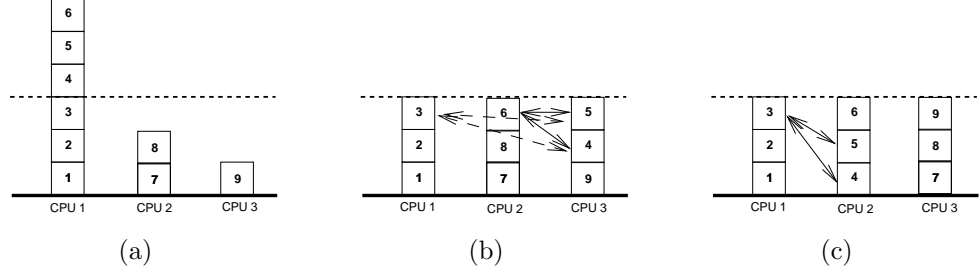


Figure 2.9: The procedure of load migration: (a) initial imbalance state; (b) balancing by max-min load balancing; (c) balancing by space-filling load balancing.

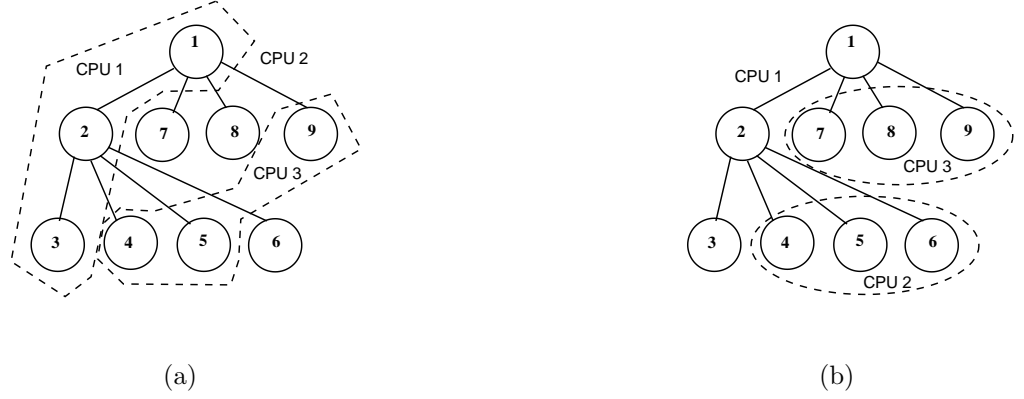


Figure 2.10: The reconstruction of the quadtree with: (a) max-min load balancing; (b) space-filling load balancing.

basic algorithm implementing this curve resembles a depth-first tree traversal, where the tree represents the relationship of blocks between refinement levels. The algorithm to do tree traversal can be found in many textbooks about programming data structures. Nevertheless, a mature parallel implementation of a tree data structure is not available. Hence, the parallel Morton-order in this work is set up by computing and sorting the Morton number of each block [136].

The algorithm computing the Morton number is summarised below. It works for two-dimensional cases, from which a three-dimensional algorithm also can be developed easily. First of all, these two definitions of even-dilated and odd-dilated representation are defined for an unsigned integer represented by binary numeral system, *i.e.* $i = \sum_{k=0}^{w-1} i_k 2^k$, where w is the number of bits. The even-dilated representation

is

$$\overrightarrow{i} = \sum_{k=0}^{w-1} i_k 4^k, \quad (2.1)$$

and the odd-dilated representation is

$$\overleftarrow{i} = 2 \overrightarrow{i}. \quad (2.2)$$

Consequently the Morton number of any integer index (m, n) is

$$M(m, n) = \overrightarrow{m-1} + \overleftarrow{n-1} + 1. \quad (2.3)$$

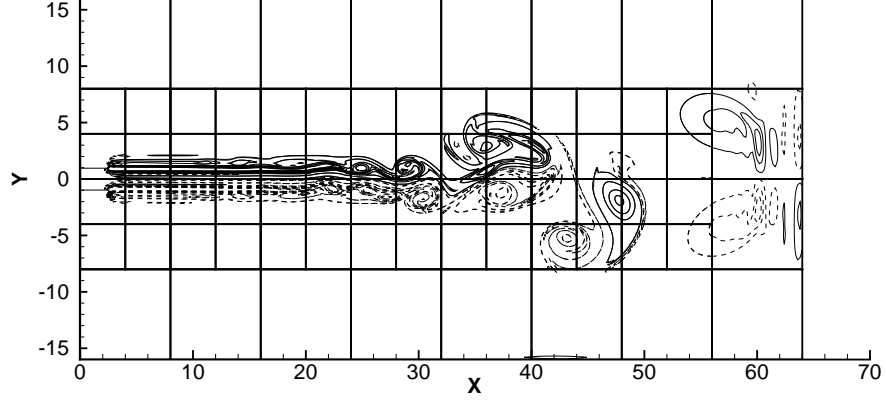
where (m, n) is the integer index of a block in two-dimensional domain. It applies to the Cartesian grids directly. After getting the Morton number of each block, the whole domain is mapped into a one-dimensional array orderly by using a quicksort algorithm [137]. A practical problem solved with the AMR method has been given in Figure 2.11 as an example, where a jet flow is developed from the left boundary and the mesh is subsequently adaptively refined with respect to a measure of the fluid vorticity and a space-filling curve is constructed as soon as the regridding operation is finished.

In this work, the method is revised a little to allow for multi-block body-fitted meshes. The following example of a mesh surrounding an engine intake illustrates the working procedure (Figure 2.12). In the first step, the physical domain in Figure 2.12(a) is transferred to square blocks, as shown in Figure 2.12(b). The connection relation is kept while the physical length is replaced with the dimensionless unit. In addition, the coordinate indices of child blocks relate to parent block only (Figure 2.12(b)). In the second step, the Morton number of each block is gained with Eq. (2.3) according to the corresponding coordinate indices. Finally, plus the Morton number of its parent, the overall Morton number of a block in an adaptively refined multi-block body-fitted mesh is obtained.

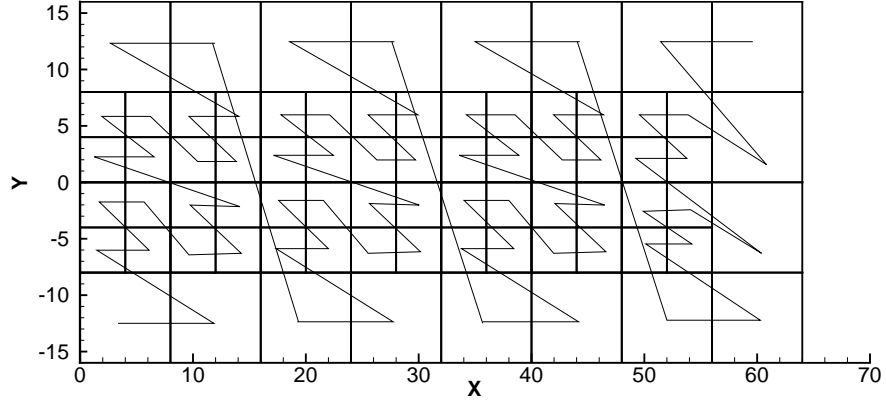
Once regridding and dynamic load balancing operations are finished, a tree data structure storing the relations between blocks enables the subsequent parallel communications associated with other AMR operations. It is discussed in the next section.

2.2.3 Parallel Communications

Figure 2.13 shows the various types of parallel communications of an AMR application on three processors, where unidirectional communications are operated for the restriction and prolongation operations and unidirectional/bidirectional communica-



(a)



(b)

Figure 2.11: A space-filling curve of a jet flow case solved with the AMR code: (a) vorticity contours; (b) space-filling curve.

tions are used for the ghost construction operation. For example, some blocks being computed on CPU 1 have neighbouring blocks located remotely on CPU 2 and vice versa. Therefore, bidirectional communications between them are triggered to construct ghost areas surrounding the blocks. In contrast, the child blocks on CPU 3 do not have neighbouring blocks holding with the same refinement levels on other processors. There is no need for these child blocks to provide ghost area information to any other processors. Hence unidirectional communications are enough to set up the ghost areas for blocks on CPU 3.

In addition to the various types of communications, different AMR operations also have different communication costs. For example, in Figure 2.13 prolongation

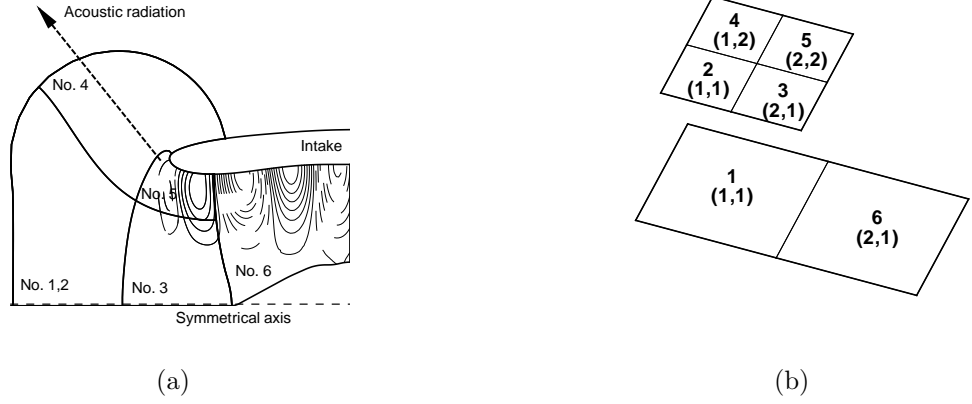


Figure 2.12: Computing Morton number for an engine intake case: (a) the computational domain with sound solutions; (b) the corresponding abstract blocks on which the top number is the Morton number and the bottom pair is the block coordinate indices.

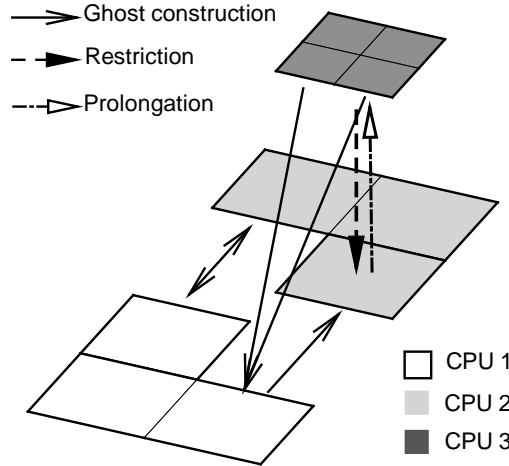


Figure 2.13: The parallel communication of AMR operations.

and restriction operations working on a block transport the whole block's solutions, whilst a ghost construction operation transports a part of the block's solutions.

In short, with respect to different AMR operations the corresponding parallel communications are different in terms of communication size and communication type (bidirectional or unidirectional). For that reason, the established AMR frameworks generally designed and optimized parallel communications specifically for each AMR

operation in order to maximise communication efficiency [57, 71]. However, these AMR frameworks implementing various parallel communication subroutines tended to be complex due to the existence of many logical operations and branching. In addition, in this work it has been found that the developing and debugging of the AMR communication code by using MPI are especially difficult and time-consuming on distributed memory machines. In order to make the development task of this work more manageable, the parallel communications associated with AMR operations are simplified. It is introduced below.

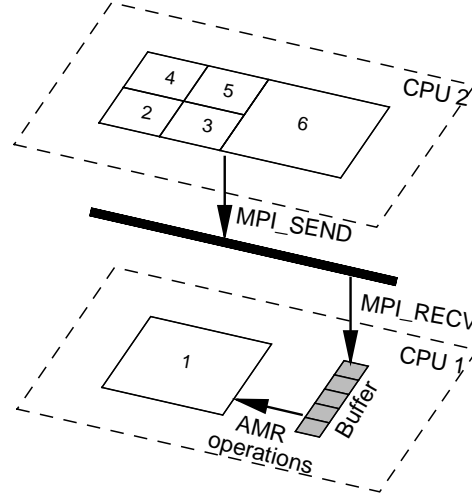


Figure 2.14: The simplified parallel communication of AMR operations.

To achieve a simple parallel implementation, the parallel communications of the prolongation, restriction and ghost construction operations are combined together. The case shown in Figure 2.12 is studied as an example again. For simplicity the blocks are assumed to be distributed on two processors (CPU 1 and CPU 2) as shown in Figure 2.14, where the computation on block 1 requires the restriction operation from block 2–block 5 and the ghost construction operation from block 6. Rather than issuing several communications for both AMR operations, the AMR code sets up a local buffer on CPU 1 to receive the solutions of blocks 2–6 on CPU 2. In other words, all solutions on one processor that are required by the other processor will be bundled together firstly and then will be send to a local buffer located on that remote processor.

The previous simplified procedure is called communication setup, or `Comm_Setup` in the code for brevity. The simplification does reduce logical options and help

to make code concise. It obviously incurs some overhead costs in communication and memory, what is called false sharing. The term has appeared in the area of memory and cache management of operating system for years [138]. In this work false sharing denotes the transfer of unnecessary data across the network. It happens where solutions of a whole block are transferred while some solutions are actually useless. To clarify the concerns over the efficiency penalties, the cost of each AMR operation is revealed by profiling the whole code in the next chapter. It justifies that the collective penalty introduced by the false sharing is affordable in solving problems included in this work.

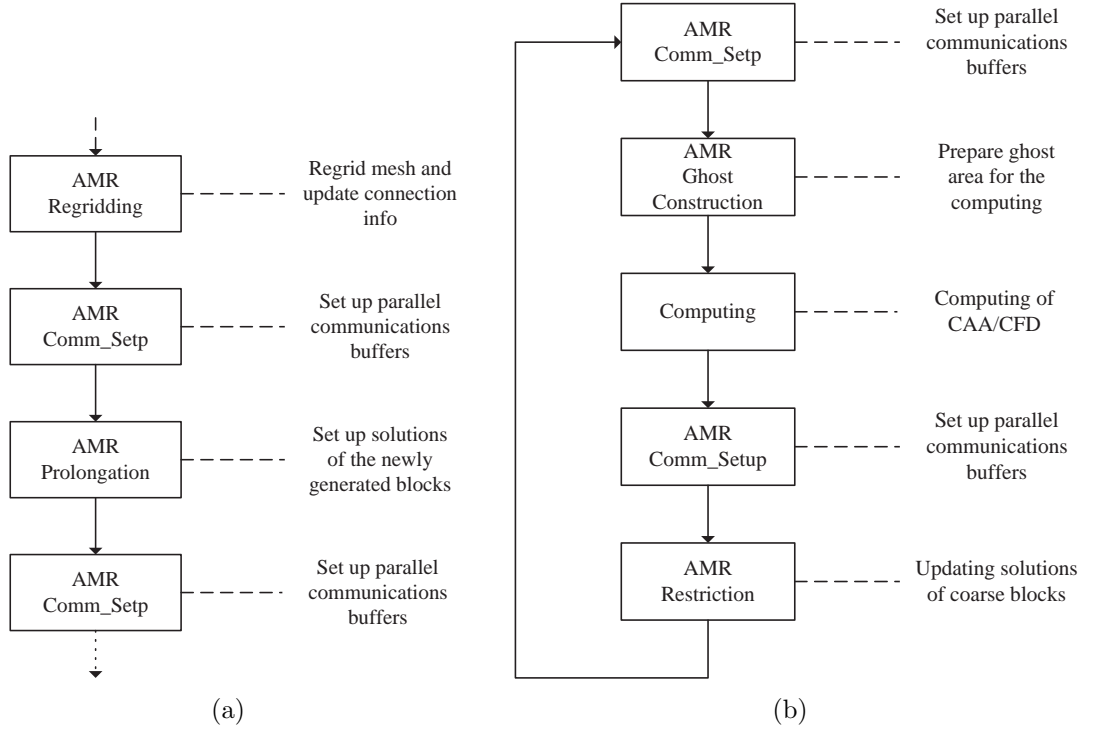


Figure 2.15: The flowchart of parallel AMR, where two parts correspond to: (a) steps (3)–(4) and (b) steps (5)–(8) of Figure 2.4.

Finally, it is worth emphasizing that when developing an application code based on the parallel AMR framework, the flowchart is slightly different from that shown in Figure 2.4, especially for the prolongation operation and the computing process. Parallel communications are required to set up local buffer correctly before and after some AMR operations. The main part of the flowchart is displayed in Figure 2.15.

2.3 Summary

Prior to applying the AMR code to CAA applications, there are some concerns raised over its required programming effort and the relevant computational cost. To address the first concern of the programming effort, the most complex elements of parallel AMR: dynamic load balancing and parallel communications, have been summarised in this chapter to demonstrate that the required programming workload is reasonable. In addition, several simplifications are tried to make the code complexity minimal and subsequently to reduce the required programming effort. To answer the second concern of the computational cost, both analysis and numerical experiment are to be operated. First of all, a brief qualitative analysis compares the cost of the AMR algorithm and the cost of the underlying CAA computation here. The AMR algorithm employs a tree data structure so that the computational cost of each AMR operation is $O(N_b \log(N_b))$, where N_b is the number of blocks and $O()$ is the O -notation that is usually used in algorithm analysis to represent the asymptotic upper bound of the computational cost. At the same time, the cost of a CAA computation in a single time step is generally $O(N_c)$, where N_c is the number of cells. Obviously, $N_b \ll N_c$. It is therefore clear that the cost introduced by the AMR algorithm is only a fractional part of CAA computational cost. This analytical conclusion is confirmed with a numerical experiment in Chapter 4, where an acoustics benchmark problem is solved to provide quantitative information.

In this chapter, Section 2.1 introduces the basic principles of the AMR algorithm and Section 2.2 discusses the parallel implementation method. Several different implementation methods have been mentioned and compared. The implementation method employed in this work is selected according to the principles of simplicity and portability. Its fundamental parts are described, whereas other parts shared the same features as established AMR methods are either omitted or only introduced briefly.

Chapter 3

Numerical Issues

This chapter provides underlying numerical issues associated with this work. The governing equations employed throughout this work are introduced firstly. A particular attention is then paid to characteristics of high-order schemes at a fine-coarse interface on an adaptively refined mesh. Several aspects such as stencils, stability and accuracy are discussed below in more detail.

3.1 Governing Equations

Several governing equations are used to verify the AMR code and the employed CAA schemes and to solve practical CAA problems, where the problem of a spinning mode sound radiation from a duct is of particular interest. These equations are summarised in the following sections in the form of Cartesian and cylindrical coordinate system formulations respectively. All variables appeared in the equations throughout this work are nondimensionalised using a reference length L^* , a reference sound speed a^* , a reference pressure $\rho^* a^{*2}$ and a reference density ρ^* . For the numerical examples presented in the paper, these have been taken as 1 m, 340 m/s, 141610 N/m² and 1.225 kg/m³, respectively.

3.1.1 One-dimensional Advection Equation

The one-dimensional advection equation is,

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0, \quad (3.1)$$

which describes the advection of a scalar $u(x, t)$ with a nondimensional uniform speed, where x is the Cartesian coordinate, t is time. It is employed to assess the effect of

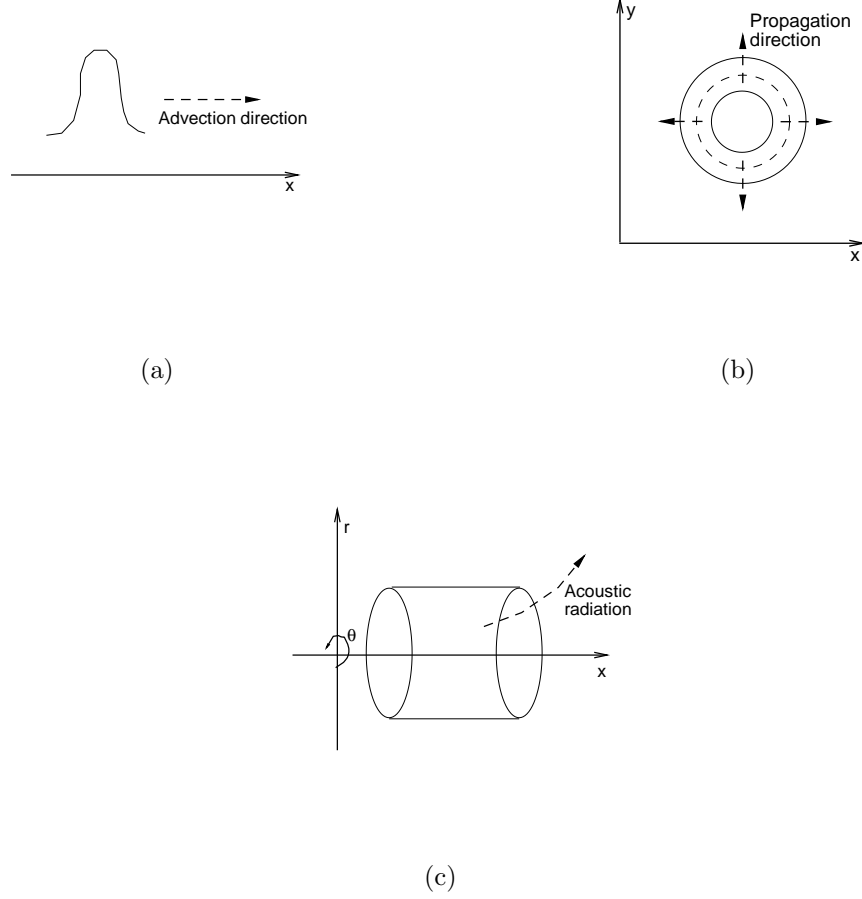


Figure 3.1: Benchmark applications: (a) advection; (b) wave propagation; (c) duct acoustic radiation.

interpolation methods and the performance of spatial discretization stencils associated with AMR mainly because of its simplicity for a rapid testing. A diagram in Figure 3.1(a) indicates the case.

3.1.2 Two-dimensional Wave Equations

The two-dimensional wave equations describe an initial Gaussian pulse propagating in a stationary medium, as illustrated in Figure 3.1(b). It is mainly employed to verify the AMR code herein. For the reason that the case problem is symmetrical in either coordinate direction it is convenient to rapidly find out potential bugs, especially those in the ghost construction operation. In addition, it is helpful to studying the

effect of the CAA schemes employed in this work.

The equations are:

$$\begin{aligned}\frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} &= 0, \\ \frac{\partial v}{\partial t} + \frac{\partial p}{\partial y} &= 0, \\ \frac{\partial p}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0,\end{aligned}\tag{3.2}$$

where x and y are the Cartesian coordinates, t is time, u and v are velocity perturbation in the x and y directions respectively and p is pressure perturbation.

3.1.3 Spinning Mode Radiation Equations

The problem of a spinning mode radiation of a duct is of particular interest in this work. A diagram in Figure 3.1(c) shows the basic problem. The case is also extended to simulate acoustic radiations of an aircraft engine intake and exhaust duct. The relevant discussion is given in the later chapters.

In order to simplify the problem with a subsonic mean flow, viscous diffusion, viscous dissipation and heat conduction are neglected. Hence the compressible Euler equations in cylindrical coordinates are used to model fluids around an axisymmetric duct, written in the conservative form as follows:

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \left(\frac{\partial}{\partial r} + \frac{1}{r}\right)(\rho v) + \frac{1}{r} \frac{\partial(\rho w)}{\partial \theta} &= 0, \\ \frac{\partial \rho u}{\partial t} + \frac{\partial(\rho u^2)}{\partial x} + \left(\frac{\partial}{\partial r} + \frac{1}{r}\right)(\rho uv) + \frac{1}{r} \frac{\partial(\rho uw)}{\partial \theta} + \frac{\partial p}{\partial x} &= 0, \\ \frac{\partial \rho v}{\partial t} + \frac{\partial(\rho uv)}{\partial x} + \left(\frac{\partial}{\partial r} + \frac{1}{r}\right)(\rho v^2) + \frac{1}{r} \frac{\partial(\rho vw)}{\partial \theta} + \frac{\partial p}{\partial r} &= 0, \\ \frac{\partial \rho w}{\partial t} + \frac{\partial(\rho uw)}{\partial x} + \left(\frac{\partial}{\partial r} + \frac{1}{r}\right)(\rho vw) + \frac{1}{r} \frac{\partial(\rho w^2)}{\partial \theta} + \frac{1}{r} \frac{\partial p}{\partial \theta} &= 0, \\ \frac{\partial e}{\partial t} + \frac{\partial((e+p)u)}{\partial x} + \left(\frac{\partial}{\partial r} + \frac{1}{r}\right)((e+p)v) + \frac{1}{r} \frac{\partial((e+p)w)}{\partial \theta} &= 0,\end{aligned}\tag{3.3}$$

where ρ is the density, p the pressure, u the axial velocity, v the radial velocity, w the azimuthal velocity, e the energy, defined by $e = p/(\gamma - 1) + \rho(u^2 + v^2 + w^2)/2$, x and r are axial and radial coordinates and θ is the azimuthal angle. These equations are taken to solve sound propagation from ducts with an axisymmetric mean flow field.

Assuming small perturbations are about a steady mean flow, acoustic wave propagation can be described by the LEE that are given below. To be concise, the variables

in this paragraph are represented in the form of vectors. For example, the velocity \mathbf{u} denotes (u, v, w) , satisfying $\mathbf{u} = \mathbf{u}_0 + \mathbf{u}'$, where $\mathbf{u}_0 = (u_0, v_0, w_0)$ is the mean flow velocity, $\mathbf{u}' = (u', v', w')$ is the velocity perturbation. Subsequently, the momentum equations of the LEE can be described in the vector form of:

$$\frac{\partial \mathbf{u}'}{\partial t} + (\mathbf{u}_0 \cdot \nabla) \mathbf{u}' + (\mathbf{u}' \cdot \nabla) \mathbf{u}_0 + \frac{\rho'}{\rho_0} (\mathbf{u}_0 \cdot \nabla) \mathbf{u}_0 = -\frac{1}{\rho_0} \nabla p'. \quad (3.4)$$

The preceding three-dimensional equations are still too expensive to solve. For that reason a further simplified assumption is taken. If the acoustic disturbances are restricted to the multiples of the blade passing frequency and propagate on an axisymmetric mean flow field without swirl, it is possible to write the disturbances in terms of a Fourier series. For example, the series of the pressure disturbance p' at a single frequency k is:

$$p' = \sum_{m=0}^{\infty} p'_m(x, r) e^{i(kt - m\theta)}, \quad (3.5)$$

where x is the axial coordinate, r the radial coordinate, t is time, m is the circumferential mode and θ the circumferential angle. Consequently, there are two important relations for the circumferential velocity disturbance w' and the pressure disturbance p' correspondingly. They are:

$$\frac{\partial w'}{\partial \theta} = -\frac{m}{k} \frac{\partial w'}{\partial t}, \quad \frac{\partial^2 p'}{\partial t \partial \theta} = m k p'. \quad (3.6)$$

By using Eq. (3.6) the general LEE in the cylindrical coordinates can be simplified to a set of two-dimensional equations that were generally called 2.5D LEE [7]. For convenience it is also called LEE in this work. The complete governing equations in the cylindrical coordinates for a single blade passing frequency k are:

$$\begin{aligned} \frac{\partial \rho'}{\partial t} + \frac{\partial(\rho' u_0 + \rho_0 u')}{\partial x} + \frac{\partial(\rho' v_0 + \rho_0 v')}{\partial r} - \frac{m \rho_0}{k r} w'_t + \frac{\rho' v_0 + \rho_0 v'}{r} &= 0, \\ \frac{\partial u'}{\partial t} + u_0 \frac{\partial u'}{\partial x} + v_0 \frac{\partial u'}{\partial r} + (u' + \frac{\rho'}{\rho_0} u_0) \frac{\partial u_0}{\partial x} + (v' + \frac{\rho'}{\rho_0} v_0) \frac{\partial u_0}{\partial r} + \frac{\partial p'}{\rho_0 \partial x} &= 0, \\ \frac{\partial v'}{\partial t} + u_0 \frac{\partial v'}{\partial x} + v_0 \frac{\partial v'}{\partial r} + (u' + \frac{\rho'}{\rho_0} u_0) \frac{\partial v_0}{\partial x} + (v' + \frac{\rho'}{\rho_0} v_0) \frac{\partial v_0}{\partial r} + \frac{\partial p'}{\rho_0 \partial r} &= 0, \\ \frac{\partial w'_t}{\partial t} + u_0 \frac{\partial w'_t}{\partial x} + v_0 \frac{\partial w'_t}{\partial r} + \frac{m k}{\rho_0 r} p' + \frac{w'_t v_0}{r} &= 0, \end{aligned} \quad (3.7)$$

where superscript $(')$ and subscript (0) denote perturbation and mean properties respectively; u' and v' are velocity perturbations in the x and r directions respectively; $w'_t = \partial w' / \partial t$. The fluid is modelled as a perfect gas. $p' = C_0^2 \rho'$, where C_0 is sound

speed. The boundary treatment for w'_t is the same as that for w' .

The incident wave is defined as follows:

$$\begin{aligned}
\rho'(x, r, \theta, t) &= a[J_m(k_r r) + c_1 Y_m(k_r r)] \cos(kt - k_a x - m\theta), \\
u'(x, r, \theta, t) &= \frac{k_a}{k - k_a M_j} p', \\
v'(x, r, \theta, t) &= -\frac{a}{k - k_a M_j} \frac{d[J_m(k_r r) + c_1 Y_m(k_r r)]}{dr} \sin(kt - k_a x - m\theta), \\
w'_t(x, r, \theta, t) &= -\frac{amk[J_m(k_r r) + c_1 Y_m(k_r r)]}{r(k - k_a M_j)} \sin(kt - k_a x - m\theta), \\
w'(x, r, \theta, t) &= \frac{am}{r(k - k_a M_j)} [J_m(k_r r) + c_1 Y_m(k_r r)] \cos(kt - k_a x - m\theta), \\
p'(x, r, \theta, t) &= a[J_m(k_r r) + c_1 Y_m(k_r r)] \cos(kt - k_a x - m\theta),
\end{aligned} \tag{3.8}$$

where M_j is nondimensional velocity inside the duct; a is fixed at 10^{-4} to ensure small relative changes in density (as required for LEE); J_m is the m^{th} -order of first kind Bessel function; Y_m is the m^{th} -order of second kind Bessel function. The n^{th} radial wavenumber k_r is the n^{th} solution of the following equation determined by the hard-wall boundary conditions of the duct

$$\frac{d[J_m(y_{outer} k_r)]}{dr} \frac{d[Y_m(y_{inner} k_r)]}{dr} - \frac{d[J_m(y_{inner} k_r)]}{dr} \frac{d[Y_m(y_{outer} k_r)]}{dr} = 0, \tag{3.9}$$

where y_{outer} and y_{inner} are the height of the inlet duct inner wall and the inner hub radii in the inflow boundary. The axial wavenumber k_a is calculated from

$$k_a = \frac{k}{1 - M_j^2} \left(-M_j \pm \sqrt{1 - \frac{k_r^2(1 - M_j^2)}{k^2}} \right), \tag{3.10}$$

where the selection of plus or minus (\pm) in the parenthesis is determined by the direction of the spinning wave. The constant c_1 satisfies the following relation

$$c_1 = -\frac{\frac{d}{dr}[J_m(y_{outer} k_r)]}{\frac{d}{dr}[Y_m(y_{outer} k_r)]} \tag{3.11}$$

and

$$c_1 = -\frac{\frac{d}{dr}[J_m(y_{inner} k_r)]}{\frac{d}{dr}[Y_m(y_{inner} k_r)]}. \tag{3.12}$$

On the centerline boundary where $r = 0$ a singularity exists. The singularity is

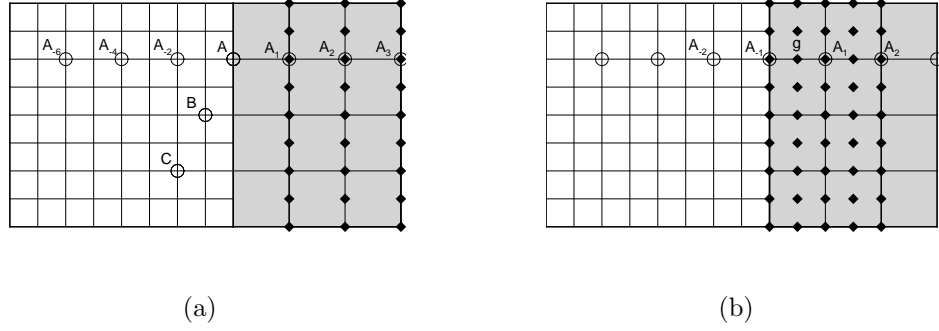


Figure 3.2: Two approaches to compute spatial differences around a fine-coarse interface: (a) a multi-size DRP stencil; (b) a stencil with interpolation, where the shadowed side is the ghost cells area; (o) the original solutions on the block; (◆) the interpolated solutions of ghost cells.

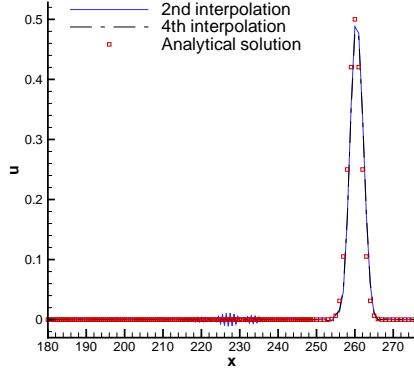
treated by using l'Hopital's rule to approximate $1/r$ by $\partial/\partial r$ at the singularity.

The above equations can be solved numerically by replacing the partial derivatives with finite differences on a discrete numerical grid and advancing the solution in temporal axis via a time-marching algorithm.

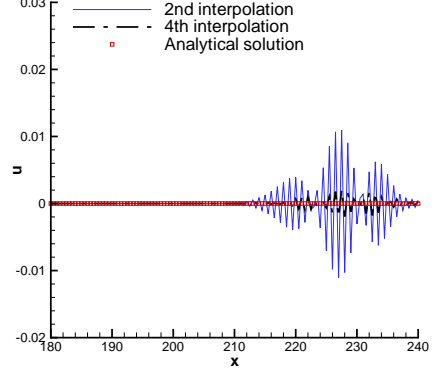
3.2 Spatial Discretization at Fine-Coarse Interfaces

On an adaptively refined mesh, there are interfaces between blocks with different refinement levels. They are called fine-coarse interfaces or fine-coarse block interfaces interchangeably herein. The original spatial difference schemes employed on a uniformly fine mesh do not work correctly at grids around interfaces anymore. There are two approaches to obtain spatial differences of grid points around fine-coarse interfaces. The first approach is designing special stencils based on wavenumber optimizations [110]. For example, in Figure 3.2(a), the stencil from A_{-6} to A_3 is used to compute the horizontal derivative at grid point A . The second approach is constructing a layer of ghost points that have the same cell size as the surrounded block [42]. Figure 3.2(b) shows an example of the approach. The solution at grid point g is obtained by an interpolation method whose coefficients have been listed in Appendix A. The second approach is employed in this work in order to keep the code uniform.

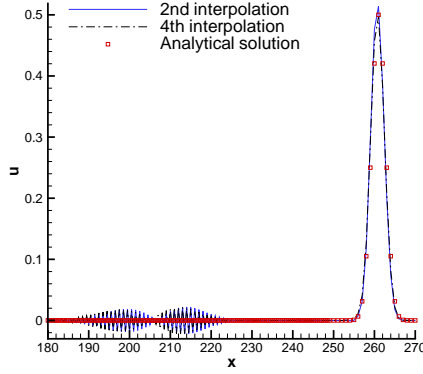
To assess the performance of the second approach, the problem of one-dimensional



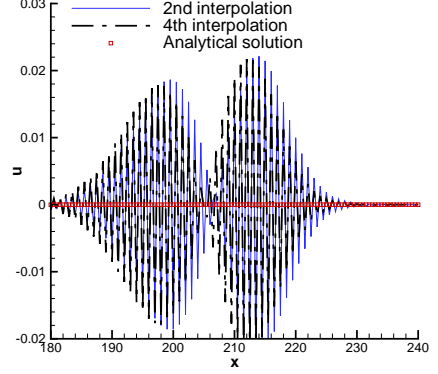
(a) The DRP scheme.



(b) An enlarged part of (a).



(c) The compact scheme.



(d) An enlarge part of (c).

Figure 3.3: Gaussian pulse propagation through a fine-coarse block interface at $x=250$.

advection (Eq. (3.1)) is considered. It is subjected to the following initial condition:

$$u(x, 0) = 0.5e^{-\ln(2)(x-230)^2/4}. \quad (3.13)$$

It is solved over the domain $0 \leq x \leq 750$ using a combination of two fine and coarse meshes. A fine-coarse interface is placed at $x = 250$. The cell size to the left of the interface is set at $\Delta x = 0.5$ and to the right $\Delta x = 1.0$. A classical 4th-order Runge-Kutta method with the same time step $\Delta t = 0.3$ across the mesh is selected to ensure that the numerical errors are essentially caused by the employed spatial scheme. Both the DRP [109] and the prefactored compact schemes [101] have been tested. The test code with the DRP scheme is included in Appendix A.

Solutions at $t = 30$ are shown in Figure 3.3, with either a 2nd- or a 4th-order

interpolation method. Coefficients of the employed spatial schemes and interpolation methods are given in Appendix A. No matter which scheme is used, a similar phenomenon is discovered. It can be seen that the presence of a fine-coarse interface induces short wavelength spurious waves propagating in the domain. Comparing to the approach with the 2nd-order interpolation method, the DRP scheme working with the 4th-order interpolation method reduces the amplitude of the spurious wave remarkably (Figure 3.3(b)). However, the effect of higher order interpolation is not so distinctive if the compact scheme is used (Figure 3.3(d)). It may be caused by the larger boundary stencils of the prefactored compact scheme [101] which lead to more numerical errors and higher amplitude of spurious waves.

The techniques to remove spurious waves are discussed in Section 3.4. Other important issues including stability and accuracy of the spatial schemes employed under the AMR environment are also discussed below.

3.3 Stability Analysis

By using an eigenvalue analysis, asymptotic stability of compact schemes on a uniform mesh was proved in [99, 101]. The same approach is followed over the whole hierarchical mesh to analyse asymptotic stability of the proposed treatment at a fine-coarse block interface. It works for both the explicit and the compact schemes. For the sake of simplicity, only the DRP and the standard explicit schemes [42] are analysed here. The coefficients of the employed schemes are given in Appendix A.

Eq. (3.1) is solved on a computational domain of $x \in [0, 1]$ and inflow boundary condition $u(0, t) = g(t)$. It is assumed that there are only two refinement levels in the analysed mesh. The domain is divided into N cells. A fine-coarse interface is located at the center of the domain, where the left spatial discretization size Δx_l and the right spatial discretization size Δx_r are uniform and satisfy the relation $\Delta x_l = \Delta x_r/2$.

After applying an explicit spatial discretization stencil to Eq. (3.1), it yields a system of ordinary differential equations, which may be written in the vector form:

$$\frac{d\mathbf{U}}{dt} = \mathbf{M}\mathbf{U} + \mathbf{B}g(t), \quad (3.14)$$

where $\mathbf{M} \in R^{N \times N}$. It is obtained from a 4th-order central difference schemes coupled with biased stencils at the inflow boundary. A quantitative example of the matrix \mathbf{M} used in the analysis is given in Appendix A. \mathbf{U} is an N -dimensional vector representing solutions at the nodal points. \mathbf{B} is a vector of dimension N . For the convenience of stability analysis $g(t)$ is set to 0 with little loss of generality [99].

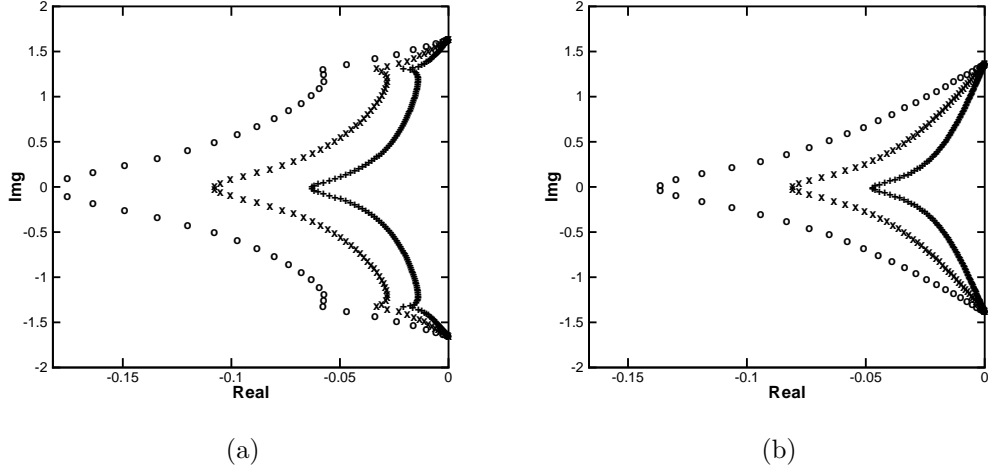


Figure 3.4: Eigenvalue spectra of 4th-order: (a) DRP and (b) standard explicit schemes, working on a uniform mesh with single-side stencils near domain boundaries, (o) $N=50$, (x) $N=100$, (+) $N=200$.

The eigenvalues \mathbf{E} of \mathbf{M} obtained with the help of MATLAB determine the asymptotic stability of the ordinary differential equations. In general, these values are complex numbers depending on the size of \mathbf{M} , an interior spatial differential scheme and a boundary scheme. All eigenvalues \mathbf{E} of \mathbf{M} should lie on the left half of the complex plane to ensure the numerical stability.

Eigenvalues of the example of Eq. (3.1) on a uniform mesh is displayed in Figure 3.4 that illustrates the effect of increasing N for a matrix \mathbf{M} . Either a 4th-order DRP or a 4th-order standard explicit scheme is tested. All eigenvalues are on the left half plane, indicating the spatial scheme used in this experiment is asymptotic stable.

Following the same approach, the eigenvalues of Eq. (3.1) on a hierarchical mesh with a fine-coarse interface is solved and displayed in Figure 3.5, where $N=50$. It shows the distribution pattern of eigenvalues \mathbf{E} change on the hierarchical mesh. In addition, the real parts of some eigenvalues \mathbf{E} become bigger. In spite of that, these values still lie on the left half of the complex plane. For example, $\max(\text{real}(\mathbf{E}))$ is $-8.5e^{-4}$ if $N=50$, $-1.0e^{-4}$ if $N=100$ and $-1.3e^{-4}$ if $N=200$.

In the previous experiments, a 4th-order interpolation is used around the fine-coarse interface. It is worth emphasizing that a 2nd-order interpolation is preferred in some situations, *e.g.* in the case of computing a Jacobian matrix in coordinates transform. The asymptotic stability for the matrix \mathbf{M} with a 2nd-order interpolation has been tested as well and the similar results are obtained. For the sake of brevity they are omitted.

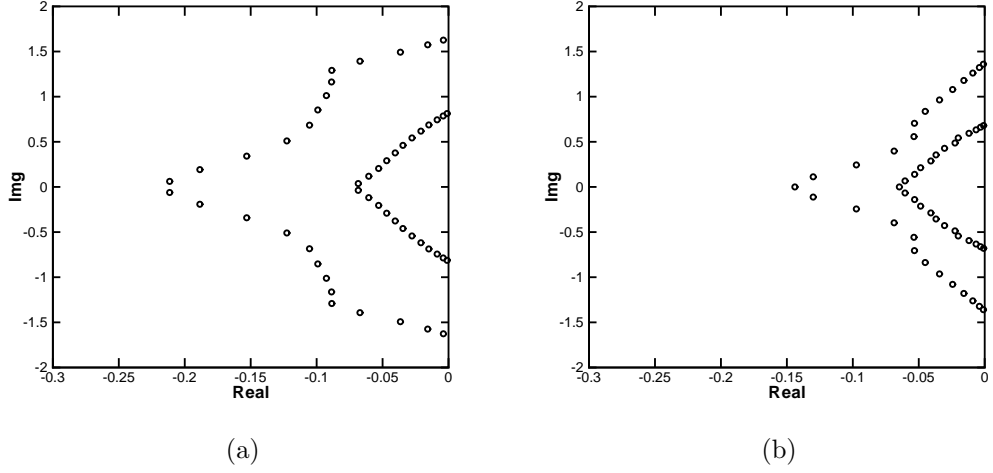


Figure 3.5: Eigenvalue spectra of 4th-order: (a) DRP and (b) standard explicit schemes, working on a hierarchical mesh with single-side stencils near domain boundaries, using a 4th-order interpolation method around the fine-coarse interface, $N=50$.

The mentioned eigenvalue analysis only provides sufficient stability conditions for normal matrix \mathbf{M} that satisfies $\mathbf{M}^* \mathbf{M} = \mathbf{M} \mathbf{M}^*$, where \mathbf{M}^* is the conjugate transpose of \mathbf{M} [139, 140]. However, \mathbf{M} of any high-order spatial scheme used in this work is non-normal. For a non-normal matrix, ϵ -pseudospectra analysis has been used to measure the stability margin of high-order methods [139]. It was also applied to analyse a wave equation [141]. The same technique is employed herein.

According to the definition 2 given by Embree and Trefethen [140], ϵ -pseudospectra are defined in terms of eigenvalues of perturbed matrices:

$$\mathbf{E}_\epsilon(\mathbf{M}) = \{z \in \mathbb{C} : z \in \mathbf{E}(\mathbf{M} + \mathbf{D}) \text{ for some } \mathbf{D} \text{ with } \|\mathbf{D}\| \leq \epsilon\}, \quad (3.15)$$

where \mathbf{D} is disturbance, $\|\cdot\|$ is L_2 -norm, ϵ denotes the supremum of disturbance magnitude.

Using this definition, the mentioned example Eq. (3.1) is solved again with either a DRP or a standard explicit scheme on either a uniform mesh or a hierarchical mesh to find out the corresponding ϵ -pseudospectra. The result of the matrix \mathbf{M} (with $N=50$) is illustrated in Figures 3.6-3.7, where the eigenvalues are plotted as black dots on the complex plane and the solid lines mark the possible distribution areas of pseudospectra values with respect to different ϵ . The coloured bar on the right of the figure denotes $\log_{10}(\epsilon)$. In other words, the distribution of the ϵ -pseudospectra

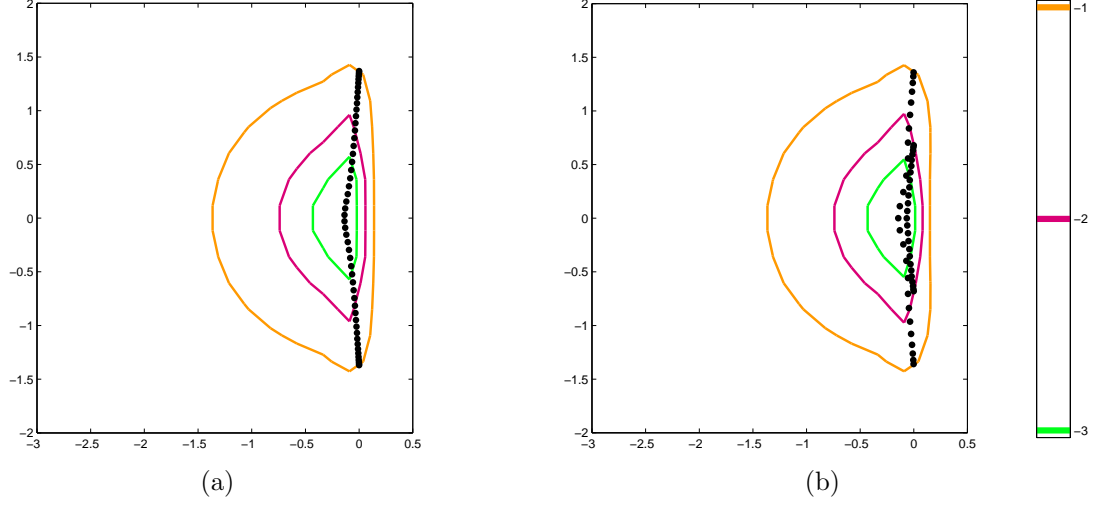


Figure 3.6: Spectrum (black dots) and ϵ -pseudospectra distribution boundaries (coloured lines) of an operator matrix obtained from a 4th-order standard explicit scheme on: (a) a uniform mesh; (b) a hierarchical mesh with a fine-coarse interface around which a 4th-order interpolation method is employed.

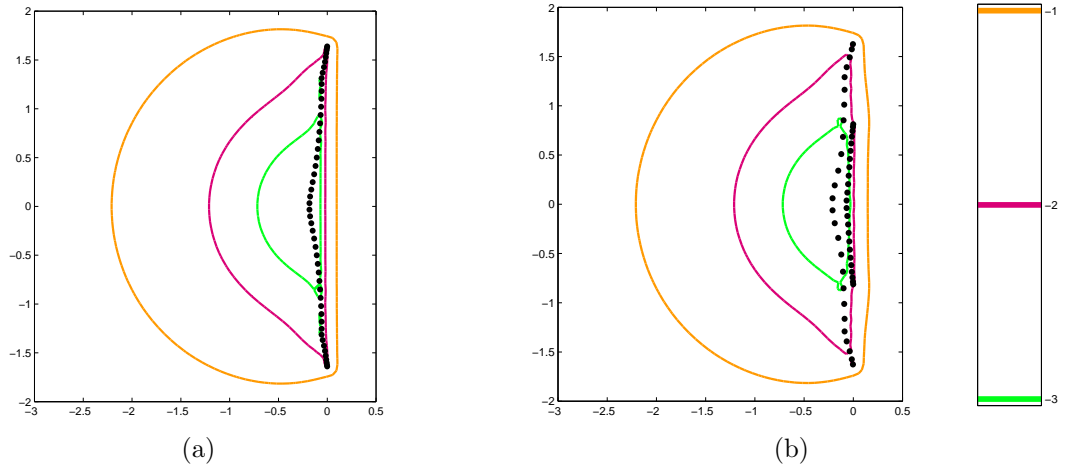


Figure 3.7: Spectrum (black dots) and ϵ -pseudospectra distribution boundaries (coloured lines) of an operator matrix obtained from a 4th-order DRP scheme on: (a) a uniform mesh; (b) a hierarchical mesh with a fine-coarse interface around which a 4th-order interpolation method is employed.

corresponds to $\epsilon = 10^{-1}, 10^{-2}$ and 10^{-3} accordingly.

In Figures 3.6-3.7, it is clear that the potential ϵ -pseudospectra may lie on the right half plane. It implies the high-order spatial schemes employed in this work, no matter on a uniform mesh or a hierarchical mesh, have a transient behavior that differs from the asymptotic behavior suggested by the aforementioned eigenvalues analysis. In other words, these schemes admit spurious numerical solutions induced by disturbances, such as truncation error, in the computational procedure [140, 141]. Generally, the spurious numerical solutions are in the form of a short wavelength spurious wave. Obviously, both the explicit and the compact schemes are central difference schemes and do not provide numerical dissipation to absorb spurious waves. The presence of the fine-coarse block interfaces in an adaptively refined mesh makes the situation worse by introducing nonlinearity to a computational domain. If left unchecked, the spurious waves could ruin solutions or destroy the process of adaptive refinement. To suppress the spurious waves, either a filter or an artificial damping method has been used. The relevant details are given in the next section.

3.4 Artificial Selective Damping and Filters

Several methods can be used to remove spurious waves appeared in the process of computation. The first method is the artificial dissipation that was designed for capturing shock as well as giving sufficient numerical stability to a central difference scheme [114]. The method is generally leading to excessive dissipation for the time-dependent CAA problems. In contrast, the methods of artificial selective damping [110] and implicit/explicit filters [9, 142] absorb spurious numerical waves in the unresolved high wavenumber, whilst kept the resolved wave components unaffected. Among these methods, an implicit filter provides better performance than an explicit filter. However, it is too complicated to implement on an adaptively refined mesh. Artificial selective damping and an explicit filter are therefore used in this work.

An explicit filter is operated over the original solutions. It generally takes the form:

$$\bar{u}_i = u_i - \beta \sum_{k=0}^K a_k (u(i+k) + u(i-k)), \quad (3.16)$$

where u is the original variable, \bar{u} the filtered variable, u_i is the solution at the i^{th} gridpoint. Besides, single-side biased filters are used near the boundaries of a computational domain. Details of a 10th-order explicit filter employed in this work are given in Appendix A.

The original governing equations remain in the event of using the explicit filter. In

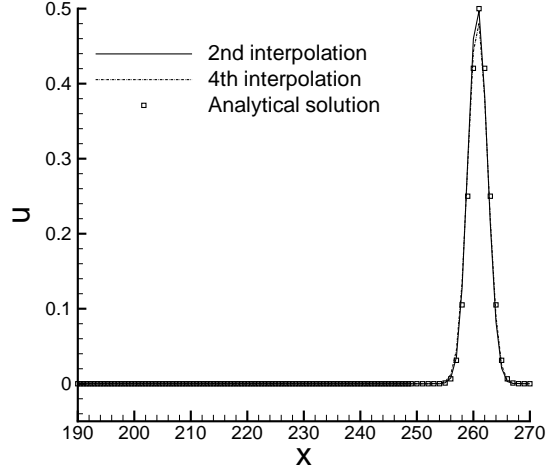


Figure 3.8: One-dimensional Gaussian pulse propagation with a 10^{th} -order explicit filter and a 4^{th} -order Compact scheme.

contrast, an extra viscous term is introduced to the governing equations when using the artificial selective damping method. Taking Eq. (3.1) as an example, by including artificial selective damping terms [110] and assuming discretization over a uniform grid, the discrete form of the scalar wave equation at gridpoint i can be rewritten as:

$$\left(\frac{\partial \bar{u}}{\partial t}\right)_i + \left(\frac{\partial u}{\partial x}\right)_i = -\frac{v_a}{(\Delta x)^2} \sum_{j=-3}^3 d_j u(i+j), \quad (3.17)$$

where u is the original variable, \bar{u} the damped variable, v_a an artificial kinematic viscosity, j the damping stencil index, d_j damping coefficients and Δx the spatial discretization size. The specific coefficients are also given in Appendix A.

Several benchmark cases are used to test the effect of the mentioned methods, including Gaussian pulse propagation in both one-dimensional and two-dimensional spaces.

The first case of the one-dimensional problem solves Eq. (3.1) with the initial condition given by Eq. (3.13). At the fine-coarse interface ($x = 250$), both the 2^{nd} - and the 4^{th} -order interpolations are tested. A 10^{th} -order explicit filter is used throughout the computational procedure. The result is displayed in Figure 3.8. Compared to Figure 3.3, it is clear that the spurious waves generated in the computational domain are suppressed with both interpolation methods applied at the fine-coarse interface.

The second test case is the two-dimensional acoustic propagation problem, which

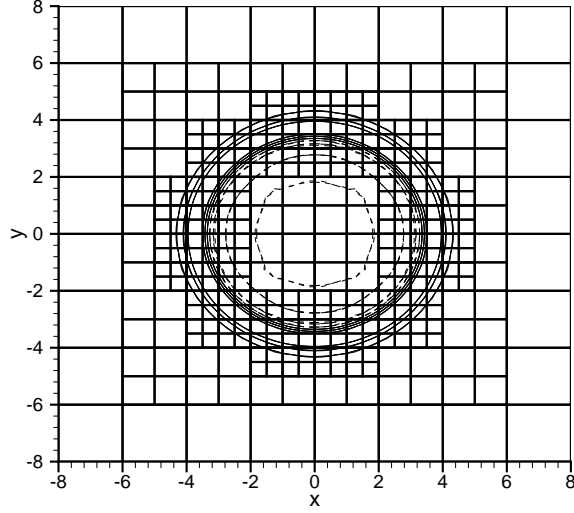


Figure 3.9: Two-dimensional Gaussian pulse propagation with an artificial selective damping method and a 4th-order DRP scheme, where pressure contours with 6 levels between ± 0.05 are displayed.

is governed by Eq. (3.2). The initial conditions are:

$$p(x, y, 0) = e^{-a(\frac{x^2+y^2}{0.1})}, \quad u(x, y, 0) = 0, \quad v(x, y, 0) = 0, \quad (3.18)$$

where the computational domain covers an area of $-8 \leq x \leq 8$ and $-8 \leq y \leq 8$. The problem is solved under the AMR environment. The two-dimensional problem is solved on an adaptive mesh that contains two refinement levels. Initial Gaussian pulse is located at the center of the mesh, only on which the finer mesh is superimposed to increase the resolution. After several computing time steps, the existing mesh is tested and regridded to capture the propagation of the wave. The spatial discretization is performed by the 4th-order DRP scheme and the temporal integration by the 4-6 low-dissipation and low-dispersion Runge-Kutta method [111]. The 4th-order interpolation is employed around the fine-coarse interfaces. The 4th-order artificial selective damping is used throughout the computational procedure. Without using the damping method, spurious waves are generated at the fine-coarse block interfaces and will cumulate towards the centre of the computational domain. If not treated properly, the centre area of the computational domain may be refined improperly or will not be coarsened. Introducing the damping method removes the spurious disturbances and allows the refinement process to run smoothly. A sample of the pressure waves is shown in Figure 3.9.

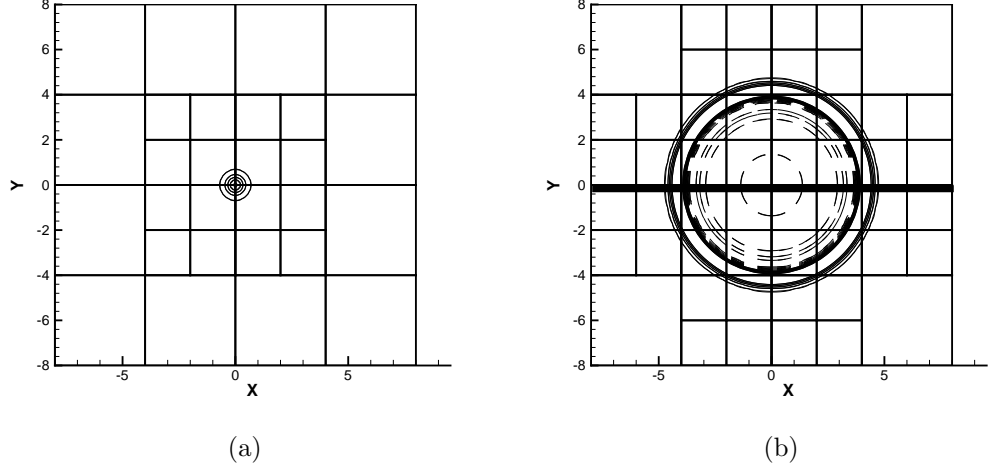


Figure 3.10: Pressure contours: (a) $t=0.1$; (b) $t=4$, where there are sixteen contour levels between ± 0.05 . Gray lines are blocks' borders. Solutions on the centre dark line in (b) are compared to the analytical solution.

3.5 Convergence Rate

To examine the global convergence rate on an adaptive mesh, the previous case is used again. Figure 3.10 illustrates the setup. The case has been tested with several different schemes (coefficients are listed in Appendix A). The 4-6 low-dissipation and low-dispersion Runge-Kutta [111] is used for temporal integration. The 10^{th} -order filter [143] is employed throughout the domain to remove spurious waves. In the ghost construction operation, several interpolation methods (2^{nd} -/ 4^{th} -order linear interpolations and a 6^{th} -order polynomial interpolation) have been tested. In order to guarantee that the wave is always contained in the finer mesh, the interval of mesh regridding (T_{amr}) is five computing steps, which is set in accordance with the temporal step (Δt) and the length of the ghost area (W_g) used in this test case.

The L_2 -norm errors of pressure are plotted in Figure 3.11, where the cells number N of each block is increased from 20×20 up to 50×50 , *i.e.* the total cells number ranges from 6,400 to 200,000.

The results show that different interpolation methods may affect the convergence rate. A 2^{nd} -order interpolation keeps the 2^{nd} -order convergence rate when working with a 2^{nd} -order explicit scheme. It may, however, degrade the convergence rate below 3 when a 4^{th} -order explicit scheme is employed. In general cases, a 6^{th} -order interpolation method is suggested to work with a 4^{th} -order spatial scheme to keep the 4^{th} -order convergence rate. For example, a two-dimensional interpolation with a

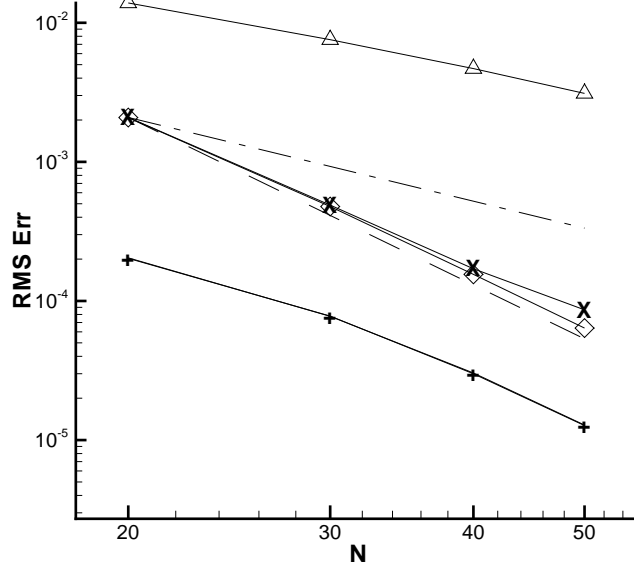


Figure 3.11: L_2 -norm error of the pressure solution, where N is the number of cells on each block border. Several schemes employed are: (Δ) the 2nd-order standard explicit scheme and the 2nd-order interpolation; (X) the 4th-order standard explicit scheme and the 2nd-order interpolation; (\diamond) the 4th-order standard explicit scheme and the 6th-order interpolation; (+) the 4th-order DRP scheme and the 6th-order interpolation; (-.-) the 2nd-order ideal slope; (- -) the 4th-order ideal slope.

36-points stencil was used in [18]. However, it is too cumbersome to implement. By contrast, in this work a 6th-order polynomial interpolation [137] is operated in each coordinates direction separately. The convergence rate is increased to around 3.7 as N increases.

3.6 Summary

In this chapter numerical issues associated with this work are introduced. The first section summarises the governing equations used to solve CAA problems in this work. The second section studies the spatial schemes applied at a fine-coarse interface that is associated with AMR. The third section discusses the stability of the spatial schemes employed on either a uniform mesh or a mesh with a fine-coarse interface. Both asymptotic and transient behaviours are analysed by computing the eigenvalues and the ϵ -pseudospectra of the operator matrix. The results suggest that spurious waves will be generated in the computational domain for both meshes. The fourth section

then introduces the techniques to remove spurious waves in the computation. A 10^{th} -order explicit filter and a 4^{th} -order artificial selective damping method are tested. Both methods remove spurious waves satisfactorily. Finally, the fifth section presents the overall accuracy results of the mentioned methods on an adaptively refined mesh. The convergence rates of the 4^{th} -order standard explicit scheme and the 4^{th} -order DRP scheme are around 3.7 when working with the 6^{th} -order interpolation method. Hence both schemes are used in the following case studies.

Chapter 4

Results of Benchmark Problems

This chapter solves some benchmark problems with the AMR method, which function and efficiency are validated against analytical solutions.

4.1 Introduction

When applying the AMR method to CAA applications, there are several potential problems/issues worried by most people. The first one is whether the AMR method requires reasonable programmer effort. The second one is whether the parallel AMR operations require reasonable cost. The third one is whether the original accuracy of solutions on a uniformly fine mesh is still kept on an adaptively refined mesh. The first question has been answered in Chapter 2 by discussing the details of the implementation. To answer the latter two questions, the algorithm of the parallel block-based AMR method is decomposed and analysed in this chapter. Several benchmark problems are solved under the AMR environment. Results are compared with available analytic solutions.

The parallel test environment includes a cluster running Red hat Linux 9, with seven nodes connected by a Gigabit switch; each node contains four Intel Xeon™ 3.06GHz CPU sharing 2GBytes memory. The portability of the code is satisfactory, which can be compiled with any popular compiler, *e.g.* Intel, PGI and GNU G95 Fortran Compiler, and is working fine with either MPICH or LAM/MPI. The profiling data are given by using a GNU profiler, gprof [144].

4.2 Two-dimensional Acoustic Wave Propagation

In order to verify the AMR code, the first problem of category 4 defined at the first computational aeroacoustics workshop [145] is considered. This benchmark problem also validates the parallel performance of the code.

In this case two-dimensional wave equations (Eq. (3.2)) appeared in Chapter 3 are used. A 4th-order DRP scheme [109] is employed to compute the spatial derivatives and a 4th-order low-dissipation and low-dispersion Runge-Kutta method scheme [111] is employed for the temporal integrations. The initial computational domain is defined to contain 4 refinement levels with 100 cells per each block, where the whole domain is adaptively refined by using the regridding operation so as to capture the acoustic propagation. Subsequently the blocks number develops from 16 to 1,360. In other words the cells number increases from 1,600 to 136,000.

In this work, the refinement criterion employed in an application is designed individually to suit the specific physical and computational requirements. For the reason that the amplitude of the gradient of pressure, $\|\nabla p\|$, reflects the sound propagation procedure, it is employed in this and the next case studies. The regridding flag ζ , therefore, is constructed in the form of:

$$\zeta = \frac{D_{block}}{D_{global}}, \quad (4.1)$$

where D_{block} and D_{global} are the local block maximum and global maximum of the $\|\nabla p\|$ respectively. A block will be: (a) refined if ζ is larger than a predefined refinement threshold τ_r ; (b) coarsened if ζ is smaller than a preset coarsening threshold τ_c ; and (c) retained if $\tau_c < \zeta < \tau_r$. The mesh is regridded in every T_{amr} time steps. The bigger this interval is the higher computation efficiency. However there is a tradeoff between efficiency and accuracy in terms of the computational cost. In the following experiments the width of the ghost area is generally set to 5 and the CFL number is always less than 1. T_{amr} is subsequently set to 5 to always resolve the interested sound solutions on the finest mesh.

The instantaneous pressure contours with six levels between ± 0.05 are displayed in Figure 4.1(a). The dynamic load balancing operation is executed after each regridding operation, by which the total computational load is distributed evenly amongst processors over the whole computational procedure. An example of the load balancing result amongst six processors is shown in Figure 4.1(b), where the whole computational domain is partitioned evenly.

Although the AMR algorithm provides the potential of increasing the compu-

4. RESULTS OF BENCHMARK PROBLEMS

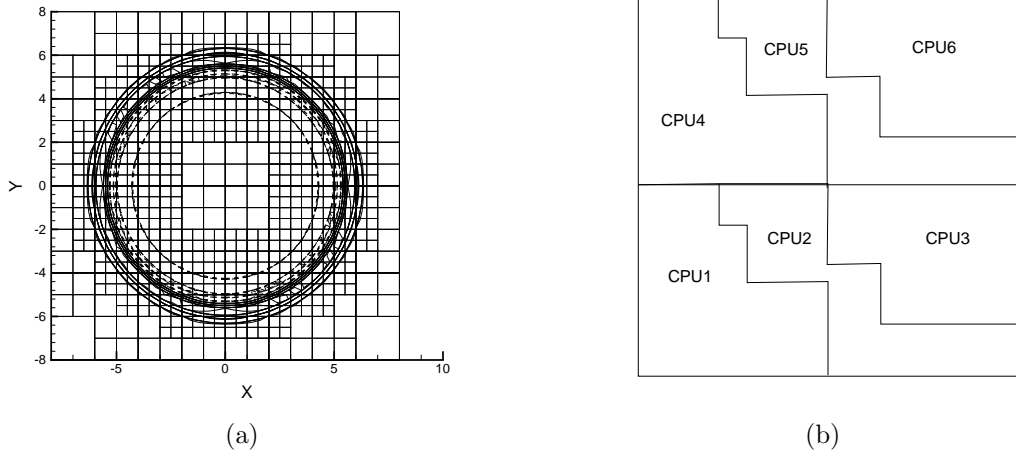


Figure 4.1: Parallel computational results of two-dimensional acoustic propagation: (a) pressure perturbation contours; (b) the load distribution among six CPUs.

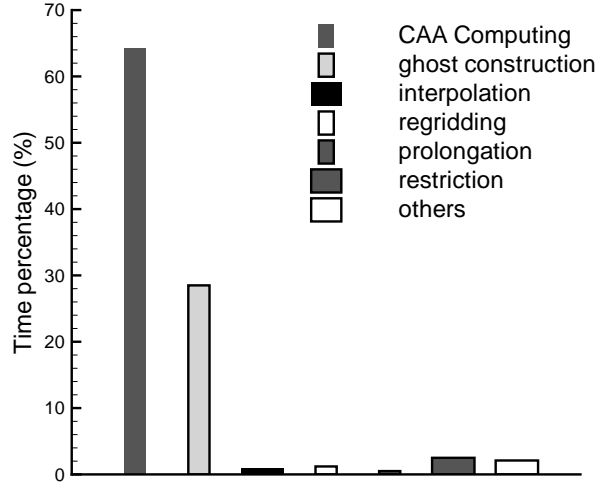


Figure 4.2: The computational time percentage of the code in the case study of two-dimensional acoustic propagation working on a single CPU.

tational efficiency, it incurs an extra computational cost. A qualitative analysis of the cost of AMR has been given in the previous chapter. In this chapter the code is profiled in order to examine the cost of AMR quantitatively. Firstly, the test is operated on a single processor. The result of this benchmark case is displayed in Figure 4.2. It shows that the total costs of AMR operations are around 34%, in

4. RESULTS OF BENCHMARK PROBLEMS

which the most part is consumed by the operation of ghost construction, whereas the costs of regridding, prolongation and restriction operations are trivial. Meanwhile, the CAA computation consumes 64% CPU time and the collective communications, memory management and file input-output consume the other 2% CPU time.

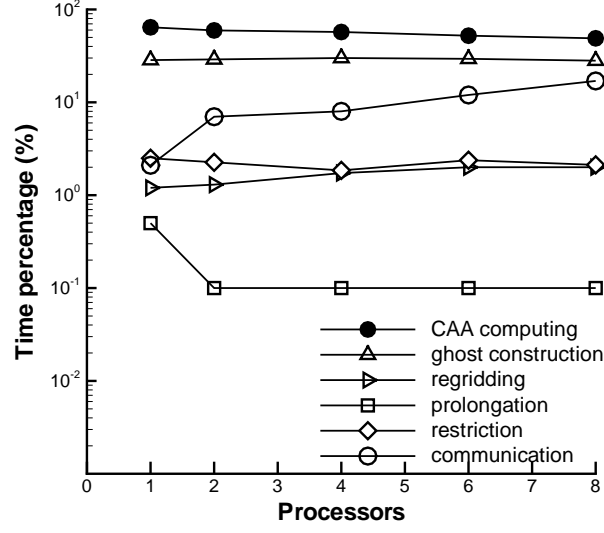


Figure 4.3: Time percentage of every subroutine in the case study of two-dimensional acoustic propagation.

Secondly, the test is operated on up to eight processors to indicate the parallel performance. The results are displayed in Figures 4.3–4.4, where values in the time axis are shown in the log scale. Figure 4.3 shows that the percentages of the CAA computing and most AMR operations keep constant along with the increase of the processors number. However, the percentage of the prolongation operation is reduced slightly, whereas the parallel communication associated with AMR operations is increased from an insignificant part to a fairly big part. The rapid increase on the communication cost percentage is caused by the increased communication in the computational procedure with the AMR method.

The parallel speedup results are plotted in Figure 4.4, where the term of speedup refers to how much an algorithm working on a parallel machine is faster than the corresponding algorithm working on a single processor machine. To show the figure clearly, only the main results are displayed. It shows that the regridding operation gains the poorest speedup performance because it has to maintain a parallel tree representing hierarchical meshes throughout parallel processors. The speedup performance of others is affected by the communication cost extensively.

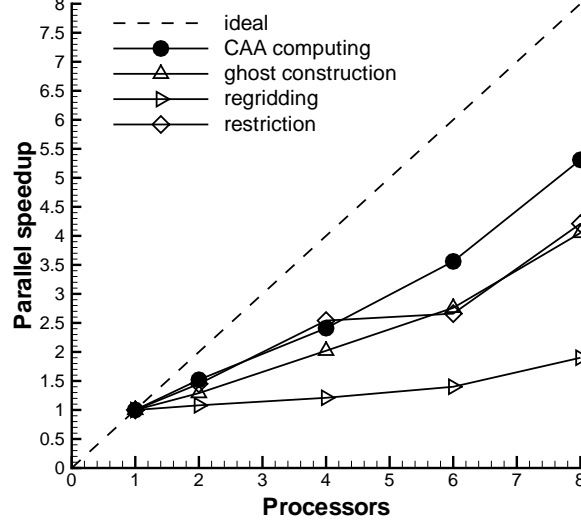


Figure 4.4: Performance speedup in the case study of two-dimensional acoustic propagation.

Generally speaking, in AMR operations most costs are spent on the ghost construction operation, which consumes from 20% to 30% of the total computing costs. Not only for AMR, the same operation is also required for computing spatial differences of gridpoints around block interfaces on a uniformly fine mesh. It is also expensive and generally costs 5% to 15% of the total computing costs in the established SotonLEE code [7], depending on the specific case conditions. The ghost construction operation in this AMR method consumes more computing resources for its simplified algorithm that has false sharing, which has been introduced in Chapter 2, to reduce the programming efforts whereas to increase the communication costs. In order to mitigate the costs of this operation, the ghost construction operation should be rewritten to reduce the false sharing communications in the future work.

Although all results provided here are for the case study of two-dimensional acoustic propagation only, another parallel speedup result presented in the next chapter gives the similar information that confirms the quantitative discussion herein.

4.3 Acoustic Wave Scattering from Cylinder

For applications with general geometries, the generation of a body-fitted mesh is always a tedious task. The use of boundary non-conforming grids with the immersed boundary method is a potential solution [91].

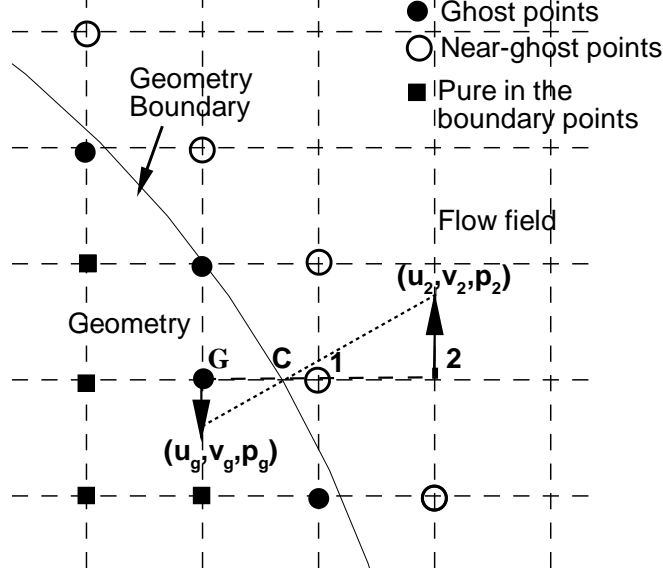


Figure 4.5: A schematic of the immersed boundary method.

Figure 4.5 displays the working procedure that flags grids surrounding the geometry automatically. First of all, all grids inside of the immersed boundary are found and identified. The inside boundary grids are then identified as the ghost points if at least one of their neighbouring grids is outside of the immersed boundary. After that the neighbouring grids of the ghost points are flagged as near-ghost grids if they are located in the flow domain. Finally the total domain is divided into four areas: ‘pure’ boundary grids, ghost grids, near ghost grids and ‘pure’ flow grids.

The solid boundary condition is then enforced by setting the values of ghost points according to the solutions of surrounding near-ghost grids. Either a 0^{th} - or a 1^{st} -order interpolation method is employed to simulate a slip-wall boundary condition. Higher order interpolations constructing an immersed boundary condition are too complex to employ here. For the x-axis direction, the simpler 0^{th} -order interpolation used in the work is:

$$u_g = -u_1, \quad p_g = p_1. \quad (4.2)$$

The 1^{st} -order interpolation employed is:

$$u_g = -\frac{(x_2 - x_c)u_2}{(x_c - x_g)}, \quad p_g = \frac{(x_2 - x_c)p_2}{(x_c - x_g)}, \quad (4.3)$$

4. RESULTS OF BENCHMARK PROBLEMS

where u is velocity in that direction and p is the pressure. x_c is the cut-through position in x-axis. x_2 is the second nearest points to ghost in the flow. The reason to use x_2 rather than the nearest ghost point, *i.e.* x_1 , is to keep numerical stability. It is noted that by the 0^{th} -order interpolation a cut-cell boundary is actually degenerated to a stepwise boundary. It avoids the problem of stability, whereas increases the numerical error. The mentioned interpolation methods have been analysed by comparing L_2 -norm results to show their performance in the immersed boundary method [146]. The same analysis of the convergence result is not repeated here.

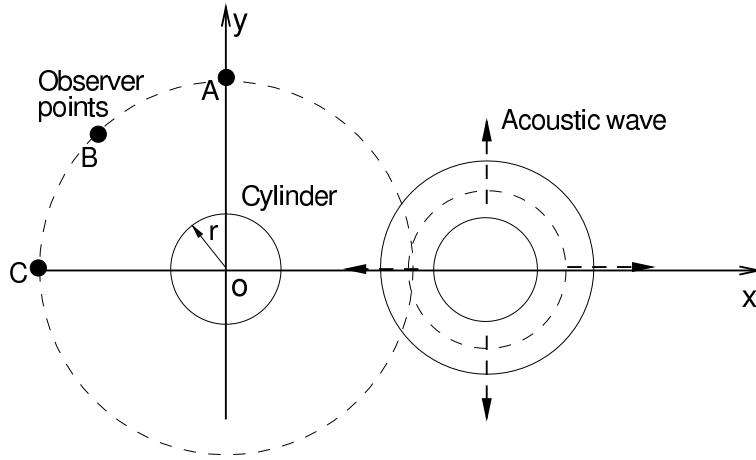


Figure 4.6: The two-dimensional acoustic scattering off a cylinder.

To study the presented approach under the AMR environment, the two-dimensional acoustic scattering problem from the second CAA workshop [122] is solved here. It is sketched in Figure 4.6. A cylinder with the radius of 0.5 is located at the origin, whilst an initial acoustic wave propagates towards and scatters off the cylinder. The problem asks for the unsteady pressure time history at three observer points $A(x = 0, y = 5)$, $B(x = -\frac{\sqrt{5}}{2}, y = \frac{\sqrt{5}}{2})$ and $C(x = -5, y = 0)$, over the time interval $t = 5 \rightarrow 10$. The solution is found by solving the LEE, Eq. (3.2). The initial sound pressure is given by

$$p(x, y, 0) = e^{-\log(2.0)((x-4)^2+y^2)/0.04}. \quad (4.4)$$

All variables are nondimensionalised with reference values mentioned in Chapter 3.

Spurious reflection waves from the outflow boundaries of the domain are absorbed by an explicit form of the buffer zone technique [147]. The solution vector is explicitly damped after each time step in the buffer zone using:

$$\mathbf{F}(x, y, t + \Delta t) = \bar{\mathbf{F}}(x, y, t + \Delta t) - \sigma(\bar{\mathbf{F}}(x, y, t + \Delta t) - \mathbf{F}_0(x, y)), \quad (4.5)$$

4. RESULTS OF BENCHMARK PROBLEMS

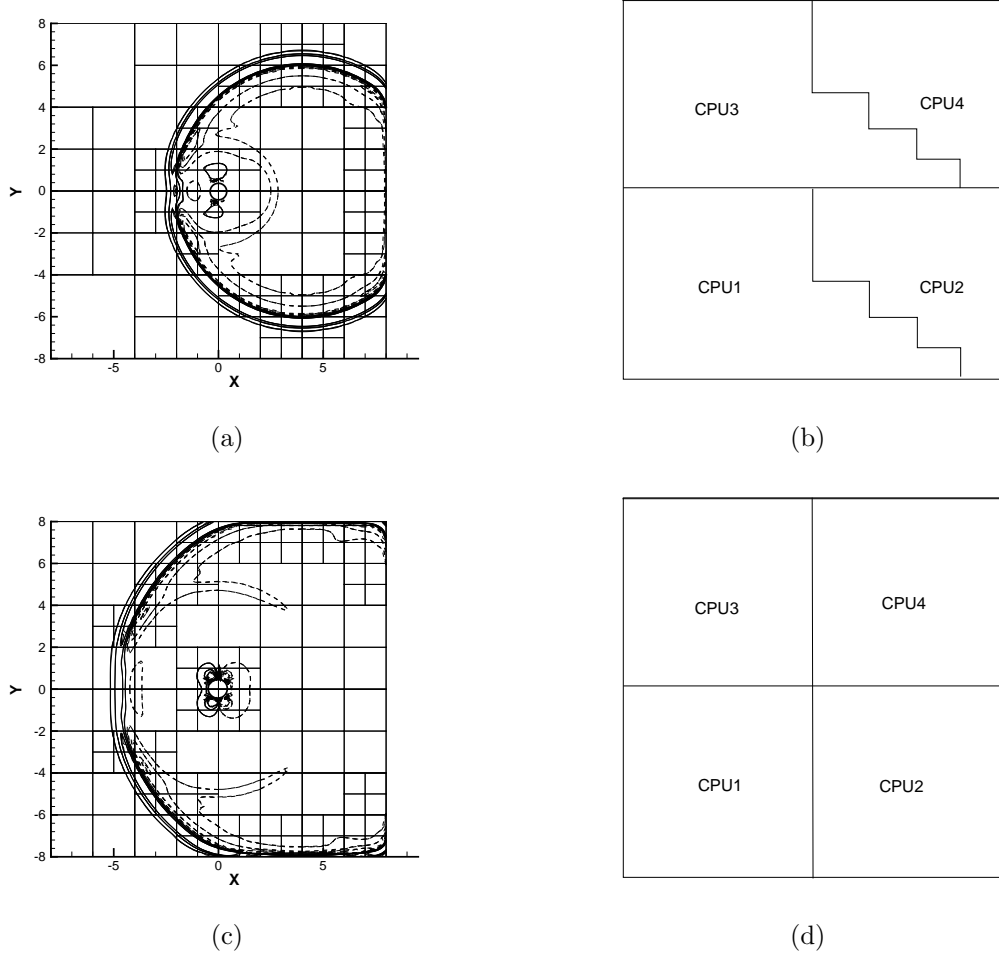


Figure 4.7: Sound pressure contours and dynamic load balancing in the case study of two-dimensional acoustic scattering off a cylinder, where (a-b) $t = 6.125$, (c-d) $t = 8.75$.

where $\bar{\mathbf{F}}(x, y, t + \Delta t)$ is the solution vector after each time step. The damping coefficient, σ , varies according to the function:

$$\sigma(x, y) = \sigma_{max} \left| \frac{L_{bz} - \chi_{bz}}{L_{bz}} \right|^\beta, \quad (4.6)$$

where L_{bz} is the width of the buffer zone, χ_{bz} is the distance between the damping position and the outer boundary of the buffer zone. σ_{max} and β are coefficients that determine the exact nature of the damping and set to 1.0 and 3.0 respectively. The target solution \mathbf{F}_0 is set as zero. The size of the buffer zone is set at 10, which proves to be enough in the study.

In the vicinity area of the solid wall boundary, a 4th-order single-side spatial

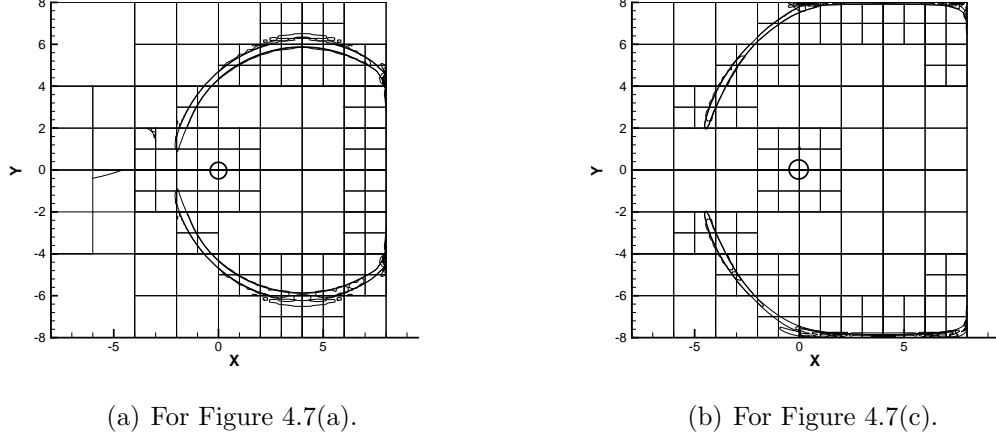


Figure 4.8: The AMR regridding flag of the two-dimensional acoustic scattering off a cylinder.

scheme [148] is employed combining with the aforementioned 0^{th} - or the 1^{st} -order interpolation methods. In other area a 4^{th} -order DRP scheme is used. The resulted pressure contours are displayed in Figure 4.7(a) and Figure 4.7(c) with six level contours between ± 0.02 . A hierarchical mesh with three refinement levels is constructed prior to the computation and is regridded and redistributed amongst parallel processors in the subsequent process of computation (Figs 4.7(b) and Figs 4.7(d)). The corresponding regridding flags are displayed in Figure 4.8.

Finally, Figure 4.9 compares the time history results with analytical solutions at three observer points. There are visible dissipations for AMR results at all points, especially evident at point C, which are partially contributed by the artificial viscosity terms added to governing equations to suppress the high-frequency spurious wave and partially contributed by the distortion of the low-order interpolations employed in the immersed boundary method. Therefore, in the following chapter the existing AMR code is extended to support a body-fitted multi-block mesh as was used in [62].

4.4 Spinning Mode Duct Radiation

The spinning mode acoustic radiation from a duct is of the particular interest in this work. Firstly, a benchmark case, the spinning mode acoustic radiation from an unflanged duct described by Homicz & Lordi [149], is solved to verify the performance of the AMR code. A schematic of the problem is shown in Figure 4.10. Only half of the complete geometry is displayed as the problem is axisymmetric. A spinning

4. RESULTS OF BENCHMARK PROBLEMS

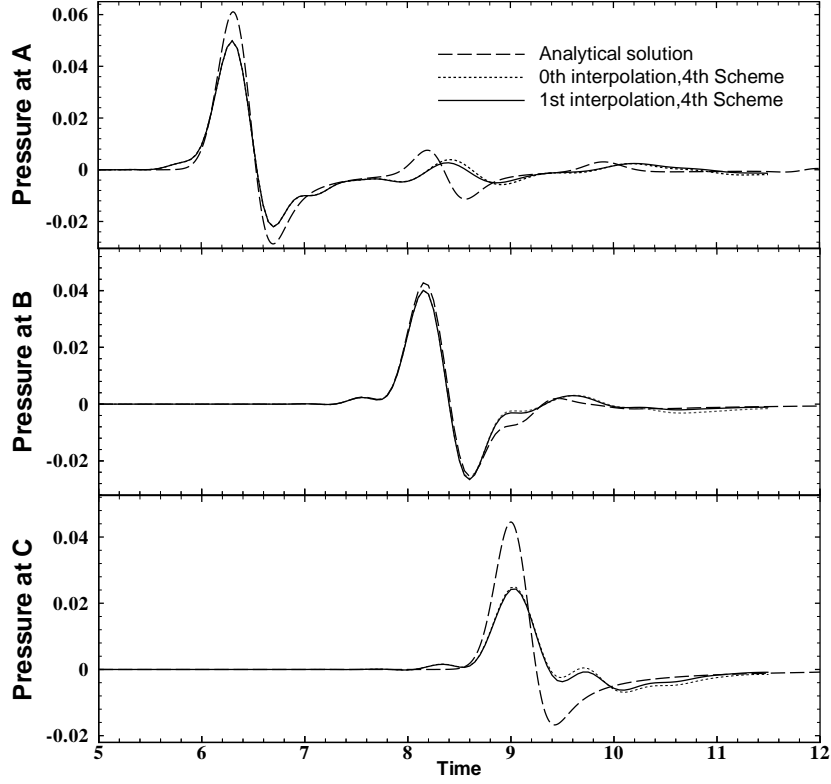


Figure 4.9: The pressure history at three observer points demonstrates the performance of several cut-cell finite-difference approaches of the immersed boundary method.

mode acoustic wave is introduced into the computational domain at the left bottom inlet area. It propagates inside and radiates from the duct. An absorbing condition is applied on the outflow boundary to remove numerical reflections.

As mentioned in the previous chapter, assuming small perturbations (ρ', u', v', w') about a steady mean flow (ρ_0, u_0, v_0, w_0) , acoustic wave propagation is described by the LEE, Eq. (3.7), that can be simplified further if the steady mean flow considered is $(\rho_0, u_0, 0, 0)$, where ρ' is density perturbation, ρ_0 density, (u', v', w') axial, radial and azimuthal velocity perturbations, (u_0, v_0, w_0) axial, radial and azimuthal velocity. Moreover, y denotes radial axis here in order to be differentiable with the far-field observer radius. Subsequently, the governing equations for a single frequency k are:

$$\begin{aligned} \frac{\partial \rho'}{\partial t} + u_0 \frac{\partial \rho'}{\partial x} + \rho_0 \left(\frac{\partial u'}{\partial x} + \frac{\partial v'}{\partial y} + \frac{kv' - mw'_t}{ky} \right) &= 0, \\ \frac{\partial u'}{\partial t} + u_0 \frac{\partial u'}{\partial x} + \frac{\partial p'}{\rho_0 \partial x} &= 0, \end{aligned}$$

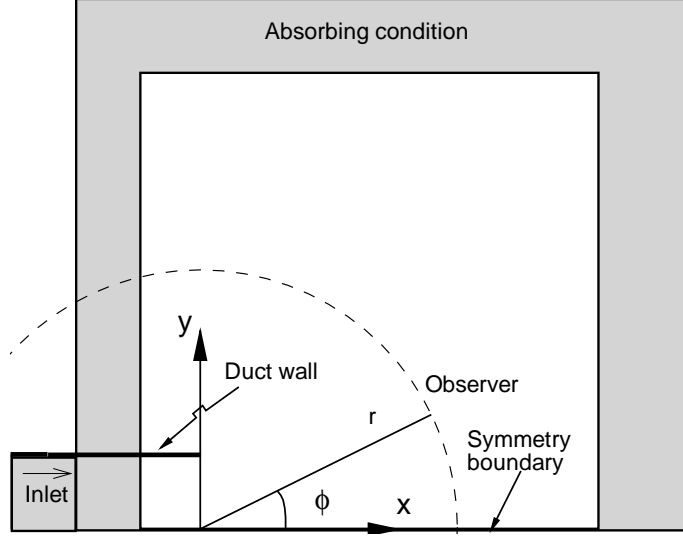


Figure 4.10: Schematic of the computation domain for the case study of sound radiation from an axial symmetrical duct.

$$\begin{aligned} \frac{\partial v'}{\partial t} + u_0 \frac{\partial v'}{\partial x} + \frac{\partial p'}{\rho_0 \partial y} &= 0, \\ \frac{\partial w'_t}{\partial t} + u_0 \frac{\partial w'_t}{\partial x} + \frac{mk}{\rho_0 y} p' &= 0, \end{aligned} \quad (4.7)$$

where x and y are axial and radial coordinates; p' is pressure perturbation; w'_t is the time derivative of azimuthal perturbation velocity, $w'_t = \partial w' / \partial t$; m is the azimuthal mode number. In this case study the nondimensional radius of the duct is 1. Two computation domains are used: (a) 8×8 to test the AMR code and (b) 16×16 to give the far-field solutions.

Boundary conditions include slip-wall, inflow and non-reflecting boundary conditions. The wall of the duct is regarded as an infinitely thin hard wall. Spurious reflection waves are absorbed by an explicit form of the buffer zone technique applied in the surrounding outflow area [147]. The solution vector is explicitly damped after each time step in the buffer zone using Eqs. (4.5)–(4.6). The target solution \mathbf{F}_0 is also set as zero. The size of the buffer zone is set at 13 for this case problem.

A single (m, n) mode, where m is the azimuthal mode number and n is the radial mode order, propagating and radiating from a semi-infinite unflanged duct with thin rigid walls is taken as the incident acoustic perturbation with the form of Eq. (3.8). In the present study $(m = 4, n = 1)$ mode is used. Assuming $u_0 = 0$ the incident

4. RESULTS OF BENCHMARK PROBLEMS

wave is defined as follows:

$$\begin{aligned}
 u'(x, y, \theta, t) &= \frac{ak_x}{k} J_m(k_y y) \cos(kt - m\theta - k_x x), \\
 v'(x, y, \theta, t) &= -\frac{a}{k} \frac{dJ_m(k_y y)}{dy} \sin(kt - m\theta - k_x x), \\
 w'(x, y, \theta, t) &= \frac{am}{k_y} J_m(k_y y) \cos(kt - m\theta - k_x x), \\
 p'(x, y, \theta, t) &= a J_m(k_y y) \cos(kt - m\theta - k_x x),
 \end{aligned} \tag{4.8}$$

where the dimensionless spinning mode amplitude a is set to 10^{-4} to ensure small relative changes in density (as required by LEE). J_m is the m^{th} -order Bessel function of the first kind. The radial wavenumber k_y is determined by computing the turning points of the Bessel function. The axial wavenumber k_x is calculated using $k_x = \sqrt{k^2 - k_y^2}$.

At the lip of the duct, a special treatment is applied to guarantee consistency between computational blocks. More details can be found in the description of the trailing edge singularity of an airfoil [142].

For the results presented here, the 4th-order DRP scheme is used to compute the spatial derivatives and the 4th-order low-dissipation and low-dispersion Runge-Kutta method is used for the temporal integration.

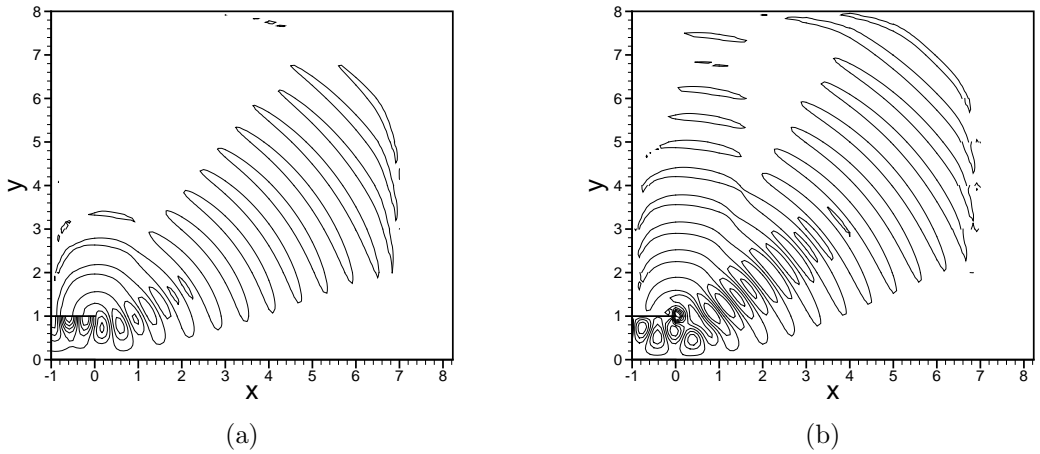


Figure 4.11: The spinning mode sound radiation from an unflanged duct solved on a uniform mesh: (a) contours of pressure perturbation; (b) contours of radial velocity perturbation, where $m = 4$, $n = 1$, $k = 10$.

For this spinning mode radiation ($m = 4, n = 1$), the radiation pattern at a frequency of $k = 10, k_y = 5.3176$ is displayed in Figure 4.11. The computation is

4. RESULTS OF BENCHMARK PROBLEMS

performed on a uniform mesh of 104×104 grid points. From the figure, it is evident that fine spatial resolution is not necessary in large areas of the computational domain except in the area around the duct. Accordingly, AMR can be used to reduce the cost of computation by refining the mesh to track the propagation and radiation of sound. In the following case studies a hierarchical mesh with three refinement levels is therefore used and the same time step is applied to advance the solutions. The regridding criterion employed in this case depends on the amplitude of the gradient of the radial velocity perturbation, $\|\nabla v'\|$. More precisely, the regridding flag ζ is:

$$\zeta = \frac{D_{block}}{D_{global}}, \quad (4.9)$$

where D_{block} and D_{global} are local block maximum and global maximum of $\|\nabla v'\|$ respectively. A block will be: (a) refined if ζ is larger than a predefined refinement threshold τ_r ; (b) coarsened if ζ is smaller than a preset coarsening threshold τ_c ; and (c) retained if $\tau_c < \zeta < \tau_r$. The mesh is regridded in every T_{amr} time steps, which is set to 5 empirically in this study.

For the same spinning mode radiation ($m = 4, n = 1$), an AMR computation is performed. Figure 4.12 shows the corresponding development of the adaptively refined mesh. When the incident waves are inside the duct only the blocks in the immediate vicinity of the duct are refined; blocks located away from the duct remain coarse. The process of wave diffraction at the lip and propagation outside the duct is captured by the AMR method. Some reflections on the fine-coarse block interfaces can be detected in Figures 4.12(b)-4.12(c). However either the damping or the filter reduces the amplitude of spurious waves in the iteration loop and guarantees that these oscillations will not grow up to corrupt the computation process.

Several values of thresholds τ_r and τ_c are tested. The result presented in Figure 4.13(a) is obtained with $\tau_r = 0.15$ and $\tau_c = 0.1$. The AMR process fails to capture the process of wave diffraction at the lip and the sound propagation pattern does not have satisfactory details. By contrast, in Figure 4.13(b) result is obtained with τ_r reduced to 0.015 and τ_c to 0.01. The diffraction at the lip and the radiation pattern are captured correctly.

The directivity of the radiated sound is also used to assess the performance of the current AMR algorithm. The cases employed are: (a) 104×104 uniform mesh; (b) AMR with $\tau_r = 0.15$, $\tau_c = 0.1$; and (c) AMR with $\tau_r = 0.015$, $\tau_c = 0.01$. Figure 4.14 compares instantaneous pressure of three cases at a nondimensional observer distance of 12. It shows that, although case (b) has the best computational efficiency, the computation does not predict the wave diffraction at the lip correctly. The secondary

4. RESULTS OF BENCHMARK PROBLEMS

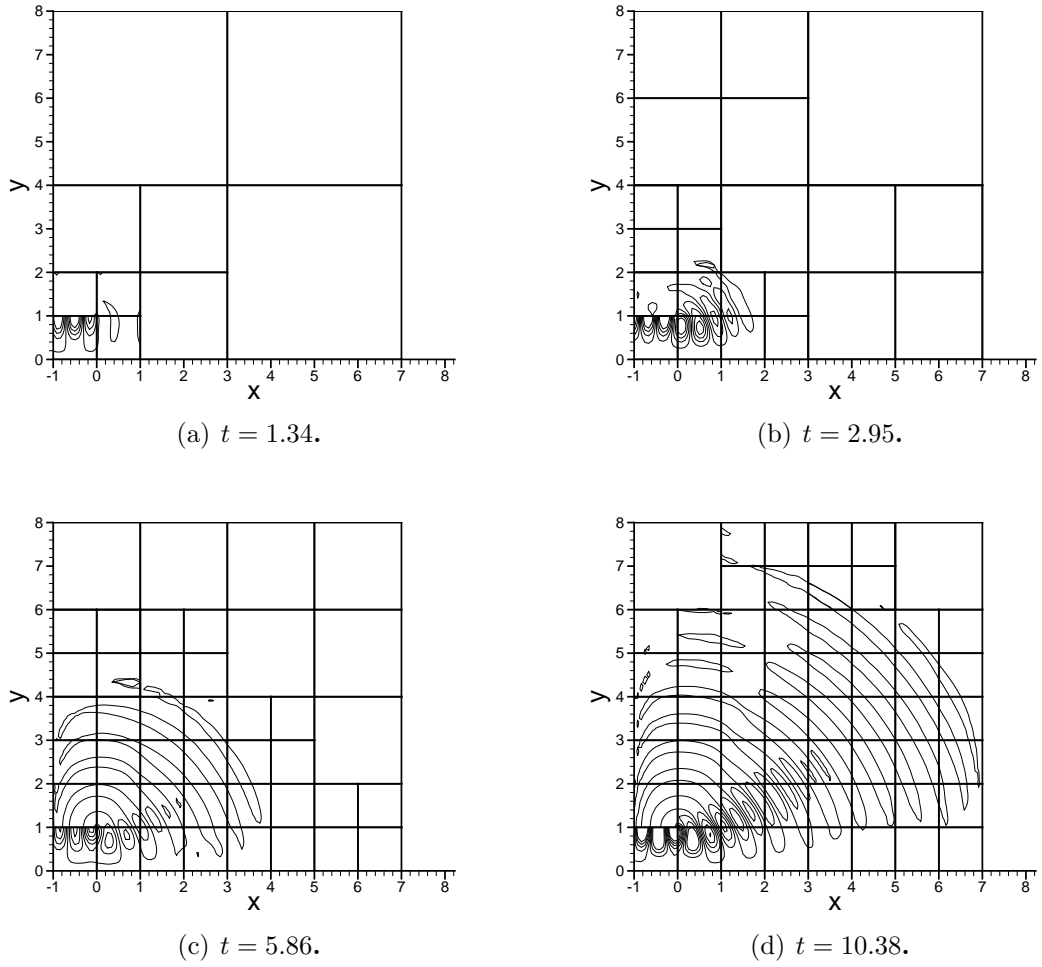


Figure 4.12: Contours of pressure perturbation of the spinning mode sound radiation from an unflanged duct on an adaptively refined mesh, where $m = 4$, $n = 1$, $k = 10$.

4. RESULTS OF BENCHMARK PROBLEMS

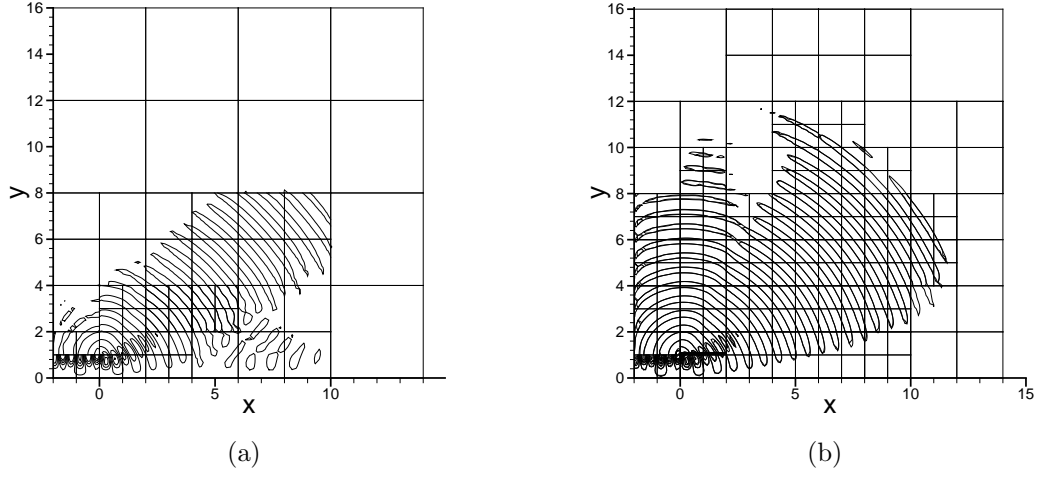


Figure 4.13: Pressure perturbation of the spinning mode sound radiation from an unflanged duct with two regridding thresholds: (a) $\tau_r = 0.15$, $\tau_c = 0.1$; (b) $\tau_r = 0.015$, $\tau_c = 0.01$, where $m = 4$, $n = 1$, $k = 10$, $t = 14.67$.

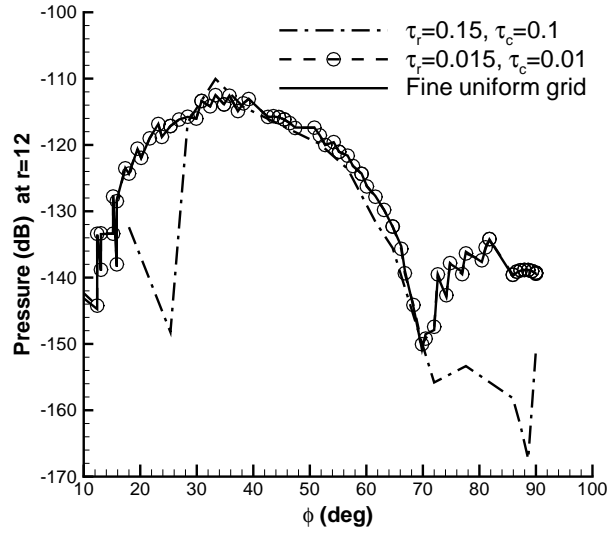


Figure 4.14: Instantaneous pressure perturbation of the spinning mode sound radiation from an unflanged duct: (---) solution computed by the big threshold; (o) solution computed by the small threshold; (—) solution computed in the finest uniform mesh, where $m = 4$, $n = 1$, $k = 10$, $t = 14.67$.

radiation peak and interference dip angle expected for this problem do not appear in the prediction. For case (c), a good agreement with the uniform mesh computation is achieved. Both the main and the secondary radiation peaks and the interference

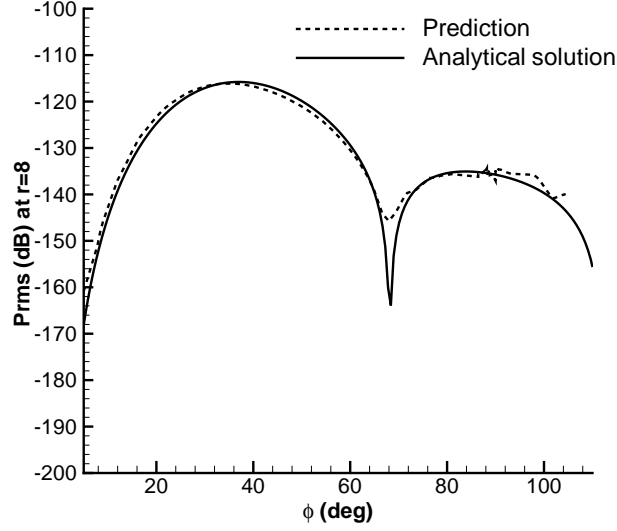


Figure 4.15: Comparison of sound directivity patterns between AMR prediction and analytical solution for the case of the spinning mode sound radiation from an unflanged duct, where $m = 4$, $n = 1$, $k = 10$.

dip angle are predicted accurately. In Figure 4.15 the predicted directivity of p'_{rms} at a nondimensional observer distance of 8 is compared with Cargill's asymptotic solution [121], which will be described in Appendix B. For most of the observer angle range, the result agrees well with the asymptotic solution.

In terms of the computational efficiency, case (c) uses nearly one-third of the time required for the uniform mesh computation. The testing environment is a 1.2 GHz Pentium III PC with 512 MBytes memory. To arrive at the far-field directivity estimation, the computation requires 2191 seconds with the finest uniform mesh, 426 seconds for case (b) and 830 seconds for case (c).

4.5 Summary

This chapter has tested several CAA benchmark problems against analytical solutions using the AMR method. In Section 4.2 a symmetrical two-dimensional acoustic propagation problem is employed to verify the working of the AMR method, where parallel performance is also profiled. In Section 4.3 the immersed boundary method is applied to an acoustic scattering problem working with the AMR method. The evident difference between AMR results and analytical solutions implied that the low order immersed boundary method is not suitable for CAA problems discussed in this

4. RESULTS OF BENCHMARK PROBLEMS

work. To solve the problem, a body-fitted multi-block AMR method will be developed in the next chapter. In Section 4.4 a more complex case, spinning mode sound radiation from an unflanged duct, is solved with the AMR method. The particular techniques that will be employed in the following chapters are applied in this case study firstly. Other than that, it also shows that there are tradeoffs between the accuracy and efficiency in an AMR computation, implying the importance of setting the regridding parameters appropriately. These case studies validate the successful working of the AMR method in solving benchmark problems with simple geometries. It will be applied to practical problems in the forthcoming chapters.

Chapter 5

Acoustic Radiation from Engine Intake Duct

In the earlier chapter the AMR method was verified and validated. From now on the main attention is focused on the particular problem of spinning mode sound radiation from a generic aero-engine intake duct. This chapter begins with the introduction of a physical model of an aero-engine intake duct. The method of AMR is then extended to solving cases with a general geometry, around which the radiation of realistic acoustic modes generated by the engine fan and fan-stator flow interactions is calculated. Combining the 2.5D LEE and the FW-H equation together, the problem of axisymmetric spinning mode sound radiation from the aero-engine intake duct is solved. The far-field solution is compared with the result obtained with an established finite element method (FEM) solver to validate the presented AMR approach.

5.1 Introduction

Stringent noise regulation requirements for modern aircraft have promoted research into efficient and accurate numerical methods capable of predicting aircraft noise. A simple sketch of an aero-engine is plotted in Figure 5.1, where spinning mode sound radiation from the intake duct is a major concern. The physical process of acoustic generation and radiation is governed by the Navier-Stokes equations. At present, a full numerical solution of acoustic generation, propagation and radiation process using the Navier-Stokes equations is not feasible. However, certain aspects of the acoustic propagation and radiation process can be modelled by linearised equations. For example, in the duct upstream of the rotor-stator region of an aero-engine, where nonlinear and viscous noise generation effects are minimal, the propagation of the

5. ACOUSTIC RADIATION FROM ENGINE INTAKE DUCT

rotor-stator noise can be studied using the inviscid linearised equations about the mean flow. A significant amount of research has been undertaken to develop theoretical and computational methods to predict engine tone noise propagation and radiation. However, the development of a cheap and quick computational method is still a challenging job. Of the three main numerical approaches for engine duct noise propagation and radiation problems, boundary element methods (BEM) [150] are confined to problems of acoustic noise through uniform mean flows; finite/infinite element (FE/IE) methods [151] are generally restricted to acoustic propagation through irrotational mean flows; and computational aeroacoustic methods based upon the Euler equations or LEE are more general in terms of governing physics [125].

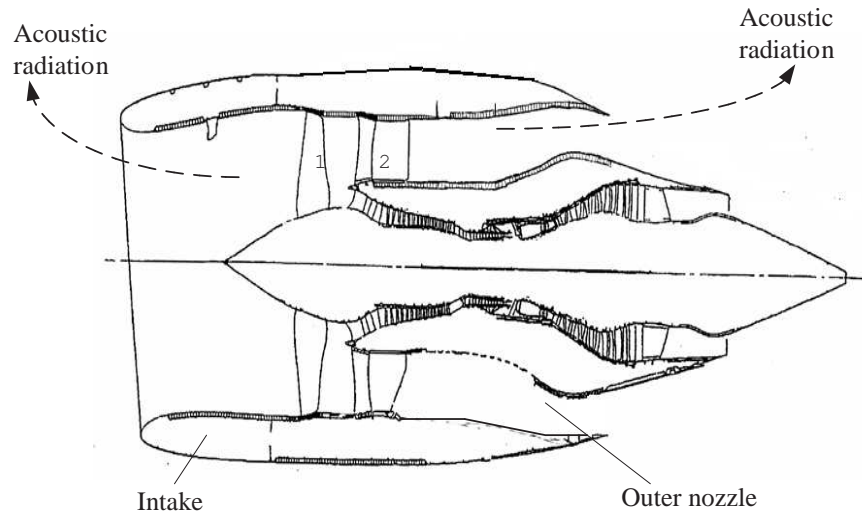


Figure 5.1: Schematic of noise radiation off an aero-engine bypass duct, where: 1 is rotor and 2 is stator.

Computational aeroacoustic methods are generally more expensive. Hence the AMR method is applied here with the aim of reducing the computational cost. In order to solve aeroacoustic problems of practical significance, *e.g.* acoustic radiation from a general aero-engine intake, the previous mentioned AMR method is extended to support body-fitted meshes. The calculation is based upon the radiation of realistic acoustic modes generated by the engine fan and fan/stator flow interactions [120].

5.2 Problem Setup

5.2.1 Numerical Issues

The acoustic radiation of this case is described by LEE (Eq. (3.7)). The computational schemes applied here include ingredients such as high-order spatial stencils, temporal schemes, inflow/outflow and surface conditions. More specifically, a 4th-order explicit scheme [42] is used to compute the spatial differences, a 4th-order low-dissipation and low-dispersion Runge-Kutta scheme [111] is used for the time integration and an explicit form of buffer zone techniques [152] is used as the outflow condition. A 10th-order filter [143] is applied throughout the domain to remove spurious waves. In the ghost construction operation, a 4th-order interpolation is used. The interval of mesh regridding is 5 computing steps, which is set according to the temporal step and the length of the ghost area, to guarantee that the wave propagation is always contained in the finest mesh.

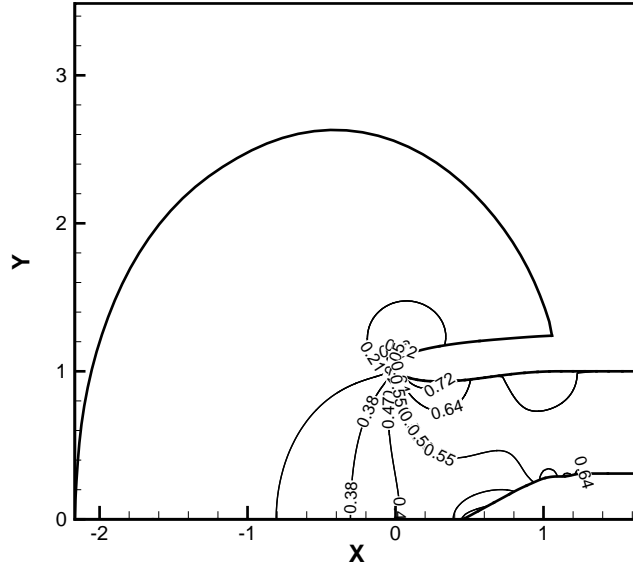


Figure 5.2: Mach number contours of the mean flow around an aero-engine intake: free stream Mach number is 0.25; ambient pressure is 94250 Pa; intake Mach number is set to 0.55 and intake pressure is 79687 Pa.

In the previous chapter the method of AMR was used in the computation of acoustic radiation along and away from an unflanged cylindrical duct. Here the method is extended to a generic aero-engine intake with a realistic background mean

5. ACOUSTIC RADIATION FROM ENGINE INTAKE DUCT

flow. An example mean flow field is shown in Figure 5.2, where a high engine power setting is used. This flow field is referred to as sideline condition in the rest of this work. In addition, the mesh is generated by the Gridgen software and the mean flow field is computed by the Fluent software.

Inflow conditions are set according to Eq. (3.8). The basic parameters of several case studies solved in this chapter are summarised in Table 5.1, where one case has sideline condition and others have stationary medium.

Table 5.1: Summary of the incoming waves for the case of an intake duct.

m	n	$f(Hz)$	k	M_j	k_r	k_a
4	1	1082.3	20	0	5.31	-19.28
12	1	1082.3	20	0	13.88	-14.40
12	2	1082.3	20	0	18.75	-6.97
13	1	903.7	16.7	0.57	14.93	-30.91
26	1	2267.3	41.9	0	28.42	-30.79

5.2.2 Curvilinear Coordinate System

In the last chapter it was shown that a Cartesian mesh with low-order immersed boundary method [91] performed more poorly than a body-fitted mesh does in solving acoustic propagation problems with curved geometries. To improve the accuracy there were attempts of using AMR for body-fitted multi-block meshes [12, 62], where curved geometries were allowed to be transformed into a uniform computational domain. It is achieved by using the coordinate transformation given by Eqs. (5.1)–(5.3), which represent a transformation from the physical to the computational coordinates. For simplicity only equations for two-dimensional problems are given and the time variance of both coordinate systems is not considered.

$$\xi = \xi(x, r), \quad \eta = \eta(x, r). \quad (5.1)$$

The first order spatial derivatives of the governing equations are evaluated using the chain rule:

$$\begin{aligned} \frac{\partial}{\partial x} &= \frac{\partial \xi}{\partial x} \frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x} \frac{\partial}{\partial \eta}, \\ \frac{\partial}{\partial r} &= \frac{\partial \xi}{\partial r} \frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial r} \frac{\partial}{\partial \eta}, \end{aligned} \quad (5.2)$$

with the transformation metrics defined as

$$\begin{aligned}\frac{\partial \xi}{\partial x} &= J \left(\frac{\partial r}{\partial \eta} \right), & \frac{\partial \xi}{\partial r} &= J \left(-\frac{\partial x}{\partial \eta} \right), \\ \frac{\partial \eta}{\partial x} &= J \left(-\frac{\partial r}{\partial \xi} \right), & \frac{\partial \eta}{\partial r} &= J \left(\frac{\partial x}{\partial \xi} \right).\end{aligned}\tag{5.3}$$

J is the transformation Jacobian relating the geometric properties of the physical space to the uniform computational space and is given by

$$J = \left[\frac{\partial x}{\partial \xi} \frac{\partial r}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial r}{\partial \xi} \right]^{-1}.\tag{5.4}$$

5.2.3 Far-field Directivity Prediction Method

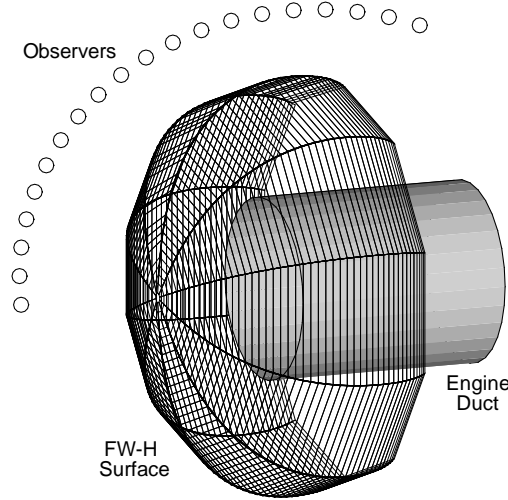


Figure 5.3: Three-dimensional FW–H integral surface around an engine duct.

For CAA methods, a finite computational domain is used so that a radiation model and a non-reflective acoustic boundary condition are required. The former one estimates the far-field directivity and is generally in the form of an integral representation. More precisely, the FW–H equation is solved on an integral surface in the computational domain to predict the far-field directivity [118]. An established

5. ACOUSTIC RADIATION FROM ENGINE INTAKE DUCT

solver, which was implemented numerically to allow both the near- and far-field noise levels to be determined efficiently [119], is applied in this work. The integral surface configuration and sensitivity test are discussed below.

Figure 5.3 displays a three-dimensional FW–H integral surface around the engine duct. The three-dimensional solutions on the surface are extended from the two-dimensional solutions of Eq. (3.7) using Eq. (3.5), *i.e.* $p'(\theta, t) = p'(0, t - m\theta/k)$. The three-dimensional solutions are then provided to the established FW–H solver, along with the geometry information of the integral surface, to predict the far-field directivity.

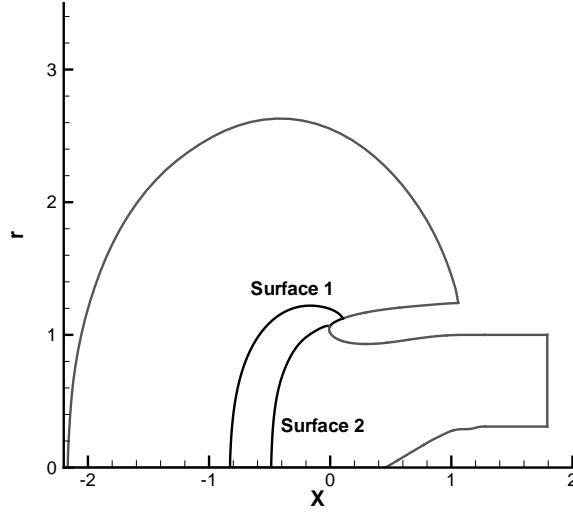


Figure 5.4: Two integral surfaces are placed in the computational domain for the sensitivity study of the FW–H solver.

Generally the placement of the integral surface and the resolution of azimuthal grid points Z_{num} will affect the accuracy of the prediction result. With respect to this intake case several numerical experiments have been done to test the sensitivity of the FW–H solver in terms of the integral surface position and the grid resolution. As shown in Figure 5.4, two positions of integral surface are tested. The number of azimuthal grid points (Z_{num}) varies from 11 to 61. The prediction results of a spinning mode sound ($m = 4, n = 1, k = 20$) are displayed in Figure 5.5. It shows that for a smaller number of Z_{num} , *e.g.* 11, the results of directivity computed on both integral surface are not consistent. In contrast, if the resolution of Z_{num} satisfies $Z_{num} \geq 10 \times m$, the directivities agree well with each other, no matter which integral surface is used. The results suggest either integral surface can be used if a

5. ACOUSTIC RADIATION FROM ENGINE INTAKE DUCT

sufficient grid resolution is selected. Therefore, in the following studies the resolution of $Z_{num} = 10 \times m$ and integral surface 1 are used to predict the far-field solutions.

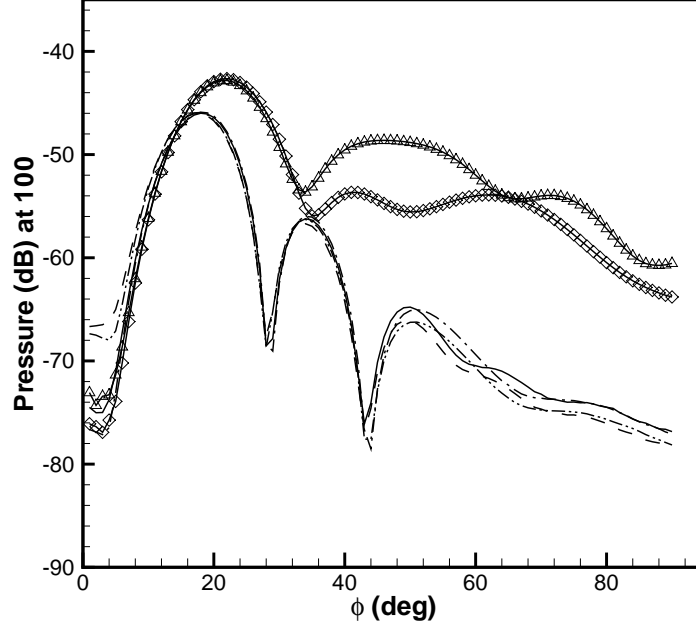


Figure 5.5: Directivity plot for different FW-H surface placements and azimuthal grids number (Z_{num}) in the case of the general intake: (\triangle) surface 1 and $Z_{num} = 11$; (-) surface 1 and $Z_{num} = 41$; (- -) surface 1 and $Z_{num} = 61$; (\diamond) surface 2 and $Z_{num} = 11$; (-.-) surface 2 and $Z_{num} = 41$; (-.-.-) surface 2 and $Z_{num} = 61$, where $m = 4$, $n = 1$, $k = 20$ with the stationary mean flow.

5.2.4 Absorbing Numerical Noise

By using ϵ -pseudospectra analysis [141], it was found that a high-order, *e.g.* 4th-order, spatial scheme is more susceptible to numerical errors than its low-order counterpart on an adaptively refined mesh. Either a 10th-order explicit filter [101] or an artificial selective damping [110] is used in this study to absorb high-frequency numerical nuisance. The effects of filter and damping methods are compared in Figure 5.6, where an AMR mesh with two refinement levels is adaptively refined to capture a spinning mode sound radiation from the general aero-engine intake with $m = 12$, $n = 1$, $k = 20$. In the first experiment with the damping technique, the damping coefficients that were used in a cavity flow simulation [110] are used here directly. Figures 5.7(a)-5.7(b) show the damping method works fine when sound wave propagates inside the

5. ACOUSTIC RADIATION FROM ENGINE INTAKE DUCT

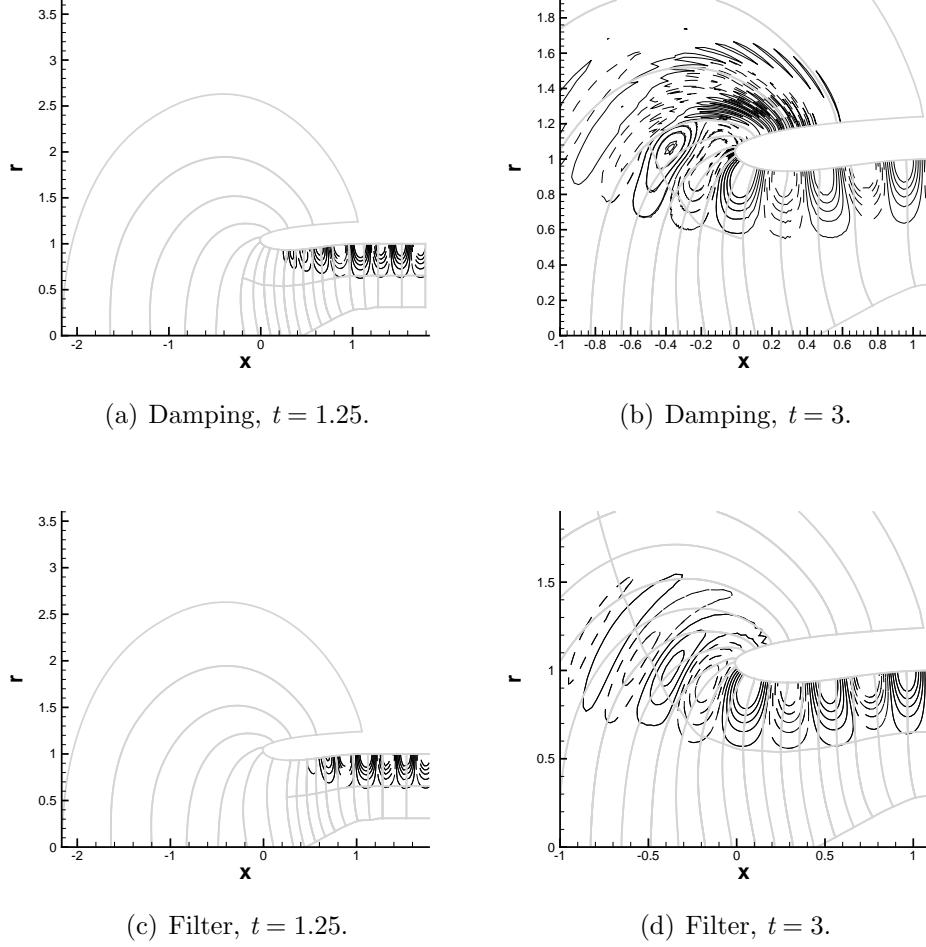


Figure 5.6: Effect of spurious wave treatment methods for a spinning mode sound radiation from an aero-engine intake duct, where perturbation pressure contours are displayed around the lip of the intake, gray lines denote the boundaries of blocks on the adaptively refined mesh; $m = 12$, $n = 1$, $k = 20$.

intake, whereas it fails to absorb numerical noise generated around the lip of the intake. It reveals the fact that the coefficients of artificial selective damping have to be adjusted case by case. By contrast, the filter technique is not problem specific and more general. It is applied in the second experiment. Figures 5.6(c)-5.6(d) indicate that the filter method removes spurious waves effectively, although above the lip there is still a small wiggle which is caused by the spurious wave generated in the adaptively refinement mesh that is not fully removed by the filter. Therefore, in the rest of this chapter the 10th-order explicit filter is applied to remove the spurious waves.

5.3 Results

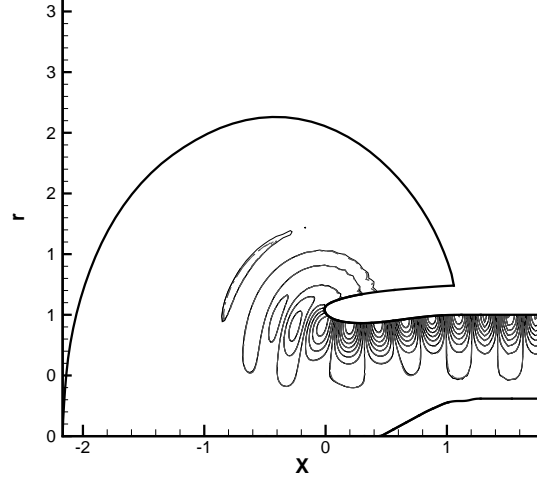
5.3.1 Near-field Propagation and Far-field Directivity

With the aforementioned techniques, the acoustic propagation in and radiation from the aero-engine intake is solved with AMR. In the computation, typically two to three refinement levels are used. Once the waves reach the outflow boundary of the computation domain, the finest blocks span the whole computational domain and the regridding operation is stopped to improve efficiency. Two background mean flow configurations have been used. One is the stationary medium and the other is the sideline mean flow.

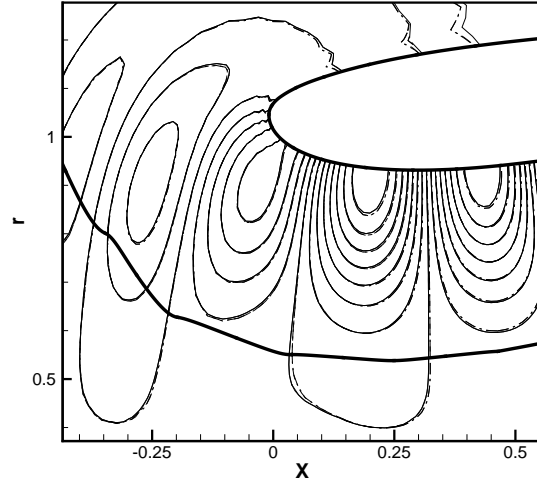
Firstly, a comparison is made between the result of AMR and the result computed on a uniformly fine mesh. The mean flow is the stationary medium. The near-field solutions of the whole domain are compared in Figure 5.7, which indicates that both meshes generate similar wave patterns, validating the working of the AMR method for this study. More precisely, a comparison is made in Figure 5.8, where the instantaneous pressure perturbations agree well with each other on a selected line (as shown in Figure 5.7(b)). It confirms the working of the proposed AMR method.

Secondly, Figures 5.9-5.10 compare the result of instantaneous perturbation pressures computed on an adaptively refined mesh with the previous result computed on a fixed uniformly fine mesh that was presented in Richard's work, using a high-order CAA scheme [153]. Once again, there are little differences between these two results in the case of the stationary medium, whereas for the sideline mean flow case the radiation pattern is slightly influenced by the AMR method. In order to show the differences much more clearly, the far-field directivity results computed with various strategies are compared in Figure 5.11. The integration surfaces used in this case study are displayed in Figures 5.9-5.10.

In the case with the stationary medium, an adaptively refined mesh with two refinement levels is set up. The filter method is applied to absorb the spurious waves at the coarse and fine block interfaces. It appears that both the peak level and the peak radiation angle agree well with the results of Richards [153]. The peak radiation is predicted at 47.0deg (Figure 5.11(a)). This compares well with the prediction (47.27deg) of Richards's work [153]. The dynamic range of the prediction is typically higher than 60dB which is good enough for most of the engineering applications. Using the filter alone in the computation, the peak radiation level is 0.55dB lower than the result of Richard's work. It reflects the fact that the filter introduces an excessive level of dissipation surrounding the lip of the aero-engine intake. Meanwhile, the prediction does not follow the decaying envelope at low observation angles to the



(a)



(b)

Figure 5.7: The pressure perturbation contours around the aero-engine intake duct, computed on either an adaptively refined mesh or a uniformly fine mesh, where: (a) shows the whole domain; (b) is an enlarged part; (—) the AMR result; (---) the result computed on the uniform mesh, $m = 12$, $n = 1$, $k = 20$, $t = 2.8$. The bottom bold line displayed in (b) is for Figure 5.8.

axisymmetrical axis ($\phi \leq 25$ deg). The dynamic range of the prediction is somewhat smaller than the prediction of Richard. This is as expected as the order of the spatial

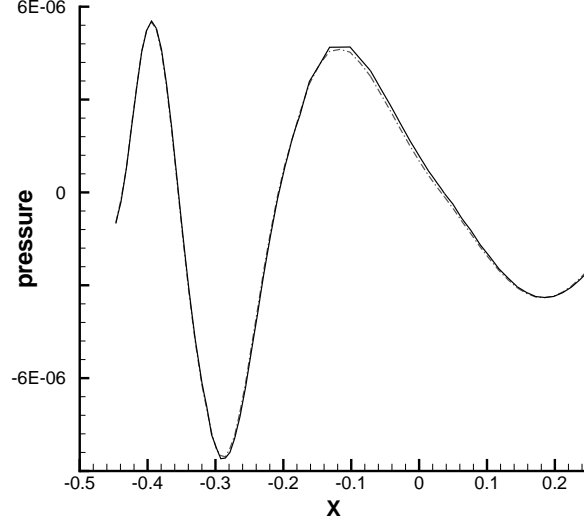
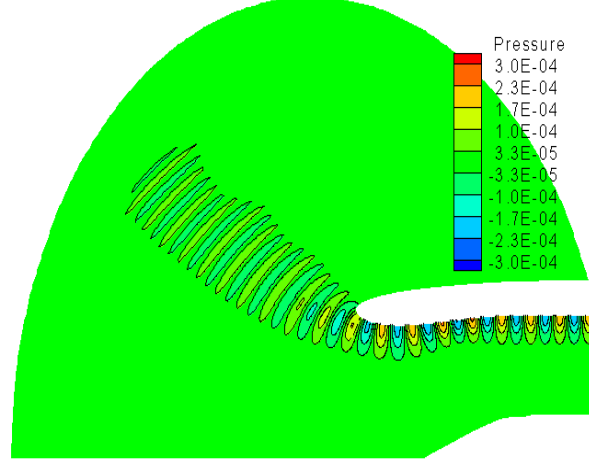


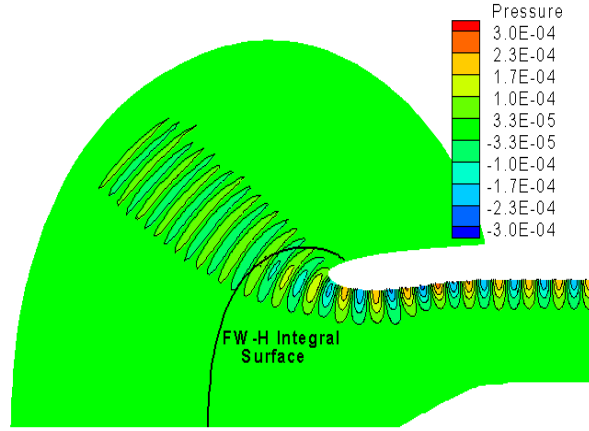
Figure 5.8: Comparing instantaneous pressure perturbations on a selected line (as shown in Figure 5.7(b)) for the intake case, where: (—) denotes the AMR result; (—) is the result on a uniformly fine mesh.

scheme on an adaptively refined mesh is demonstrated earlier to be less than 4. This particular feature might also be influenced by the spurious waves generated at the fine-coarse interfaces in the AMR operations. The accuracy suffers slightly as the observation angle approaches 120deg, the discrepancy in pressure level being at most 2.2dB.

For the sideline case, the results are presented in Figure 5.11(b). The main peak angle and the peak level of the AMR result match the other two solutions well. The peak radiation angle is at 59.9deg (Figure 5.11(b)) which compares favourably with 59.4deg predicted by Richard who solved LEE on a uniformly fine mesh and 60.8deg by using an established FEM solver [153]. The differences of the peak radiation level between these results are less than 0.5dB, whilst the peak radiation angles differ from each other by less than 0.7deg. Towards the low observation angle range ($\phi \leq 22$ deg), the discrepancy in the pressure level increases again. The reason of this feature is the same as that explained in the previous case. The dynamic range of the prediction is also about 60dB. Nevertheless, the prediction deteriorates toward the high observation angles, especially around the shadow interference dip angles at 88.3deg, where there is 7dB difference between the AMR and the FEM results and 5dB between the AMR and Richard's predictions on a fixed mesh. It could be caused by the spurious wave generated above the lip of the intake, as indicated by the wiggles



(a)



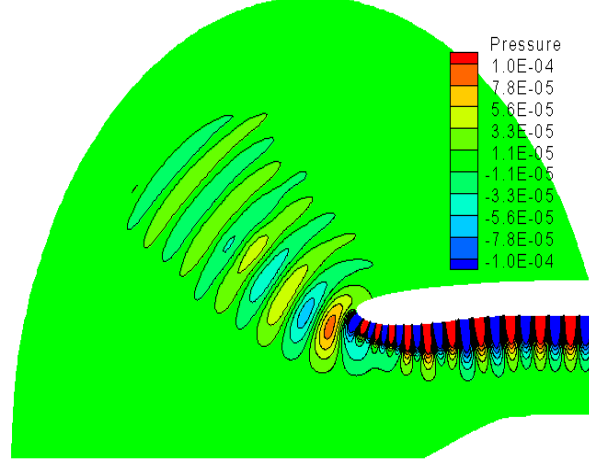
(b)

Figure 5.9: Perturbation pressure contours around the aero-engine intake duct, where $m = 26$, $n = 1$, $k = 41.9$, with the stationary medium and (a) LEE on a fixed mesh [153]; (b) LEE on an adaptively refined mesh.

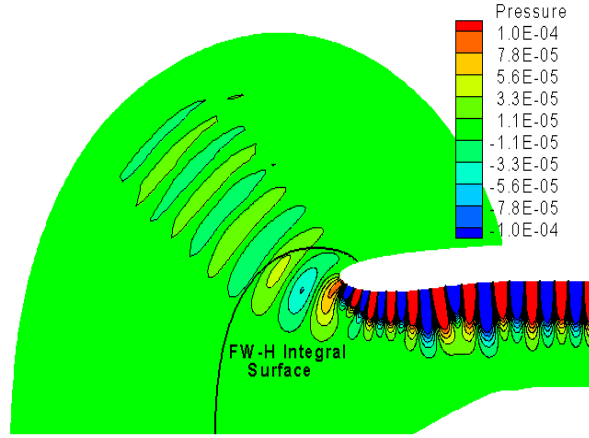
shown in the perturbation pressure contours (Figure 5.6(d)).

In the AMR computation of the stationary medium case, the total number of cells increases from 13872 to 41616 as the wave propagates out of the intake. The

5. ACOUSTIC RADIATION FROM ENGINE INTAKE DUCT



(a)

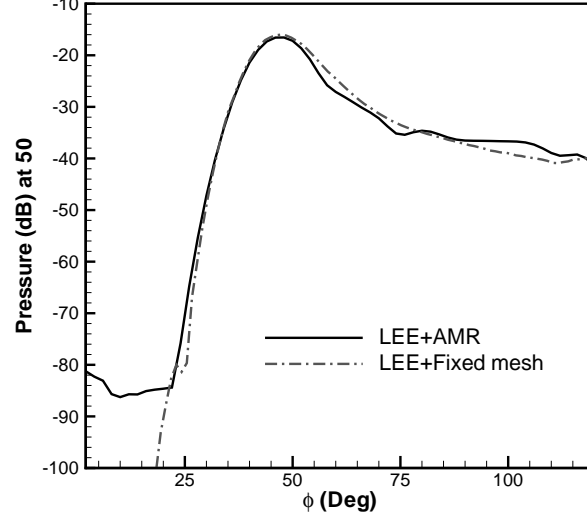


(b)

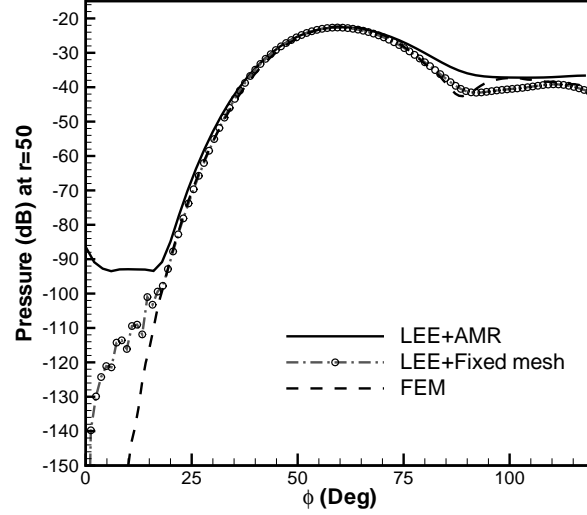
Figure 5.10: Perturbation pressure contours, where $m = 13$, $n = 1$, $k = 16.7$, with the sideline mean flow and (a) LEE on a fixed mesh [153]; (b) LEE on an adaptively refined mesh.

computing time is 3463 seconds on a desktop computer (Pentium IV 1.3GHz, 768MB). In contrast, the computing time is 5400 seconds with the same AMR code working on a uniformly fine mesh without running the regridding operation. In an earlier

5. ACOUSTIC RADIATION FROM ENGINE INTAKE DUCT



(a)



(b)

Figure 5.11: Far-field directivities for the aero-engine intake radiation. (a): $m = 26$, $n = 1$, $k = 41.9$ with the stationary medium and (b) $m = 13$, $n = 1$, $k = 16.7$ with the sideline mean flow.

computation with the SotonLEE code which also solves LEE [7], 7560 seconds is required to achieve the same results on a uniform mesh consisting of 81600 cells. It suggests that on a uniform mesh, the efficiency of the AMR code is around 26%

lower than the efficiency of the SotonLEE code due to the introduction of the AMR data structure management and the additional AMR operations, such as the ghost construction operation. In the next section the parallel performance of the AMR code for the case studies is discussed in more detail.

5.3.2 Parallel Performance

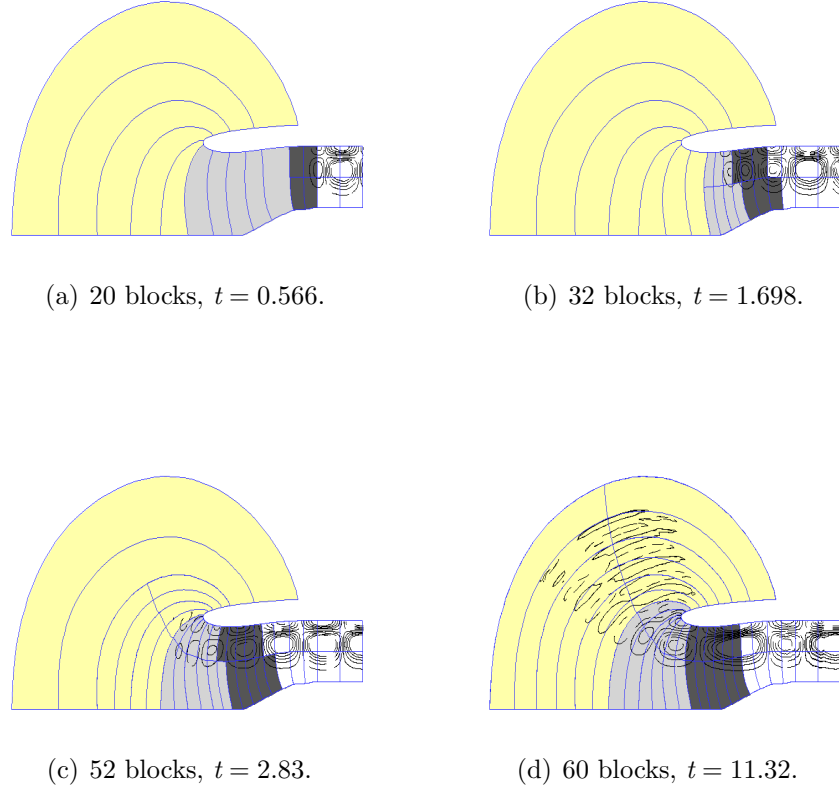


Figure 5.12: Perturbation pressure contours, $m = 12$, $n = 2$, $k = 20$. The computing load is distributed over 4 processors, which are denoted by different coloured areas.

Figure 5.12 illustrates the process of regridding and dynamic load balancing within four processors which are represented by different coloured areas. It can be seen that the computational load is redistributed evenly among processors along with the sound propagation and radiation. In Figure 5.13 the parallel speedup performance of the AMR code is compared with the parallel speedup performance of the SotonLEE code

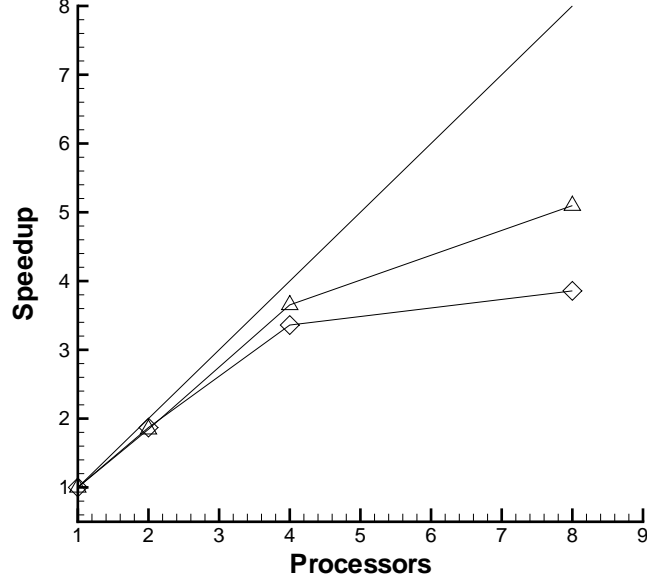


Figure 5.13: The parallel speedup of the case of spinning mode sound radiation from an aero-engine intake with the sideline condition: (—) ideal speedup; (\triangle) the result of the SotonLEE code; (\diamond) the result of the AMR code.

on a Beowulf PC cluster that consists of twenty-eight processors distributing on seven nodes connected by a Gigabit Ethernet. It is discovered that the communication cost of the AMR code is generally one to three times higher than the communication cost of the SotonLEE code. The cost is mainly contributed by the expensive communication cost of the AMR ghost construction operation. That operation consists of a lot of memory movements and network communications. Its performance is then limited by the present memory and network technology. Therefore, the speedup performance of the AMR method deteriorates slightly along with the increase of the communications.

More precisely, the performance of the computation and communication costs with respect to both mean flow configurations (stationary medium and sideline condition) are recorded by the Complete System Performance Monitor [154], a graphical tool monitoring the system performance under Linux environment. Table 5.2 presents the results working on one to eight processors. When the processor number is less than eight, each node will have one working AMR process. Otherwise, the first node will have two working AMR process. To save space, only information on the first processor on the PC cluster is given. It is indicated that the communication costs for both case studies are roughly the same for the reason that the transported flow

5. ACOUSTIC RADIATION FROM ENGINE INTAKE DUCT

solutions, *e.g.* (ρ', u', v', w', p') , are of the same size. In addition, both communication costs grow slightly as the processors number increases from two to four. Furthermore, when the processors number reaches eight, there are two AMR processes working on the first node coincidentally so that the communication cost of the first node jumps abruptly accordingly. It subsequently affects the parallel speedup performance. In the meantime, the computation costs in Table 5.2 reveal that the case study with the stationary mean flow is simpler than the sideline case in terms of computational complexity. However, both case studies with different mean flow fields, as mentioned before, have similar communication costs. For that reason the parallel speedup of the first case study with the stationary mean flow is worse than the second case study with the sideline condition.

Table 5.2: The parallel communication and computation costs of the AMR computation of the aero-engine intake radiation.

	1CPU	2CPUs	4CPUs	8CPUs
Communication costs of				
Stationary mean flow	N.A.	7.06MB	8.90MB	27.10MB
Sideline mean flow	N.A.	7.01MB	7.68MB	27.05MB
Computation costs of				
Stationary mean flow	610s	500s	469s	477s
Sideline mean flow	1040s	555s	310s	270s

5.4 Summary

In this chapter the AMR method is applied to the prediction of spinning mode sound radiation from a generic engine intake. To model curved geometries, the AMR code is extended to support body-fitted grids. Both the explicit filter and artificial selective damping methods are applied to absorb spurious waves generated in the AMR computation. Their effects are compared and the filter method is shown to be the preferred method for the problem studied here. The accuracy of the AMR method is demonstrated by the predicted far-field directivity, which agrees well with the LEE result computed on a uniformly fine mesh and the FEM result. In terms of computation efficiency, the adaptively refined mesh represents a saving of up to 40% compared with a uniform mesh. Relied on MPI, the computation loads are shown to be distributed evenly within the processors by MPI. However, the relevant communication cost increases along with the increase of the number of processors. The

5. ACOUSTIC RADIATION FROM ENGINE INTAKE DUCT

parallel speedup performance deteriorates accordingly. In order to attain a higher efficiency on the current parallel machines, it is suggested to separate the parallel communication of the ghost construction operation from the other AMR operations to obtain an optimal performance in the future work.

Chapter 6

Acoustic Radiation from Engine Exhaust Duct

This chapter studies a spinning mode sound radiation from a generic aero-engine exhaust duct. The governing equations used here are extended acoustic perturbation equations (APE). The reason of using APE rather than the original LEE is explained in the second section along with the associated numerical issues. The AMR method is still applied in the computation. The associated mesh adaption procedure is displayed and discussed in the third section. The near- and far-field solutions are then presented. Finally, a summary is given.

6.1 Introduction

In the case of radiation from either a bypass exhaust duct or a core nozzle, there are issues associated with the presence of a background mean flow with a shear layer between the exhaust flow and the external stream. Once again, Figure 5.1 displayed in the previous chapter shows the problem. Refractive effects due to the presence of a sheared flow may change noise radiation pattern. The physical process is still governed by the Navier-Stokes equations. As mentioned before, a full numerical solution of noise generation, propagation and radiation process using the Navier-Stokes equations is not feasible due to limited computational resources. However, in the duct downstream of the rotor-stator region of an aero-engine, where nonlinear and viscous noise generation effects are minimal, the propagation of the rotor-stator noise can be studied using the inviscid linearised equations about the mean flow.

Block-structured AMR has been applied to studying the radiation of spinning modes from a unflanged duct and aero-engine intake duct problems to establish far-

6. ACOUSTIC RADIATION FROM ENGINE EXHAUST DUCT

field directivities in the earlier chapters. The same work was also reported in [41, 97]. The results of AMR were verified by comparing to both analytical solutions and FEM results. In this chapter the block-structured AMR is applied to the general case of noise radiation from a realistic high bypass ratio engine exhaust geometry with a background mean flow. A computational model used to determine the propagation and radiation of acoustic waves is outlined firstly. The computational scheme described here allows acoustic waves, propagating inside the bypass duct of a generic aircraft engine, to be admitted into a computational domain that comprises the aft duct section, the exit plane of the duct and the jet flow immediately downstream. The wave admission is realised through an absorbing non-reflecting boundary treatment which admits incoming waves and damps spurious waves generated by the numerical solutions. The exhaust geometry is axisymmetric and the mean flow axisymmetric with no swirl component. The acoustic disturbances are represented by a Fourier series in the circumferential direction. Subsequently, the wave propagation and diffraction can be calculated through solutions of LEE, Eq. (3.7), using a range of high-order schemes [126].

However, hydrodynamic shear layer instabilities associated with the presence of the sheared background mean flow induce unstable solutions in the computation of LEE, corrupting the desired acoustic solutions. To stabilize the solutions, it is a common practice to remove some mean shear terms containing $\partial u_0 / \partial r$ in LEE, Eq. (3.7). The approach was validated against Munt's analytical solution of semi-infinite duct radiation problem [155] in previous works [123, 126]. Further tests against other comparable methods are necessary on realistic geometry and flow conditions.

A set of new governing equations, APE [156, 157], are used in this work as an alternative way to validate the previous approach in computing problems with a sheared mean flow. To solve the aero-engine case problem, APE have been extended to the cylindrical coordinates. In short, the solutions of APE are compared to the previous solutions of LEE [126] through a case study of single spinning mode sound radiation from a generic engine bypass duct. The far-field directivity is estimated via an integral surface solution of the FW-H equation [118]. More details are given below.

6.2 Problem Setup

6.2.1 Extended Acoustic Perturbation Equations

In the past chapters, LEE have been used to solve the problem of a spinning mode radiation from a unflanged duct or from an aero-engine intake duct. However, in the cases with a shear layer, the LEE solver also admits hydrodynamic instabilities that can overwhelm the desired acoustic solutions. In order to suppress this type of unbounded growth of instabilities, APE have been proposed to the computation of the acoustic wave convection and refraction under Cartesian coordinates. The fundamental principle is to remove flow instability from the acoustic computation by filtering the original LEE. An in-depth discussion of the relevant theoretical background can be found in the work of Ewert and Schröder [156]. To be complete, the original APE-2 system under two-dimensional Cartesian coordinates [156] is given here:

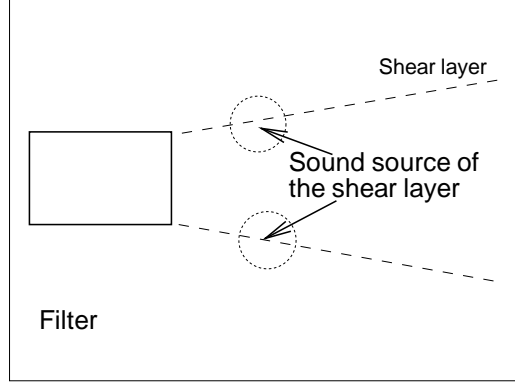
$$\begin{aligned}\frac{\partial \rho'}{\partial t} + \frac{\partial(\rho' u_0 + \rho_0 u')}{\partial x} + \frac{\partial(\rho' v_0 + \rho_0 v')}{\partial y} &= S_\rho, \\ \frac{\partial u'}{\partial t} + \frac{\partial(u_0 u' + v_0 v')}{\partial x} + \frac{\partial}{\partial x} \left(\frac{p'}{\rho_0} \right) &= S_u, \\ \frac{\partial v'}{\partial t} + \frac{\partial(u_0 u' + v_0 v')}{\partial y} + \frac{\partial}{\partial y} \left(\frac{p'}{\rho_0} \right) &= S_v,\end{aligned}\tag{6.1}$$

where the definitions of variables are the same as in Eq. (3.7), $S_{\{\rho,u,v\}}$ represents sound source generated in the shear layer, as illustrated in Figure 6.1(a)

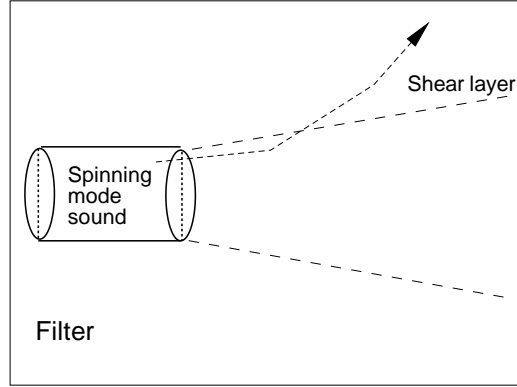
In order to extend APE to the axisymmetrical duct radiation case, several assumptions are made. The first assumption regards the sound source of $S_{\{\rho,u,v\}}$ as zero for the reason that the subject studied here is only the spinning mode sound. The similar assumption was also used in the first test case described in [156]. The second assumption is that there is no unstable flow dynamics generated in the azimuthal direction. For that reason the original LEE in the azimuthal axis is still kept. The filter removing hydrodynamic instabilities is therefore only operated in the axial-radial space, as shown in Figure 6.1(b). Other than these two assumptions, a further simplification is made to extend APE to the cylindrical coordinates. To be succinct, the following derivation takes the vector form. Some terms in Eq. (6.1) are replaced in accordance with the following derivation:

$$\nabla \left(\frac{p'}{\rho_0} \right) = \frac{\nabla p'}{\rho_0} - \frac{p'}{\rho_0^2} \nabla \rho_0 = \frac{\nabla p'}{\rho_0} - \frac{\rho' \gamma}{\rho_0^2} \nabla p_0,\tag{6.2}$$

6. ACOUSTIC RADIATION FROM ENGINE EXHAUST DUCT



(a)



(b)

Figure 6.1: The schematics of APE: (a) two-dimensional domain; (b) axisymmetrical three-dimensional domain.

where the acoustic compression is assumed to be of small amplitude and isentropic, whilst the mean flow compression is adiabatic for a perfect gas.

The final form of APE employed in this work are:

$$\begin{aligned} \frac{\partial \rho'}{\partial t} + \frac{\partial(\rho' u_0 + \rho_0 u')}{\partial x} + \frac{\partial(\rho' v_0 + \rho_0 v')}{\partial r} - \frac{m \rho_0}{k r} w'_t + \frac{\rho' v_0 + \rho_0 v'}{r} &= 0, \\ \frac{\partial u'}{\partial t} + \frac{\partial(u_0 u' + v_0 v')}{\partial x} + \frac{\partial p'}{\rho_0 \partial x} + \frac{\gamma \rho'}{\rho_0} (u_0 \frac{\partial u_0}{\partial x} + v_0 \frac{\partial u_0}{\partial r}) &= 0, \end{aligned}$$

6. ACOUSTIC RADIATION FROM ENGINE EXHAUST DUCT

$$\begin{aligned} \frac{\partial v'}{\partial t} + \frac{\partial(u_0 u' + v_0 v')}{\partial r} + \frac{\partial p'}{\rho_0 \partial r} + \frac{\gamma \rho'}{\rho_0} (u_0 \frac{\partial v_0}{\partial x} + v_0 \frac{\partial v_0}{\partial r}) &= 0, \\ \frac{\partial w'_t}{\partial t} + u_0 \frac{\partial w'_t}{\partial x} + v_0 \frac{\partial w'_t}{\partial r} + \frac{mk}{\rho_0 r} p' + \frac{w'_t v_0}{r} &= 0, \end{aligned} \quad (6.3)$$

where the first and last equations of Eq. (3.7) are kept.

6.2.2 Numerical Issues

In this case study the aforementioned 4th-order explicit schemes [42], the 4th-order 4-6 low-dissipation and low-dispersion Runge-Kutta [111] temporal scheme, the 4th-order linear interpolation and the 10th-order filter [143] are employed again. Other numerical issues associated with the AMR method on CAA applications have been addressed in the foregoing chapters.

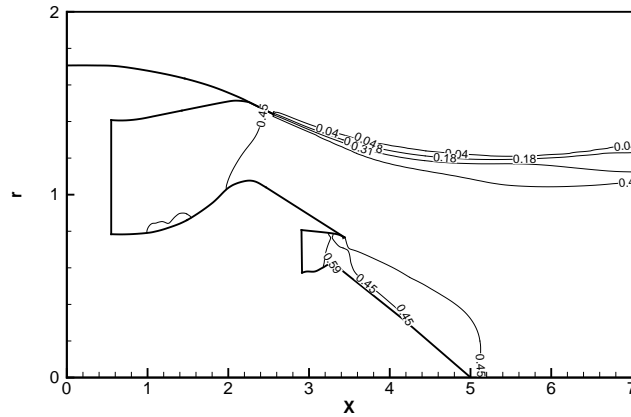


Figure 6.2: Mean Mach number distribution of the aero-engine exhaust test case.

The basic problem is illustrated in Figures 6.2-6.3 that show the computational domain on which the near-field CAA propagation calculation is performed. The specific configuration resembles the previous effort [126]. The illustrated background mean flow is in terms of Mach number, which is set to 0.338 at the inflow boundary inside both the exhaust duct and the core nozzle. The exhaust stream is issued into a stationary environment. Inside the exhaust duct, a buffer zone [152] is used to absorb the reflective spurious waves as well as to accommodate the incoming modal waves, which are of the form given by Eq. (3.8), where M_j is nondimensional velocity inside the duct; a is fixed at 10^{-4} again to ensure small relative changes in density (as required by both LEE and APE). Other related parameters have been introduced

6. ACOUSTIC RADIATION FROM ENGINE EXHAUST DUCT

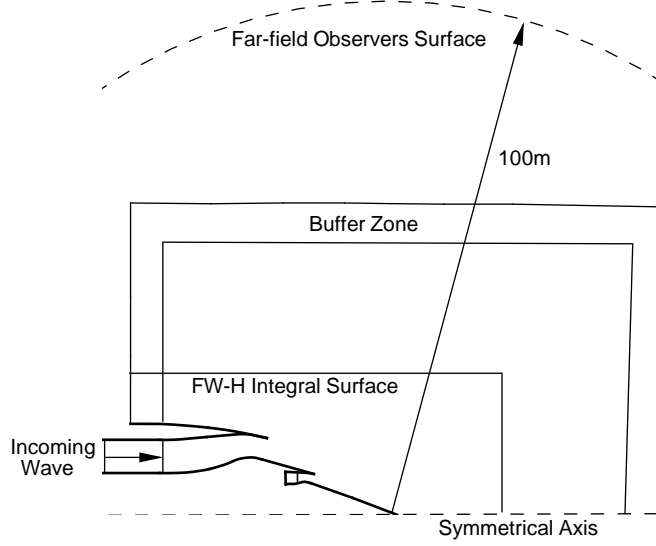


Figure 6.3: The problem setup of the aero-engine exhaust geometry that is displayed with thick lines.

before and their definitions are remained here. In this case problem, four radial modes are solved in the incoming waves. They are summarised in Table 6.1.

Table 6.1: Summary of the incoming spinning mode waves for the exhaust duct radiation.

$n(m = 13)$	$f(Hz)$	k	M_j	k_r	k_a
1	1562.7	28.3179	0.338	10.60	19.11
2	1562.7	28.3179	0.338	14.01	17.49
3	1562.7	28.3179	0.338	16.50	15.93
4	1562.7	28.3179	0.338	19.71	13.35

A buffer zone is also placed around the outer boundaries of the domain and inside the core exhaust nozzle. The target solutions of this buffer zone is set to zero to absorb spurious numerical reflections.

The far-field directivity is estimated through an integral solution of the FW–H equation [118]. For simplicity the FW–H equation integral surface shown in Figure 6.3 is located at the borders of blocks surrounding the engine exhaust. The procedure of its three-dimensional extension has been presented in the previous chapter. The far-field observers are located at the nondimensional length of 100 from the conical

rear of the exhaust geometry.

6.3 Results

6.3.1 AMR Working Procedure

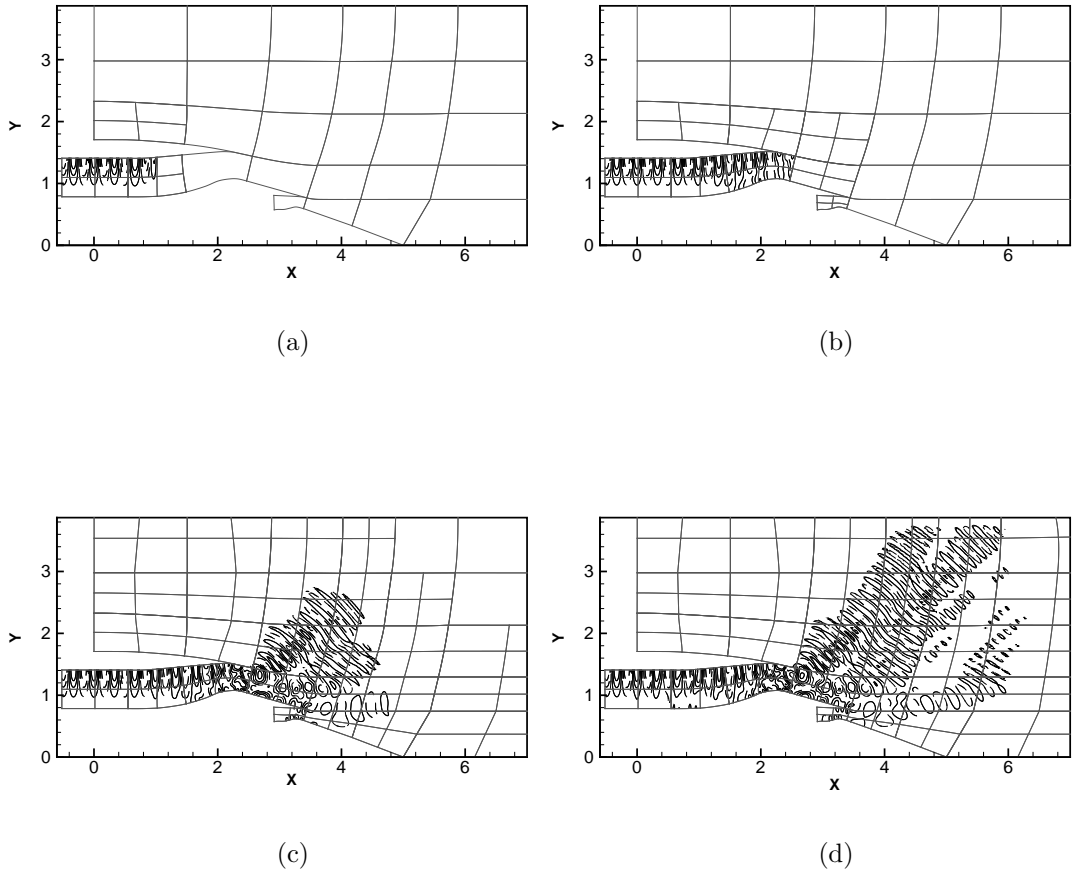


Figure 6.4: The evolution of the adaptively refined mesh with the propagation of acoustic waves from the engine exhaust, where gray lines represent the block boundaries of the adaptively refined mesh, (a) $t=1$, (b) $t=2$, (c) $t=4$, and (d) $t=6$.

AMR could provide higher computational efficiency and more flexibility than a uniform mesh. Figure 6.4 illustrates the procedure of the adaptively refined mesh as the acoustic waves propagate and radiate out of the engine exhaust. The outer buffer zone is not displayed fully to save space. In the whole procedure, the total number of grid points increases from 36,000 to 180,000.

6. ACOUSTIC RADIATION FROM ENGINE EXHAUST DUCT

The computation is executed and tested on a computer with a Pentium IV 3.0GHz CPU and 2GBytes memory. Table 6.2 shows that the computation time of AMR is increased along with the increase of grid points. In the initial stage (*i.e.* $t < 5$) the computation time of AMR is around 100% faster than the computation time on the uniform mesh. After that, the computational efficiency of AMR gradually decreases. Finally it reaches the same level of the computational efficiency on the uniform mesh, due to the extended span of the acoustic wave in the whole computational domain (see Figure 6.4(d)), where the adaptively regridding operation is no use anymore.

Table 6.2: Computation time comparison for the exhaust duct radiation.

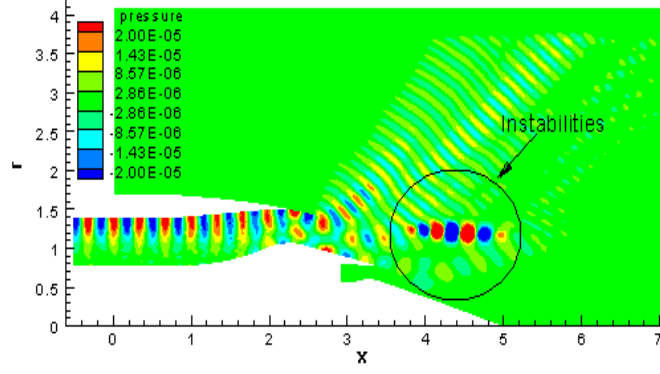
Grids	$t = 1$	$t = 2$	$t = 5$	$t = 8$
AMR	1478s	3401s	13460s	26670s
Uniform mesh	3971s	8002s	20120s	31920s

6.3.2 Near-Field Propagation

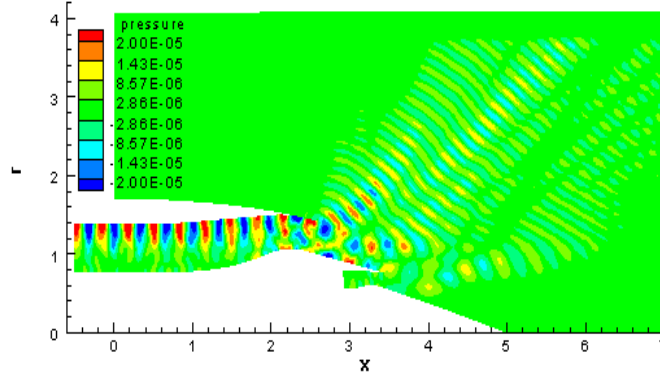
Figure 6.5 compares the near-field wave propagation predicted by either LEE or APE. Two refinement levels are used. The coarse level mesh consists of 36,000 grid points, whilst the fine level mesh is adaptively updated and the number of grid points varies accordingly. In this case, after $t = 12$, hydrodynamic instabilities developed in the shear layer are evident with the LEE method (see Figure 6.5(a)), where the original LEE (Eq. (3.7)) are used and terms containing $\partial u_0 / \partial r$ are still kept to induce instabilities. These instability waves will develop to overwhelm the desired acoustic solutions completely. The APE method, Eq. (6.3), are also applied to the case. Figure 6.5(b) shows perturbation pressure contours computed by the APE method. It indicates that the numerical instabilities are avoided, whereas the near-field propagation pattern retains the same key features as the LEE solutions: wave diffraction off the lip of the bypass duct and reflections off the surface of the afterbody of the engine exhaust. It should be noted that, for the present test case computation, the mean flow conditions in the core nozzle are the same as those in the bypass duct. Upstream traveling waves now appear inside the core nozzle and are visible in Figure 6.5(b).

Figure 6.6 shows the near-field sound pressure level, $SPL = 20\log_{10}(p'_{rms}/(2 \times 10^{-5}))$, where the selected time to compute p'_{rms} satisfies $t < 12$, in which the hydrodynamic instabilities appeared in the LEE computation still do not overwhelm the acoustic solutions. It shows that propagation patterns predicted by both methods

6. ACOUSTIC RADIATION FROM ENGINE EXHAUST DUCT



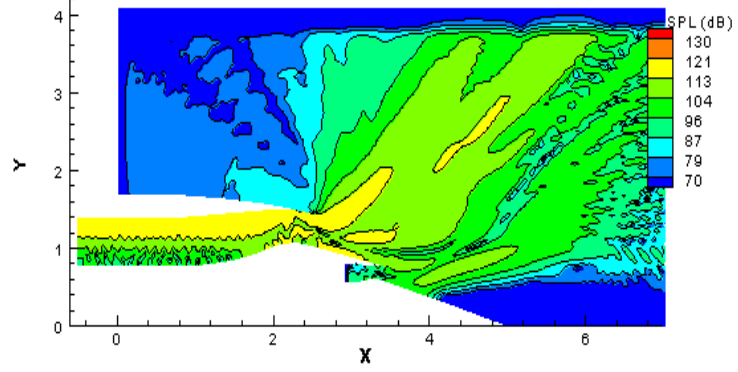
(a)



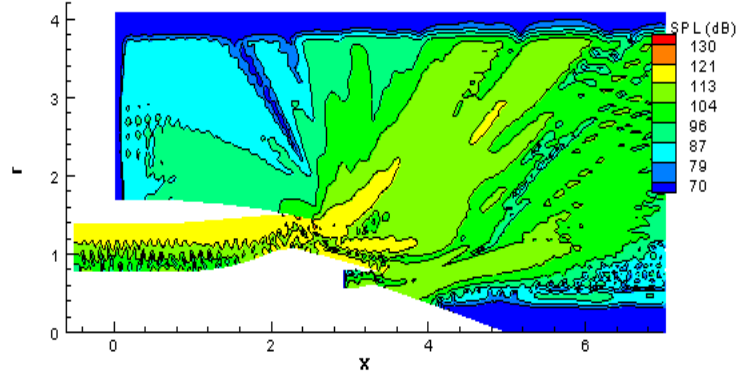
(b)

Figure 6.5: Perturbation pressure contours computed by LEE and APE, where $m=13$, $n=1$, $k=28.3179$. (a) LEE with AMR, $t=12$, (b) APE with AMR, $t=20$.

agree well in most parts, whereas the sound pressure level of the APE solution is a bit higher at high and low angles than the LEE prediction. By using the APE method, several other spinning mode waves ($n = 2 - 4$) are solved as well. Figure 6.7 displays the results of the perturbation pressures and sound pressure level contours.



(a)



(b)

Figure 6.6: SPL contours computed by LEE and APE, where $m=13$, $n=1$, $k=28.3179$, (a) LEE, $9.5 < t < 10$, (b) APE, $15 < t < 15.5$.

6.3.3 Far-Field Directivity

Through an integral surface solution of the FW-H equation, the far-field directivities of the four spinning mode radiation are predicted based on the near-field solutions of APE. The outcomes are compared with the LEE prediction [126] in Figure 6.8. To avoid the potential effect of the hydrodynamic instabilities in the computation with LEE, the terms containing $\partial u_0 / \partial r$ were omitted in the governing equations [158].

6. ACOUSTIC RADIATION FROM ENGINE EXHAUST DUCT

Results in Figure 6.8 show that the patterns predicted by both sets of equations are similar. The main peak angle and the peak level of the APE prediction match the solutions of LEE well. The differences in the peak radiation level between both results are less than 0.5dB, whereas the peak radiation angles differ from each other by less than 1.4 deg. In other parts of the directivity prediction the patterns are also similar. Nevertheless, the shape of the results of the APE method is generally smoother than the curve of the results of the LEE solution. It implies that in this case study the APE method may introduce some kind of dissipations, which may also operate to suppress the hydrodynamic instabilities. Another finding is that the amplitude of the results of the APE method is generally higher than that of the LEE method. In Figure 6.6 the same finding is discovered by comparing SPL results, especially in the area besides the lip. In addition, the maximal difference appears in the case of $n = 4$, where the difference at the high observation angles ($\phi > 60$ deg) is up to 5.0dB.

Generally speaking, both results match well with each other. It validates the previous approach that removed some terms in the original LEE to stabilise the computation. However, there is still difference between two results, especially in the magnitude. To further validate both methods, other experiments, such as solving the case with three-dimensional Navier-Stokes equations, should be set up.

6.4 Summary

In this work the body-fitted multi-block AMR method is applied to the prediction of spinning mode radiation from a generic engine exhaust with a sheared mean flow field. To model curved geometries, the AMR code is extended to support body-fitted grids. The mean flow field is assumed to be axisymmetric. Inside the duct, a spinning mode of $m = 13$ with several different radial modes ($n = 1 - 4$) is admitted into the propagation region as input on the boundary of the computation domain. To suppress hydrodynamic instabilities developed in the exhaust mean flow, APE are employed and are extended to the cylindrical coordinates. The results obtained through solutions of APE agree well with the previous results of LEE solutions by comparing the near-field propagation patterns and far-field directivities. The computation efficiency varies along with the propagation of the acoustic waves. In the initial stage, the adaptively refined mesh represents a saving of up to 160% compared with a uniform mesh. After the acoustic waves spanning the whole computation domain the efficiency of AMR is the same as that on a uniformly fine mesh.

6. ACOUSTIC RADIATION FROM ENGINE EXHAUST DUCT

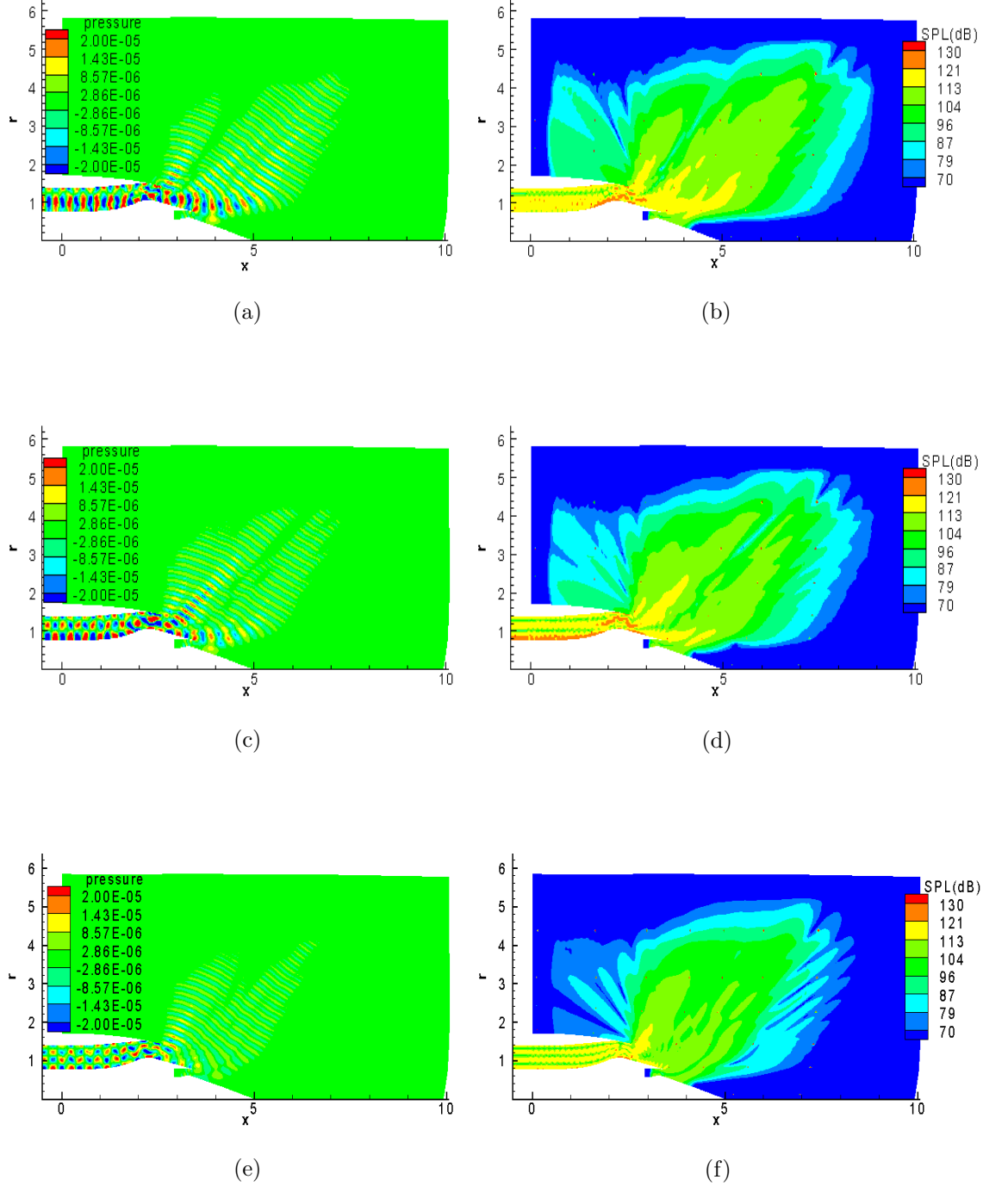


Figure 6.7: The APE prediction of perturbation pressures and SPL contours of several single spinning mode waves, where $m=13$, $n=2-4$, $k=28.3179$, $12.4 < t < 12.9$ for SPL; (a) perturbation pressures, $n=2$; (b) SPL, $n=2$; (c) perturbation pressures, $n=3$; (d) SPL, $n=3$; (e) perturbation pressures, $n=4$; (f) SPL, $n=4$.

6. ACOUSTIC RADIATION FROM ENGINE EXHAUST DUCT

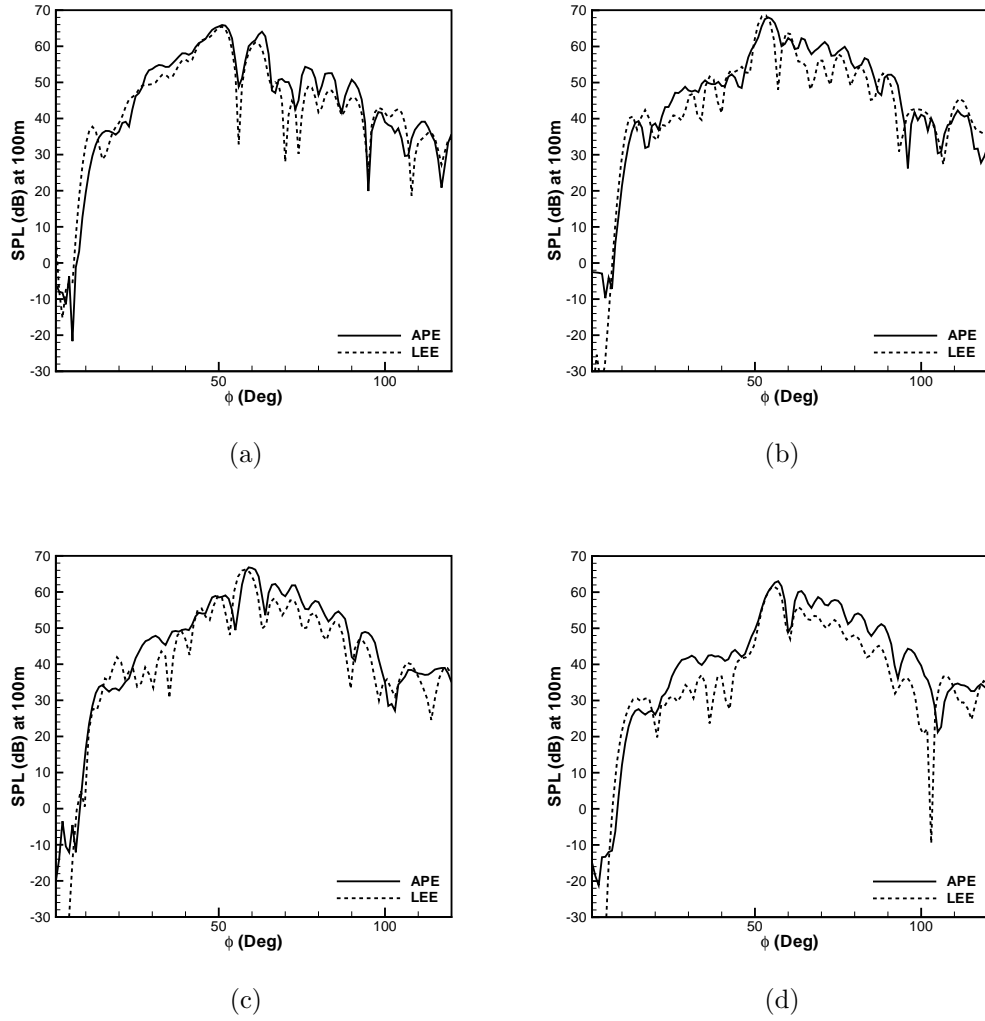


Figure 6.8: Far-field directivity of the engine exhaust duct radiation, $m=13$, $k=28.3179$, (a) $n=1$, (b) $n=2$, (c) $n=3$, and (d) $n=4$.

Chapter 7

Conclusions and Future Work

In this final chapter the main results of this work are summarised and suggestions for future research are presented.

7.1 Concluding Remarks

The objective of this work was to investigate the applications of AMR for CAA problems. This was achieved by developing a computational method, which mainly consisted of three parts: an AMR framework to manage an adaptively refined mesh; a range of 4th-order schemes to compute the near-field acoustic propagation and radiation; and several CAA applications to validate and verify the code. The layered design provided the necessary flexibility to experiment with different numerical schemes on an adaptively refined mesh. The computational technique was proved to yield higher efficiency on an adaptively refined mesh than other methods working on a uniformly fine mesh.

In the first part of this work, the algorithm of the block-structured AMR was described. The essential idea of AMR was straightforward, whereas an appropriate implementation on a distributed memory machine was far from a trivial task. In addition, it was discovered that the parallel programming of AMR was prone to subtle errors in the process of the code development. Meanwhile, the absence of a convenient parallel debugging tool exacerbated the difficulty of the development. To facilitate the development, the code was developed based on the existing PARAMESH framework. Codes in CLAWPACK and Chombo frameworks were suggestive as well. Furthermore, a couple of simplifications were made to the fundamental AMR algorithm to reduce the potential difficulty of developing and debugging on a parallel machine. The first simplification was to transport solutions of the whole block, rather than transfer solutions of a part of a block, to the corresponding processor

7. CONCLUSIONS AND FUTURE WORK

for the ghost construction operation. The second simplification was to combine all parallel communications of the AMR operations such as the ghost construction, restriction, prolongation and regridding operations together. The simplifications were justified on the discovery of the current electronic technology: the improvement of the communication performance was around two times faster than the increase of the computational performance in the last twenty years. Using the simplified algorithm, the communication cost was increased by two to three times, whilst the messy program with numerous conditional judgments was refined to a more elegant one. At the same time the simplified code still satisfied the requirements that were vital for scientific computation: portability and software re-usability.

In the second part of this work, a number of numerical issues were addressed in order to apply high-order schemes to obtain time accurate numerical solutions pertinent to aeroacoustic problems on an adaptively refined mesh. A range of high-order spatial schemes were tested, including: a standard 4th-order explicit scheme, whose coefficients were obtained by matching the corresponding Taylor series; a 4th-order explicit DRP scheme; and a 6th-order implicit compact scheme. It was found that an explicit scheme was easier to implement under the AMR environment than an implicit scheme, whilst the performance was still satisfactory in that the grid resolution could be increased easily in the desired area with the AMR method.

The most prominent problem in the present work was how to handle the innate fine-coarse block interfaces on an adaptively refined mesh. For example, in order to get the spatial differentials at the fine-coarse block interfaces, interpolations were used for the ghost construction operation. Several interpolation methods (from 2nd- to 6th-order) were tested. It was found that the type of interpolation at the fine-coarse block interfaces played an important role to preserve the favorable accuracy of high-order schemes. A combination of a 4th-order explicit scheme and a 6th-order interpolation resulted in a convergence rate of around 3.7 for a two-dimensional wave propagation case. At the same time, the stability of the employed spatial schemes combined with several interpolation methods at a fine-coarse interface was proved through spectral analysis. In addition, the transient behaviour was provided through pseudospectra analysis, which indicated that a short wavelength spurious wave could be generated at a fine-coarse block interface due to potential numerical uncertainties, such as numerical truncation errors. To remove this type of nuisance, either an explicit filter or an artificial selective damping technique was used. Their effects were discussed and compared in several case studies.

As for the temporal integration, rather than doing multi-step integrations on an adaptively refined mesh, a relatively simple 4th-order low-dissipation and low-

7. CONCLUSIONS AND FUTURE WORK

dispersion Runge-Kutta method was employed. Otherwise, high-order interpolations on temporal solutions would be required, which will introduce more numerical complexities and uncertainties to this work. In terms of its efficiency, the penalty of the Runge-Kutta scheme with the same time step on the overall mesh was regarded as acceptable for the case studies throughout this work in that generally only two to three refinement levels were set to the relevant hierarchical meshes. A multi-step time integration method may be introduced to the code as a subroutine in the future work.

In the third part of this work, some benchmark problems were solved to verify and validate the AMR code. Through these case studies, the efficiency of AMR computations was found to be affected by a number of factors, such as the updating time interval, the block size and the regridding thresholds. The tradeoff between efficiency and accuracy was assessed in a case study of sound propagation and radiation from an unflanged duct, where two different regridding threshold were used to simulate the near-field sound propagation on an adaptively refined mesh. Solutions computed on an adaptively refined mesh were compared to solutions computed on a uniform mesh through two steps. In the first step, the instantaneous perturbation pressures were studied. It showed that the AMR computation with the bigger regridding threshold was much faster than either the AMR computation with the smaller regridding threshold or the computation on a uniformly fine mesh. However, its solution failed to match the other two solutions. By contrast, the solution computed with the smaller regridding threshold matched the solution computed on the uniform mesh well. Its computational time was around 30% of the computational time on the uniform mesh. In the second step, a comparison was made between the far-field directivity computed with the AMR method and the Cargill's analytical approximation. Once again, the directivity computed with the bigger regridding threshold did not agree with the analytical approximation. By contrast, the prediction computed with the smaller regridding threshold matched the analytical approximation well. It confirmed that a properly selected regridding threshold was required to obtain accurate solutions in using the AMR method.

In this work some of the test cases that modelled by LEE were profiled. In the benchmark case of the two-dimensional acoustic propagation with a Gaussian pulse, the computation loads were distributed evenly among the processors of a cluster by MPI. The cost of each AMR operation was profiled. It showed the AMR operations consumed around 30% of the total computational cost. In a more complex case of a spinning mode sound propagation and radiation through a general aero-engine intake, the parallel scale-up performance was deteriorated along with the increase of

7. CONCLUSIONS AND FUTURE WORK

the processor number, due to the continuous increase of the communication cost. By contrast, the parallel scale-up of the well tuned SotonLEE code was a little better, where exact parts of the neighboring blocks were exchanged for the ghost construction operation and the subsequent communication cost was therefore cheaper.

In the final part of this work, the validated AMR code was applied to some general aeroacoustic applications, *i.e.* spinning mode sound radiation from aero-engine intake and exhaust ducts, where the AMR method was extended to support body-fitted multi-blocks to allow for the solutions of problems with general geometries. The sound radiation problems were solved in two steps.

Firstly, in the case of an intake duct, propagation inside the duct, diffraction at the lip of the duct and propagation into the near-field was modelled by LEE. The result was compared to the solution computed on a uniformly fine mesh and the solution of a FEM solver. It was found that the near-field solution computed on an adaptively refined mesh agreed well with the solutions computed by the other two methods. Meanwhile, the far-field radiation directivity was predicted, based on the near-field solution, by using a previously developed code to calculate the FW-H equation in the integral form. The far-field result showed that both peak amplitude and peak angle computed with the AMR method were matched well with the far-field solutions of the other two methods.

Secondly, in the case of an bypass duct, the associated exhaust mean flow generated shear layer instabilities that were failed to be suppressed by LEE. The unbounded instabilities overwhelmed the acoustic solutions. In order to inhibit this type of instabilities, LEE were replaced with APE, which were extended to the cylindrical coordinates. The suitability of the governing equations and the quality of the proposed AMR method were validated through this case study of single spinning mode radiation from a generic engine bypass duct. The far-field prediction of the APE solution was compared to the far-field results of the LEE solution. Both peak amplitude and peak angle were well agreed with these two methods, whilst the amplitudes of the APE prediction were up to 5dB higher in other lobes.

Finally, as part of this PhD thesis research, in Appendix C a Fourier based pseudospectral time domain (PSTD) method was studied and applied to some aeroacoustic benchmark problems. A new algorithm, for linear wave propagation problem, was developed and tested. A hard-wall boundary condition was supplied for simple geometry and was validated. Combined with the buffer zone technique to reduce contamination due to wave rewrap, the algorithm was fully-fledged to some linear wave propagation problems. For general problems with nonlinear terms, the original algorithm of the Fourier pseudospectral time domain method was proved to be an

alternative to high-order finite difference methods.

7.2 Future Work

To make the AMR code more robust and flexible, some further works may be necessary. They are listed below.

- (1) Improve the efficiency of the ghost construction operation. In this work the operation was applied on a temporary buffer to reduce the required memory costs. The method, however, incurred a lot of extra communication costs. The profile results showed the cost of the ghost construction operation consumed up to 30% of the whole computation cost. It is therefore worthwhile to optimize the ghost construction operation by assigning extra memory buffer for each block.
- (2) Make each block configuration more flexible to contain different number of cells. In this work each block contained the same cell number for the reason of simplicity. Unfortunately, it made the mesh generation around a general geometry much more difficult, as was shown in the aero-engine case studies. More programming efforts, as a remedy, are required to make the code suit to blocks with variable cell numbers. The AMR operations and the parallel communication subroutine should be revised accordingly.

Finally, other aeroacoustic applications with more complex geometries are expected to be tackled under the AMR environment to demonstrate its suitability and flexibility as well as to demonstrate its efficiency.

Appendix A

Coefficients of Numerical Schemes

A.1 Numerical Schemes

Assuming there are n gridpoints in each direction of the coordinates, the sketch is displayed in Figure A.1. The coefficients of several spatial discretization schemes and interpolation methods used in this work are listed here, where single-side stencils are employed for grid points besides computational domain and solid boundaries, whilst central stencils are used for other grid points.

$$1 \quad 2 \quad 3 \quad 4 \quad i-2 \quad i-1 \quad i \quad i+1 \quad i+2 \quad n-3 \quad n-2 \quad n-1 \quad n$$

Figure A.1: A stencil in the one-dimensional coordinate.

A.1.1 The DRP Scheme

The following stencils are obtained from [159] and given here to complete this work. The notation of any coefficient has the format as d_{index}^{lr} , where the subscript is the index of coefficients, and the superscript lr denotes the number of gridpoints located in the left and right of the computed gridpoint respectively. To keep the global consistency in this work, the superscript and subscript are not the same as [159].

A. COEFFICIENTS OF NUMERICAL SCHEMES

Table A.1: The coefficients of a 4th-order DRP scheme.

Index	33	06	15	24
a_0	-0.0208431427703	-2.192280339	-0.209337622	0.049041958
a_1	0.166705904415	4.748611401	-1.084875676	-0.468840357
a_2	-0.7708823805180	-5.108851915	2.147776050	-0.474760914
a_3	0	4.461567104	-1.388928322	1.273274737
a_4	0.7708823805180	-2.833498741	0.768949766	-0.518484526
a_5	-0.166705904415	1.128328861	-0.281814650	0.166138533
a_6	0.0208431427703	-0.203876371	0.048230454	-0.026369431

The coefficients of the following stencils are listed in Table A.1.

A.1.1.1 The Central Stencil

$$f'_i = \frac{a_0^{33} f_{i-3} + a_1^{33} f_{i-2} + a_2^{33} f_{i-1} + a_3^{33} f_i + a_4^{33} f_{i+1} + a_5^{33} f_{i+2} + a_6^{33} f_{i+3}}{\Delta x}. \quad (\text{A.1})$$

A.1.1.2 The Single-side Stencil at Boundaries

$$f'_1 = \frac{a_0^{06} f_1 + a_1^{06} f_2 + a_2^{06} f_3 + a_3^{06} f_4 + a_4^{06} f_5 + a_5^{06} f_6 + a_6^{06} f_7}{\Delta x}, \quad (\text{A.2})$$

$$f'_2 = \frac{a_0^{15} f_1 + a_1^{15} f_2 + a_2^{15} f_3 + a_3^{15} f_4 + a_4^{15} f_5 + a_5^{15} f_6 + a_6^{15} f_7}{\Delta x}, \quad (\text{A.3})$$

$$f'_3 = \frac{a_0^{24} f_1 + a_1^{24} f_2 + a_2^{24} f_3 + a_3^{24} f_4 + a_4^{24} f_5 + a_5^{24} f_6 + a_6^{24} f_7}{\Delta x}, \quad (\text{A.4})$$

$$f'_n = \frac{a_0^{60} f_n + a_1^{60} f_{n-1} + a_2^{60} f_{n-2} + a_3^{60} f_{n-3} + a_4^{60} f_{n-4} + a_5^{60} f_{n-5} + a_6^{60} f_{n-6}}{\Delta x}, \quad (\text{A.5})$$

$$f'_{n-1} = \frac{a_0^{51} f_n + a_1^{51} f_{n-1} + a_2^{51} f_{n-2} + a_3^{51} f_{n-3} + a_4^{51} f_{n-4} + a_5^{51} f_{n-5} + a_6^{51} f_{n-6}}{\Delta x}, \quad (\text{A.6})$$

$$f'_{n-2} = \frac{a_0^{42} f_n + a_1^{42} f_{n-1} + a_2^{42} f_{n-2} + a_3^{42} f_{n-3} + a_4^{42} f_{n-4} + a_5^{42} f_{n-5} + a_6^{42} f_{n-6}}{\Delta x}, \quad (\text{A.7})$$

where $a_{0-7}^{06} = -a_{0-7}^{60}$, $a_{0-7}^{15} = -a_{0-7}^{51}$, $a_{0-7}^{24} = -a_{0-7}^{42}$.

A. COEFFICIENTS OF NUMERICAL SCHEMES

Table A.2: The coefficients of a 4th-order standard explicit scheme.

Index	22	04	13
a_0	1/12	-11/6	-1/3
a_1	-2/3	3	-0.5
a_2	0	-3/2	1
a_3	2/3	1/3	-1/6
a_4	-1/12		

A.1.2 The Standard Explicit Scheme

The following stencils are obtained by matching Taylor series [18]. A typographical error appeared in that paper has been corrected here. The meaning of subscript and superscript are the same as with the previous section. The coefficients are in Table A.2.

A.1.2.1 The Central Stencil

$$f'_i = \frac{a_0^{22} f_{i-2} + a_1^{22} f_{i-1} + a_2^{22} f_i + a_3^{22} f_{i+1} + a_4^{22} f_{i+2}}{\Delta x}. \quad (\text{A.8})$$

A.1.2.2 The Single-side Stencil at Boundaries

$$f'_1 = \frac{a_0^{04} f_1 + a_1^{04} f_2 + a_2^{04} f_3 + a_3^{04} f_4}{\Delta x}, \quad (\text{A.9})$$

$$f'_2 = \frac{a_0^{13} f_1 + a_1^{13} f_2 + a_2^{13} f_3 + a_3^{13} f_4}{\Delta x}, \quad (\text{A.10})$$

$$f'_n = \frac{a_0^{40} f_n + a_1^{40} f_{n-1} + a_2^{40} f_{n-2} + a_3^{40} f_{n-3}}{\Delta x}, \quad (\text{A.11})$$

$$f'_{n-1} = \frac{a_0^{31} f_n + a_1^{31} f_{n-1} + a_2^{31} f_{n-2} + a_3^{31} f_{n-3}}{\Delta x}, \quad (\text{A.12})$$

where $a_{0-3}^{04} = -a_{0-3}^{40}$, $a_{0-3}^{13} = -a_{0-3}^{31}$.

A.1.3 Interpolations

$$f_i = \sum_{j=0}^5 a_i (f_{i-j} + f_{i+j}), \quad (\text{A.13})$$

where f represents a solution variable. The coefficients of a 2nd- and 10th-order interpolation methods are in Table A.3. To save space, other interpolations, such as

A. COEFFICIENTS OF NUMERICAL SCHEMES

Table A.3: The coefficients of several explicit interpolations.

Order	a_0	a_1	a_2	a_3	a_4	a_5
2^{nd}	0.25	0.25	0	0	0	0
10^{th}	0.37695	0.205	-0.117	0.0439	-0.009765	0.0009765

6^{th} -order and single-side interpolations used in the code, are not included here.

A.2 Operator Matrix in Eigenvalue Analysis

The operator matrixes \mathbf{M} used in section 3.3 have the form as follows:

$$\mathbf{M} = \begin{bmatrix} \mathbf{b} & 0 & & \dots & & & 0 \\ \mathbf{a} & 0 & & \dots & & & 0 \\ 0 & \mathbf{a} & & \dots & & & 0 \\ & \vdots & & \vdots & & & \\ 0 & 0 & \dots & \mathbf{M}' & \dots & 0 & 0 \\ & \vdots & & \vdots & & & \\ 0 & 0 & & \dots & 0 & 0.5\mathbf{a} & 0 \\ 0 & 0 & & \dots & 0 & 0 & 0.5\mathbf{a} \\ 0 & 0 & & \dots & 0 & 0 & 0.5\mathbf{a}' \end{bmatrix}, \quad (\text{A.14})$$

where \mathbf{b} is a coefficients matrix of single-sided boundary stencils, \mathbf{a} is a coefficients matrix of interior stencils, \mathbf{M}' is a coefficients matrix of stencils on the fine-coarse interface, \mathbf{a}' is a truncated coefficients matrix of central stencils. A simple example of \mathbf{M} with the dimensional size 8 is listed here for readers' reference.

$$\begin{bmatrix} -11/6 & 3 & -1.5 & 1/3 & 0 & 0 & 0 & 0 \\ -1/3 & -0.5 & 1 & -1/6 & 0 & 0 & 0 & 0 \\ 1/12 & -2/3 & 0 & 2/3 & -1/12 & 0 & 0 & 0 \\ 0 & 1/12 & -2/3 & 0 & 5/8 & -1/24 & 0 & 0 \\ 0 & 0 & 1/12 & -2/3 & 1/3 & 0.25 & 0 & 0 \\ 0 & 0 & 1/24 & 0 & -1/3 & 0 & 1/3 & -1/24 \\ 0 & 0 & 0 & 0 & 1/24 & -1/3 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 1/24 & -1/3 & 0 \end{bmatrix}, \quad (\text{A.15})$$

A. COEFFICIENTS OF NUMERICAL SCHEMES

this is derived using the aforementioned 2^{nd} -order interpolation and 4^{th} -order explicit scheme. To save space other operator matrixes are not listed.

Appendix B

Asymptotic Solution of Sound Out of Jet

In this appendix an asymptotic solution of a high frequency sound radiating out of a jet is presented for the completeness of this work.

B.1 Introduction

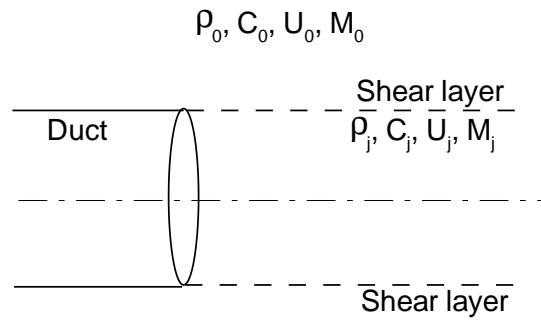


Figure B.1: Setup of an idealised jet case.

This problem is idealised as a jet issuing from a semi-infinite cylindrical pipe in which there is a uniform mean flow, as shown in Figure B.1. Munt has solved the problem exactly by using the Wiener-Hopf technique [155] that is rather complicated. Cargill presented a simpler asymptotic solution, which resembled Munt's solution in the peak lobe for high frequency sound, based on Kirchhoff approximation [121]. In this work numerical results of a spinning mode sound radiation case are tested against Cargill's asymptotic solution. However, it is noticeable that there are some small errors in [121]. To prevent possible confusions, revised equations used in the work are listed here, in which symbols and their meanings used by Cargill are kept, although some are not consistent with those symbols that have been employed in this work. To save space only the most important symbols are introduced here, others can be found in [121].

B.2 Equations

The sound pressure of a given incident duct mode is in the form of:

$$p_i = J_m\left(\frac{j'_{mn}r}{a}\right)e^{(-ik_0x - im\phi + i\omega t)}, \quad (\text{B.1})$$

where the radial wavenumber j'_{mn} satisfies the hard wall boundary condition $J_m(j'_{mn}) = 0$, a is the radius of the cylindrical pipe, k_0 is streamwise wavenumber in x axis direction, m is azimuthal mode, ϕ is azimuthal angle, ω is sound frequency, t is time.

The far-field pressure is:

$$p = -\left(\frac{a}{4R}\right)e^{(-i(\frac{\omega R}{c_0}) + i\omega t + im(\frac{\pi}{2}) - im\phi)}DIT, \quad (\text{B.2})$$

$$D = -ia(\chi + \chi_0), \quad (\text{B.3})$$

$$I = \frac{-2vaJ'_m(va)J_m(j'_{mn})}{(va)^2 - (j'_{mn})^2}, \quad (\text{B.4})$$

$$T = T(k_0) = -\frac{2}{(\pi va)(J'_m(va)H_m^{(2)}(wa) - \alpha J_m(va)H_m^{(2)'}(wa))}, \quad (\text{B.5})$$

where R is distance between the far-field observer and the origin position of the pipe

B. ASYMPTOTIC SOLUTION OF SOUND OUT OF JET

exit, c_0 is ambient sound speed, J'_m is the differential of the corresponding argument, and

$$\chi = k_0 + \frac{U_j(\omega - U_j k_0)}{c_j^2}, \quad (\text{B.6})$$

$$\chi_0 = \frac{\omega}{c_j} \left(1 - (1 - M_j^2) \frac{j_{mn}'^2 c_j^2}{\omega^2 a^2} \right), \quad (\text{B.7})$$

$$v^2 = \left(\frac{(\omega - U_j k_0)^2}{c_j^2} \right) - k_0^2, \quad w^2 = \left(\frac{(\omega - U_0 k_0)^2}{c_0^2} \right) - k_0^2, \quad (\text{B.8})$$

$$\alpha = \frac{\rho_j D_j^2 w}{\rho_0 D_0^2 v}, \quad D_0 = (\omega - U_0 k_0), \quad D_j = (\omega - U_j k_0), \quad (\text{B.9})$$

where v, w should be chosen to satisfy $\text{Im}(v) < 0$ and $\text{Im}(w) < 0$. There are also some equations do not appear in Cargill's paper, but are deserved to be mentioned, *e.g.* the fluid is modelled as a perfect gas and $c_0^2 = \gamma p_0 / \rho_0$, $c_j^2 = \gamma p_j / \rho_j$, assuming $p_0 = p_j$, the relation of ω and k is $\omega = k c_0$, and the relation of densities inside and outside of jet is $\rho_j / \rho_0 = c_0^2 / c_j^2$.

B.3 Results and Discussion

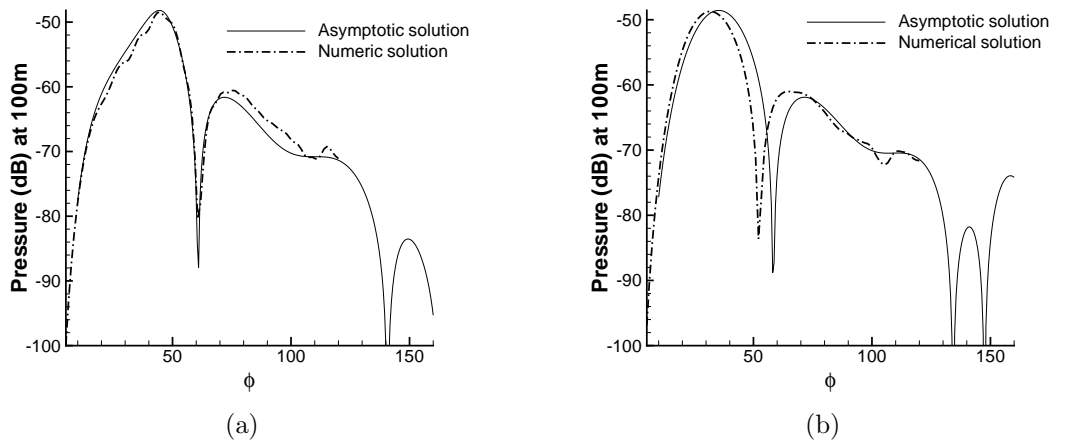


Figure B.2: Comparison of asymptotic and numerical solutions, where $(m, n) = (4, 1)$, $M_j = 0.14$, $k = 11.7$, (a) $M_0 = 0$, (b) $M_0 = 0.14$.

The equations are programmed by the script language and computed in MATLAB

B. ASYMPTOTIC SOLUTION OF SOUND OUT OF JET

to get the asymptotic solution, which are tested against previous numerical solutions. Results are compared and shown in Figure B.2. In Figure B.2(a) the asymptotic solution matches the numerical results [123] satisfactory. It illustrates that the far-field pressure is simulated reliably when the ambient flow velocity equals to zero. In case of a nonzero ambient flow velocity, the angles of the first peak lobe of both solutions do not match exactly, as demonstrated in Figure B.2(b), but the amplitude and shape are still similar. Further validation should depend on experiment data.

Appendix C

Fourier Based Pseudospectral Method

This appendix introduces a Fourier based pseudospectral time-domain (PSTD) method that is applied to wave propagation problems pertinent to CAA [160]. This chapter has little connection with the previously presented AMR work. The relevant information and governing equations are therefore provided to make this chapter self-contained.

The original algorithm of the Fourier based pseudospectral time-domain method works for periodical problems without the interaction with physical boundaries. In this appendix a slip wall boundary condition is developed, combined with buffer zone technique, to solve some non-periodical problems. For a linear sound propagation problem whose governing equations could be transferred to ordinary differential equations in pseudospectral space, a new algorithm only requiring time stepping is developed and tested. For other wave propagation problems, the original algorithm has to be employed and the developed slip wall boundary condition still works. The accuracy of the presented numerical algorithm is validated by benchmark problems and the efficiency is assessed by comparing with high-order finite difference methods. It is indicated that the Fourier based pseudospectral time-domain method, time stepping method, slip wall and absorbing boundary conditions combine together to form a fully-fledged computational algorithm.

C.1 Introduction

Pseudospectral time-domain methods were developed to achieve spectral level accuracy in numerical solutions of the partial differential equations. So far, a number of attempts were made to apply numerical algorithms based on the pseudospectral time-domain methods to simulate various wave phenomena such as electromagnetic, seismic and acoustic waves [161, 162, 163, 164], with various degrees of success. It is accepted that pseudospectral time-domain methods have high spatial resolution that meets the requirements of numerical simulation of aeroacoustic phenomena. In this work, we apply a class of pseudospectral time-domain method based on the Fourier transformation to sound propagation problems commonly encountered in aeroacoustics.

The basic idea of pseudospectral time-domain method is to represent the spatial derivatives in the spectral domain by a set of basis functions. There are two categories of orthogonal functions which are commonly used as the basis functions. One is the Fourier series that can be used in periodical problems [161]. The other and more commonly used function is the Chebyshev polynomials. The advantage of the Chebyshev pseudospectral time-domain method lies in its ability to deal with non-periodic problems on non-uniform and multi-domain computational grids [165, 166] at the cost of computational efficiency. On the other hand, the Fourier based pseudospectral time-domain method is simple to implement and has comparatively low computing cost. It does though have certain restrictions, *e.g.* solutions should satisfy Lipschitz condition; the method has to work on a uniform grid and is only applicable to periodical problems. The current work addresses some of these issues/restrictions in the development of numerical algorithms based on the Fourier based pseudospectral time-domain method, under the context of computational aeroacoustics.

In the implementation of a Fourier based pseudospectral time-domain method, discrete Fourier transforms are applied to the perturbation variables, resulting in a spectral pair of the original variables. The spatial derivatives of the original variables can then be calculated analytically through multiplications of the spatial sampling frequency and spectral pair of the variables. In the case of a one-dimensional problems, the spectral pair of the original variable $y(x, t)$ are $\bar{Y}(k_x, t)$, and the spectral pair of its derivative, $\partial y(x, t)/\partial x$, are $jk_x\bar{Y}(k_x, t)$, where k_x in the above expressions is the wavenumber rather than the meaning of sampling frequency in the temporal sequence. According to the Nyquist criteria, only two points-per-wavelength are required to obtain exact and dispersion free results [167]. This compares with other high-order finite difference methods, such as compact schemes, where typically eight

C. FOURIER BASED PSEUDOSPECTRAL METHOD

or more points-per-wavelength are required to meet the dispersion requirement.

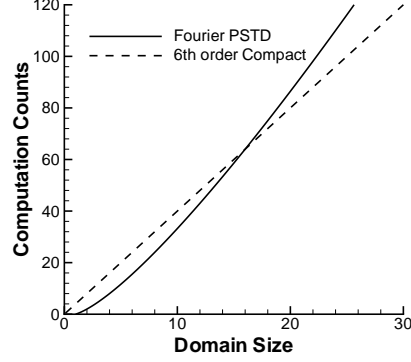


Figure C.1: A schematic of scaling of computation counts with grid size.

Comparing to a typical high-order finite difference compact scheme, a potential performance limiting factor in applying the Fourier based pseudospectral time-domain method is the relative deterioration in computation efficiency as larger grids are used. For a one-dimensional problem, the cost of performing discrete Fourier transform is proportional to $O(m \log m)$, where m is the number of the discrete spatial points. Typically, a high-order finite difference method requires $O(km)$ counts to obtain the derivative, where k is a constant for a specific scheme and generally has a value less than 6. Figure C.1 gives an illustration of the relative computation counts for one derivative scaled to the size of the computation domain. For a large computation domain, a pseudospectral time-domain method could potentially have lower computation efficiency than some high-order finite difference methods. However, it is worth emphasizing that this one-dimensional case represents the worst case scenario. For partial differential equations with multiple variables and derivatives, the discrete Fourier transforms only need to be calculated once at each time step. The relative computational performance improves accordingly.

The linear wave propagation problems are simulated using an algorithm that alleviates the performance limiting problem described above. This algorithm reduces the discrete Fourier transform operations at each time step. Details are described in Section 2 of the paper. In Section 3, issues of the points-per-wavelength requirement, a slip wall boundary condition and a buffer zone technique are addressed. In Section 4 the Fourier based pseudospectral time domain method is applied to two computational aeroacoustics benchmark problems, such as the linear problem of the propagation of a two-dimensional Gaussian pulse with reflections off a hard wall and

the sound propagation of an open rotor [145, 168]. A summary of the present work is provided in Section 5.

C.2 Governing Equations and Algorithm

C.2.1 Governing Equations

The governing partial differential equations used to describe linear wave propagation phenomena in a uniform medium are given below. Various forms of the equations are employed. The one-dimensional convection equation takes the form of

$$\frac{\partial u'}{\partial t} + \frac{\partial u'}{\partial x} = 0. \quad (\text{C.1})$$

The one-dimensional linearised Euler equations for acoustics wave propagation is given as:

$$\frac{\partial u'}{\partial t} + \frac{\partial p'}{\partial x} = 0, \quad (\text{C.2})$$

$$\frac{\partial p'}{\partial t} + \frac{\partial u'}{\partial x} = 0. \quad (\text{C.3})$$

The two-dimensional linearised Euler equations for acoustics wave propagation are given as:

$$\frac{\partial u'}{\partial t} + \frac{\partial p'}{\partial x} = 0, \quad (\text{C.4})$$

$$\frac{\partial v'}{\partial t} + \frac{\partial p'}{\partial y} = 0, \quad (\text{C.5})$$

$$\frac{\partial p'}{\partial t} + \frac{\partial u'}{\partial x} + \frac{\partial v'}{\partial y} = 0. \quad (\text{C.6})$$

In the above equations, t is the time, x and y are the Cartesian coordinates, u' and v' are velocity perturbations and p' is the pressure perturbation. For the rest of the paper, the prime sign will be dropped for convenience. The fluid is modelled as a perfect gas and all variables are nondimensionalised using a reference length L^* , a reference sound speed a^* and a reference density ρ^* .

C.2.2 Governing Equations in The Pseudospectral Domain

With the assumption that the spatial domain is periodical, the one-dimensional convection equation, Eq. (C.1) can be transformed to:

$$\frac{d\bar{U}(k_x, t)}{dt} + jk_x \bar{U}(k_x, t) = 0, \quad (\text{C.7})$$

where $\bar{U}(k_x, t)$ is the pseudospectral pair for $u(x, t)$ and k_x is the wavenumber in the x direction. By employing a suitable Runge-Kutta time-stepping scheme, *e.g.* a low-dissipation and low-dispersion scheme [111], Eq. (C.7) can be stepped directly to the new time step $t + k\Delta t$ as an ordinary differential equation to yield $\bar{U}(k_x, t + k\Delta t)$. A solution in the spatial domain is obtained by applying an inverse Fourier transform to $\bar{U}(k_x, t + k\Delta t)$, producing an updated solution $u(x, t + k\Delta t)$.

The one-dimensional linear wave equations are transformed by the Fourier based pseudospectral time-domain method to:

$$\frac{\partial \bar{U}(k_x, t)}{\partial t} + jk_x \bar{P}(k_x, t) = 0, \quad (\text{C.8})$$

$$\frac{\partial \bar{P}(k_x, t)}{\partial t} + jk_x \bar{U}(k_x, t) = 0, \quad (\text{C.9})$$

where $\bar{P}(k_x, t)$ and $\bar{U}(k_x, t)$ are the pseudospectral pair for the pressure perturbation $p(x, t)$ and velocity perturbation $u(x, t)$ respectively.

The transformed two-dimensional linear wave equations are as follows:

$$\frac{\partial \bar{\bar{U}}(k_x, k_y, t)}{\partial t} + jk_x \bar{\bar{P}}(k_x, k_y, t) = 0, \quad (\text{C.10})$$

$$\frac{\partial \bar{\bar{V}}(k_x, k_y, t)}{\partial t} + jk_y \bar{\bar{P}}(k_x, k_y, t) = 0, \quad (\text{C.11})$$

$$\frac{\partial \bar{\bar{P}}(k_x, k_y, t)}{\partial t} + jk_x \bar{\bar{U}}(k_x, k_y, t) + jk_y \bar{\bar{V}}(k_x, k_y, t) = 0. \quad (\text{C.12})$$

In Eqs. (C.10-C.12), $\bar{\bar{U}}(k_x, k_y, t)$, $\bar{\bar{V}}(k_x, k_y, t)$ and $\bar{\bar{P}}(k_x, k_y, t)$ are the two-dimensional Fourier transforms of the velocity perturbations $u(x, y, t)$ and $v(x, y, t)$ and pressure perturbation $p(x, y, t)$ respectively.

The above procedures can be applied to linear wave propagation equations with

an mean flow to obtain the transformed equations in pseudospectral domain.

C.2.3 Performance Analysis

The original algorithms of Fourier based pseudospectral time domain method [166, 167] has the following form:

$$\frac{\partial u_i}{\partial t} + \text{IDFT}(jk_{x_i}\text{DFT}(f_i(x_i))) = 0, \quad (\text{C.13})$$

where DFT and IDFT denote forward and inverse discrete Fourier transforms. This procedure works fine for general cases. But the forward and inverse discrete Fourier transforms will have to be used at every time step to obtain the spatial derivatives.

For the linear problems discussed in Section C.2.2, after the Fourier transforms are applied to Eqs. (C.1- C.6), the transformed governing equations are solved as ordinary differential equations, Eqs. (C.7-C.12). The approach adopted in this work is to apply the discrete Fourier transform at the beginning of each time stepping operation. The time stepping operation is then applied to the resulting ordinary differential equations. The updated solution in the time domain is obtained by an indirect discrete Fourier transform operation in the spatial domain. The computation cost for the spatial derivatives at each time step is reduced with this procedure.

In the case of a one-dimensional computational domain of m grid points, the fast Fourier transform algorithm requires operations in the order of $O(m\log(m))$, a typical low-dissipation and low-dispersion Runge-Kutta scheme requires operations in the order of $O(4m)$ and a typical prefactored compact scheme's computational complexity is in the order of $O(6m)$ [102]. It can be estimated that for each time step, the prefactored compact scheme's computational counts are in the order of $O(10m)$ and the Fourier based pseudospectral time domain method of the original algorithm [167] needs computation counts in the order of $O(m\log m + 4m)$. By comparison, the new computation procedure presented in this paper requires operations in the order of $O(4m)$. In fact it was acknowledged that for some applications the early algorithm for the Fourier based pseudospectral time domain method had a comparable computing speed to an efficient finite difference scheme [169], even if a coarser grid was employed.

C.3 Issues and Solutions

There are several issues in applying Fourier based pseudospectral time domain method to computational aeroacoustics problems, such as resolution requirement and bound-

ary conditions. These are discussed in this section. The discussions apply to both algorithms of the pseudospectral time domain method.

C.3.1 Points-per-wavelength Requirement

For the Fourier based pseudospectral time domain method a grid resolution of two points-per-wavelength is enough. Results in Figure C.2 demonstrate this point. In this exercise, a one-dimensional Gaussian pulse is again employed. The initial condition is defined by $u_{init} = 0.5e^{(-4\log(2)(x-50)^2)}$. Two resolutions are employed: points-per-wavelength of 4 and 2. The computed pulses compare well with the analytical results.

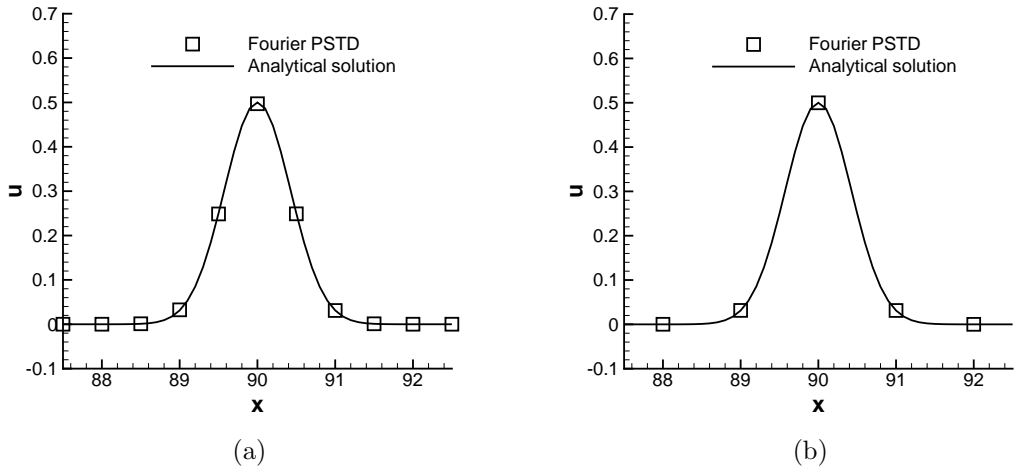


Figure C.2: One-dimensional Gaussian pulse propagation with low points-per-wavelength. (a) PPW=4, $\Delta t/\Delta x = 0.1$, steps=200; (b) PPW=2, $\Delta t/\Delta x = 0.2$, steps=100.

C.3.2 Hard-wall Boundary Condition

The Fourier based pseudospectral time domain method can effectively solve computational aeroacoustics problems with no dispersion errors. The same property can be found in Schulten's characteristic method [170]. However, the characteristic method could not solve problems with the presence of solid bodies and simulate the resulted sound reflection.

Based on the idea of generalized function [171], a hard-wall condition can be supplied now. For simplicity the one-dimensional wave propagation Eqs. (C.2)-(C.3)

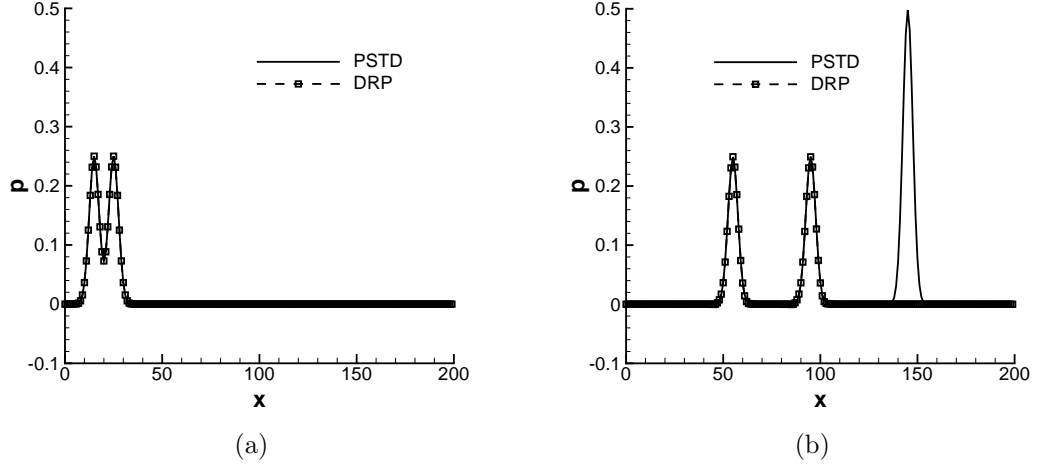


Figure C.3: One-dimensional Gaussian pulse reflected by a left hard-wall and without using buffer zone: (a) steps=10; (b) steps=150.

are used in the derivation. A stationary hard-wall condition is assumed on the left boundary of the computation domain at $x = 0$. The hard-wall condition suggests zero normal velocity at the wall. To ensure a correct velocity field, the following condition needs to be enforced:

$$u(0) = \frac{u(0+) + u(0-)}{2} = 0. \quad (\text{C.14})$$

Eq. (C.3) can be re-casted using the idea of generalized derivative for functions with discontinuities [171] to:

$$\frac{\partial p(x, t)}{\partial t} + \frac{\partial u(x, t)}{\partial x} + (u(0+, t) - u(0-, t))\delta(x) = 0. \quad (\text{C.15})$$

Eq. (C.15) is transferred by the discrete Fourier transform to:

$$\frac{\partial \bar{P}(k_x, t)}{\partial t} + jk_x \bar{U}(k_x, t) + \frac{2u(0+, t)}{\Delta x} = 0, \quad (\text{C.16})$$

where $u(0+, t)$ is approximated by $u(0, t)$, which is obtained from an inverse discrete Fourier transform operating on $\bar{U}(k_x, t)$ in each step.

An example of the application with hard-wall condition is shown in Figure C.3, where a one-dimensional wave is reflected from a hard wall at the left boundary. The

initial condition is a Gaussian pulse defined by $p_{init} = 0.5e^{(-\log(2)(x-50)^2/9)}$, $u_{init} = 0$. Eqs. (C.8) and (C.16) are used to obtain the solutions. Comparison is made with a dispersion-relation-preserving (DRP) scheme [109]. In the most part, two results agree well, but a rewrap wave appears when the Fourier based pseudospectral time domain method is used. It is generated by the periodical boundary condition and can be absorbed by the technique described in the next section.

C.3.3 Absorbing Condition for The Rewrap Waves

The original Fourier based pseudospectral time domain method works for problems with periodical boundaries. When the periodical assumption is not satisfied, wave rewrap phenomenon will appear and contaminates the solutions in the computation domain. In this work, an explicit form of buffer zone techniques [147] is applied to absorb the reflected waves. The buffer zone technique works in the spatial domain, consequently the new algorithm for the Fourier based pseudospectral time domain method requires more operation counts. The exact number depends on the width of the buffer zone; there is therefore a tradeoff between memory and speed.

In the implementation, the solution vector is explicitly damped after every several time step by:

$$\mathbf{F}(x, t + \Delta t) = \bar{\mathbf{F}}(x, t + \Delta t) - \sigma(\bar{\mathbf{F}}(x, t + \Delta t) - \mathbf{F}_0(x)), \quad (\text{C.17})$$

where $\bar{\mathbf{F}}(x, t + \Delta t)$ is the solution vector computed after regular time steps. The damping coefficient, σ , varies according to the function,

$$\sigma(x) = \sigma_{max} \left| 1 - \frac{x - L}{L} \right|^\beta, \quad (\text{C.18})$$

where L is the width of the buffer zone, x is the distance from the inner boundary of the buffer zone and σ_{max} and β are set to 1.0 and 3.0 respectively. The target solution \mathbf{F}_0 is set as zero.

C.4 Applications to Benchmark Problems

The aforementioned method is applied to two benchmark test cases. Results and discussions are given here. In the first case, a two-dimensional Gaussian pulse prop-

agation problem with hard-wall and absorbing boundaries was computed, employing temporal integration directly on pseudospectral space. In the second case, the algorithm in the form of Eq. C.13 was used to solve for an open rotor problem with nonlinear terms.

C.4.1 Wave Propagation and Reflection

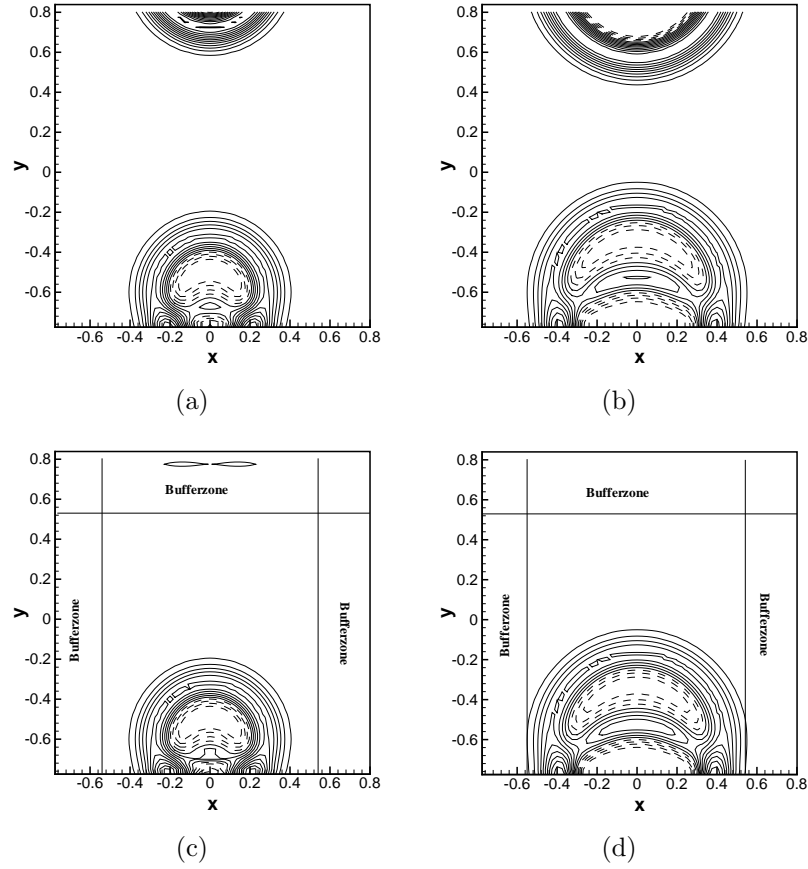


Figure C.4: The propagation of a two-dimensional Gaussian pulse: (a) $t = 0.25$, without buffer zone; (b) $t = 0.4$, without buffer zone; (c) $t = 0.25$, with buffer zone; (d) $t = 0.4$, with buffer zone.

The case is the first problem of category 4 that is defined at first computational aeroacoustics workshop. The initial condition is a Gaussian acoustic pulse given by:

$$p_{init} = e^{-\log(2.0)(x^2+(y+0.6)^2)/0.006}, \quad (\text{C.19})$$

$$u_{init} = 0, \quad (\text{C.20})$$

C. FOURIER BASED PSEUDOSPECTRAL METHOD

$$v_{init} = 0, \quad (C.21)$$

Eqs. (C.10)-(C.12) are used to solve the problem. Results are presented in Figure C.4. The hard-wall boundary condition on the bottom boundary appears to have reproduced the reflected waves off the bottom wall. In this exercise, the length of the buffer zone is set to 10 grid points. In the current computation, the buffer zone is not updated at each time step. Instead, the solutions inside the buffer zone are updated at regular step intervals, *e.g.* once every two or four steps, to save computing time furthermore. The algorithm employed for this exercise can be found at the end of this appendix.

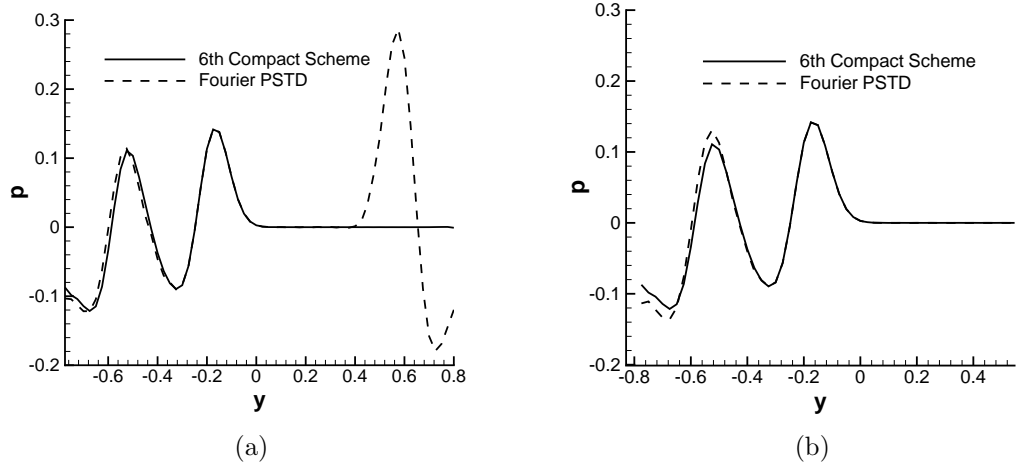


Figure C.5: A comparison of two-dimensional Gaussian pulse propagation predicted by the Fourier based pseudospectral method and the prefactored compact scheme, where pressures are along the $x = 0$ axis, $t = 0.4$: (a) without buffer zone; (b) with buffer zone.

The pressure distribution along the $x = 0$ axis is given in Figure C.5 and compared with prediction given by a prefactored compact scheme [102]. The computing time t and L_2 error against to an analytical solution of linearised Euler equations [109] are listed in Table 1, where the spatial resolution is low, from around 3 points-per-wavelength to 12 points-per-wavelength.

It has been discovered that with Fourier based pseudospectral time domain method, if a buffer zone is not applied, wave rewrapping will contaminate the solutions. However, if the rewrap wave is not considered in computing L_2 errors over two grids, they are 0.0107 and 0.00083 correspondingly, indicating the pseudospectral method is actually more exact than the compact scheme. Figure C.4(c-d) and Figure C.5(b) illustrate that a buffer zone keeps removing the rewrap wave. The agreement is gen-

C. FOURIER BASED PSEUDOSPECTRAL METHOD

Table C.1: Computing time t and L_2 error of the first benchmark problem.

Schemes	16×16	64×64
Compact scheme	$t = 1.55s, Err = 0.0425$	$t = 91s, Err = 0.0013$
Pseudospectral (no buffer zone)	$t = 0.44s, Err = 0.0938$	$t = 26s, Err = 0.0825$
Pseudospectral (buffer zone)	$t = 0.48s, Err = 0.0275$	$t = 28s, Err = 0.0075$

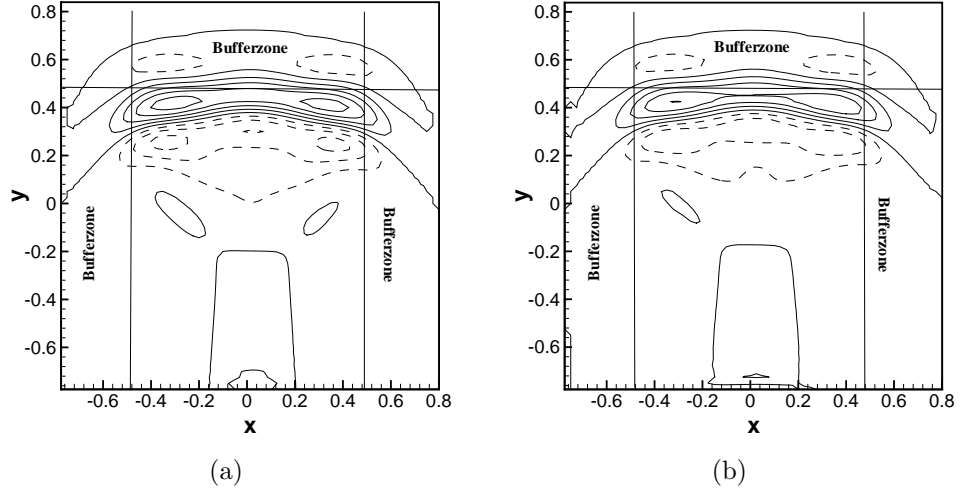


Figure C.6: A comparison of two-dimensional Gaussian pulse propagation after 110 time steps, where the pressure contours of (a) the Fourier PSTD method and (b) the compact scheme are displayed.

erally good as well. However, the buffer zone affects the spectra of the solution near the hard wall as it absorbs the rewrap wave. It has been found that this difference will remain stable as the solution develops. L_2 error results in Table 1 also indicate that the buffer zone does affect the accuracy of the Fourier based pseudospectral method. But for the majority of the domain, there are no significant differences in the solutions (see Figure C.5(b)). Figure C.6 also presents a comparison between the Fourier based pseudospectral time domain method and the prefactored compact scheme after 110 time iterations.

In terms of computation efficiency, the Fourier based pseudospectral method, with or without a buffer zone, represents a saving of up to 200% compared with the prefactored compact scheme.

C.4.2 Open Rotor

The second test case is that of the open rotor noise defined in category 2 benchmark problem at third computational aeroacoustics workshop [168]. It has nonlinear source terms. The original algorithm in the form of Eq. (C.13) is used to show its effectiveness for general problems.

The governing equations in cylindrical coordinates and problem definition are as follows:

$$\frac{\partial u}{\partial t} = -\frac{\partial p}{\partial x} + S_x, \quad (\text{C.22})$$

$$\frac{\partial v}{\partial t} = -\frac{\partial p}{\partial r} + S_r, \quad (\text{C.23})$$

$$\frac{\partial w}{\partial t} = -\frac{1}{r} \frac{\partial u}{\partial \Phi} + S_\Phi, \quad (\text{C.24})$$

$$\frac{\partial p}{\partial t} + \frac{\partial u}{\partial x} + \frac{1}{r} \frac{\partial v r}{\partial r} + \frac{1}{r} \frac{\partial w}{\partial \Phi} = 0, \quad (\text{C.25})$$

where x is the axial coordinate, r the radial coordinate and Φ the azimuthal angle. u, v, w are the velocity perturbations in the x, r, Φ directions respectively and p is the pressure perturbation. The no-linear terms are defined as:

$$S(x) = e^{-(\ln 2)(10x)^2}, \quad (\text{C.26})$$

$$S_r = 0, \quad (\text{C.27})$$

$$\begin{vmatrix} S_\Phi(r, \Phi, x, t) \\ S_x(r, \Phi, x, t) \end{vmatrix} = \text{Re} \left\{ \begin{vmatrix} \tilde{S}_\Phi(r, x) \\ \tilde{S}_x(r, x) \end{vmatrix} \right\} e^{iM(\Phi - \Omega t)}, \quad (\text{C.28})$$

$$\tilde{S}_x(r, x) = \begin{cases} S(x) J_M(\lambda_{(M,N)} r), & r \leq 1 \\ 0, & r > 1 \end{cases} \quad (\text{C.29})$$

$$\tilde{S}_\Phi(r, x) = \begin{cases} S(x) r J_M(\lambda_{(M,N)} r), & r \leq 1 \\ 0, & r > 1 \end{cases} \quad (\text{C.30})$$

where J_M is the M th-order Bessel function of first kind, λ_{MN} is the N th root of J'_M or $J'_M(\lambda_{MN}) = 0$. The parameters are $M = 8$, $N = 1$, $\lambda_{(8,1)} = 9.64742$ and $\Omega = 0.85$. The computation domain covers $[-5, 5]$ in the axial direction and the radial direction. The size of the grid Δx and the time step Δt obey the relation $\Delta t / \Delta x = 0.5$. Φ is set to 0. The governing equations are solved by the algorithm given in (C.13). The

time integration algorithm is the 4-6 low-dissipation and low-dispersion Runge-Kutta Scheme. The length of the buffer zone is 15 grid points.

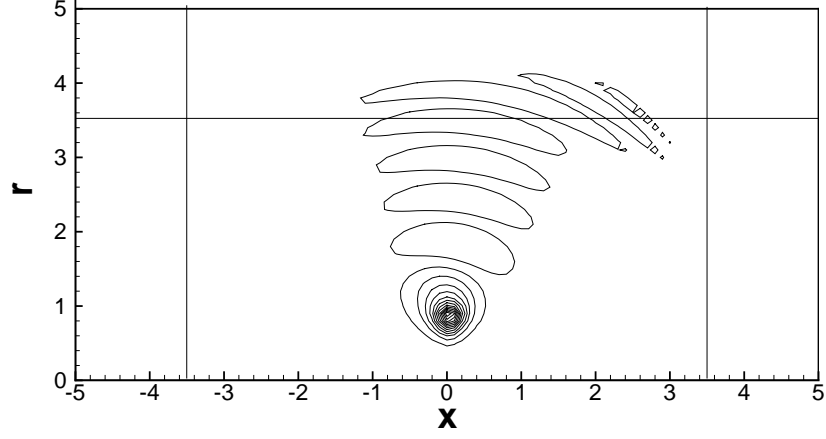


Figure C.7: Pressure contours generated by an open rotor.

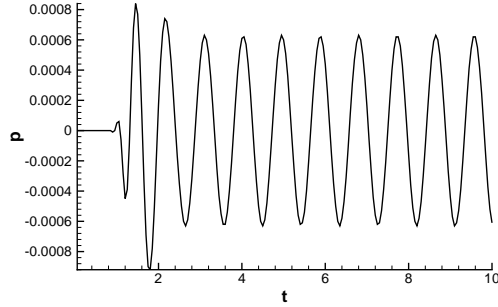


Figure C.8: Pressure time history generated by an open rotor at $(x = -0.1, r = 1.9)$.

Figure C.7 shows the pressure contours after 80 time steps. The monitored time history shown in Figure C.8 suggests that a time periodic state is reached. In spherical coordinates (r, θ, Φ) the directivity of the radiated sound is defined as:

$$D(\theta) = \lim_{r \rightarrow \infty} r^2 \overline{P^2(r, \Phi, \theta)},$$

where r , ϕ and θ are radius, azimuthal angle and polar angle respectively. $\overline{P^2(r, \Phi, \theta)}$ is the time average of $P^2(r, \theta, \Phi)$. The predicted directivity of rotor noise is compared with analytical solution [172] in Figure C.9. Here the radius r is set to 3. Although the radius is not large enough, two results still match well. The computing time t

C. FOURIER BASED PSEUDOSPECTRAL METHOD

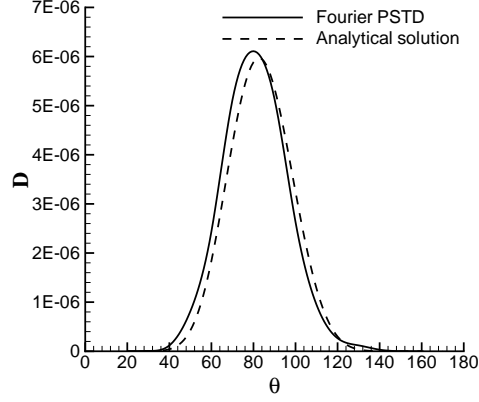


Figure C.9: Prediction and analytical solution of the directivity of an open rotor acoustic radiation, where $r = 3.0$ and θ is the polar angle.

and error Err of DRP, Compact and the Fourier based pseudospectral schemes are compared in Table 2, where

$$Err = \{\sum_{i=1}^8 |D(\theta_i) - D_{analytical}(\theta)|, \theta \in [20^\circ - 160^\circ]\}.$$

Table C.2: Computing time t and L_1 error of the second benchmark problem.

Schemes	100×100	200×200
DRP scheme	$t = 21.6s, Err = 7.9e^{-6}$	$t = 186s, Err = 6.1e^{-6}$
Compact scheme	$t = 34.1s, Err = 6.8e^{-6}$	$t = 557s, Err = 5.9e^{-6}$
Pseudospectral	$t = 20.7s, Err = 2.7e^{-6}$	$t = 180s, Err = 2.0e^{-6}$

In this case, the pseudospectral method is only employed to obtain the spatial differential terms. The numerical error affiliated with the buffer zone and the slip wall boundary condition does not exist. Consequently, results in Table 2 indicate that the Fourier based pseudospectral time domain method can obtain more accurate solutions with smaller points-per-wavelength numbers. It is also much more efficient than other high-order finite difference methods.

C.4.3 Summary

The Fourier based pseudospectral time domain method has been applied to some benchmark problems pertinent to computational aeroacoustics. For linear wave prop-

agation problems, a new algorithm has been developed and tested. The updated time stepping method relaxes time stepping restrictions. A hard-wall boundary condition is supplied for simple geometry and has been validated. Combined with the buffer zone technique to reduce contamination due to wave rewrap, the algorithm becomes fully-fledged to some linear wave propagation problems. For general problems with nonlinear terms, the original algorithm of the Fourier based pseudospectral time domain method is shown to be an alternative to high-order finite difference methods.

***Simplified Code for Two-dimensional Wave Equations**

```
function [x,y,u,v,p,clocka,clockb]=Wave2DFreqBCRewrap2(dx,ntsteps)

%*****%

%   x,y:grid; u,v,p:velocity and pressure;
%   dx:spatial step; ntsteps:operation steps;
%*****%

RK_dt=dx*0.3; %CFL, RK_dt=0.1, 0.05,and 0.025 have been tested before.
[x,y]=meshgrid(-0.8+dx:dx:0.8); %make grids.
dimen=size(x); dimen=dimen(1);
u=0*x;v=0*x;p=exp(-log(2.0)*(x.^2+(y+0.6).^2)/0.006);
Uifft=u;Vifft=v;
omegaT1=[0.25,0.333333333333,0.5,1.0]*RK_dt;    %Runge-Kutta coef
inv_dimen=1/dimen; inv_dx=1/dx;max_half=dimen/2;
tmp=-sqrt(-1)*2*pi*inv_dimen*inv_dx;
Vall=zeros(1,dimen);    %Vall is used for hard wall reflection.
totalstep=0; clocka=clock;
P=fft2(p,dimen,dimen);U=fft2(u,dimen,dimen);V=fft2(v,dimen,dimen);
U=fftshift(U);V=fftshift(V);P=fftshift(P); P0=P;U0=U;V0=V;
while(totalstep<ntsteps)
    for s=1:1:1
```

C. FOURIER BASED PSEUDOSPECTRAL METHOD

```
for subit=1:1:4
    for m=1:1:dimen
        for n=1:1:dimen
            Tmp_Px=tmp*(n-max_half-1)*P(m,n);
            Tmp_Py=tmp*(m-max_half-1)*P(m,n);
            Tmp_Ux=tmp*(n-max_half-1)*U(m,n);
            Tmp_Vy=tmp*(m-max_half-1)*V(m,n)-2*Vall(1,n)*inv_dx;
            U(m,n)=U0(m,n) + Tmp_Px*omegaT1(subit);
            V(m,n)=V0(m,n) + Tmp_Py*omegaT1(subit);
            P(m,n)=P0(m,n) + (Tmp_Ux+Tmp_Vy)*omegaT1(subit);
        end
    end
    Vifft=ifft2(V);Vall=fft(Vifft(1,:));
end
U0=U;V0=V;P0=P;totalstep=totalstep+1      %Update all
end
end clockb=clock; U0=fftshift(U0);V0=fftshift(V0);P0=fftshift(P0);
Uifft=ifft2(U0);Vifft=ifft2(V0);Pifft=ifft2(P0); %inverse FFT
u=real(Uifft);v=real(Vifft);p=real(Pifft);
```

Bibliography

- [1] G. Raman and D. K. McLaughlin D. K., “Recent aeroacoustics research in the United States,” *Noise & Vibration Worldwide*, Vol. 31, No. 10, 2000, pp. 15–20.
- [2] Advisory Council for Aeronautics Research in Europe, “Strategic Research Agenda,” Vol. 1 and 2, 2002.
- [3] M. J. Fisher and R. H. Self, “Aeroacoustics Research in Europe: The CEAS-ASC Report on 2001 Highlights,” *Journal of Sound and Vibration*, Vol. 258, No. 1, 2002, pp. 1–30.
- [4] V. L. Wells and R. A. Renaut, “Computational Aerodynamically Generated Noise,” *Annual Review of Fluid Mechanics*, Vol. 29, 1997, pp. 161–199.
- [5] S. K. Lele, “Computational Aeroacoustics: A Review,” AIAA Paper 1997-0018, 1997.
- [6] F. Bastin, P. Lafon, and S. Candel, “Computation of Jet Mixing Noise Due to Coherent Structures: the Plane Jet Case,” *Journal of Fluid Mechanics*, Vol. 335, 1997, pp. 261–304.
- [7] X. Zhang, X. X. Chen, C. L. Morfey, and P. A. Nelson, “Computation of Spinning Modal Radiation from an Unflanged Duct,” *AIAA Journal*, Vol. 42, No. 9, 2004, pp. 1795–1801.
- [8] S. Richards, X. Zhang, and X. X. Chen, “Acoustic Radiation Computation from an Engine Inlet with Aerodynamic Flow Field,” AIAA Paper 2004-0848, 2004.
- [9] M. R. Visbal and D. V. Gaitonde, “Very High-Order Spatially Implicit Schemes for Computational Acoustics on Curvilinear Meshes,” *Journal of Computational Acoustics*, Vol. 9, No. 4, 2001, pp. 1259–1286.

- [10] M. Berger, M. Aftosmis, and J. Melton, “Accuracy, Adaptive Methods and Complex Geometry,” Proceedings 1st AFOSR Conference on Dynamic Motion CFD, 1996.
- [11] M. J. Berger and J. Oliger, “Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations,” *Journal of Computational Physics*, Vol. 53, 1983, pp. 484–512.
- [12] M. J. Berger and A. Jameson, “Automatic Adaptive Grid Refinement for the Euler Equations,” AIAA Paper 1985–1633, 1985.
- [13] M. J. Berger and P. Colella, “Local Adaptive Mesh Refinement for Shock Hydrodynamics,” *Journal of Computational Physics*, Vol. 82, 1989, pp. 64–84.
- [14] T. A. Driscoll, K. G. Powell, D. L. De Zeeuw, C. R. Clauer, K. C. Hansen, W. B. Manchester, A. J. Ridley, I. I. Roussev, I. V. Sokolov, Q. F. Stout, and G. Toth, “Solution-Adaptive Magnetohydrodynamics for Space Plasmas: Sun-to-Earth Simulations,” *Computing in Science and Engineering*, Vol. 6, No. 2, 2004, pp. 14–35.
- [15] T. I. Gombosi, D. L. De Zeeuw, K. G. Powell, I. V. Sokolov, Q. F. Stout, and G. Tóth, “Adaptive Mesh Refinement MHD for Space Plasma Simulations,” International Sherwood Fusion Theory Conference, 2002.
- [16] R. Samtaney, “Adaptive Mesh Simulations of Pellet Injection in Tokamaks,” SIAM Annual Conference, 2006.
- [17] D. L. Marcus, R. B. Pember, J. B. Bell, V. E. Beckner, D. Simkins, and M. Welcome, “Multidimensional Numerical Simulation of a Pulse Combustor,” AIAA 1994–2351, 1994.
- [18] J. Ray, C. A. Kennedy, S. Lefantzi, and H. N. Najm, “High-order Spatial Discretizations and Extended Stability Methods for Reacting Flows on Structured Adaptively Refined Meshes,” Proceedings of the Third Joint Meeting of the U.S. Sections of the Combustion Institute, 2003.
- [19] G. L. Bryan and M. L. Norman, “A Hybrid AMR Application for Cosmology and Astrophysics,” In Proceedings of the Workshop on Structured Adaptive Mesh Refinement Grid Methods, 1997.
- [20] G. L. Bryan, “Fluids in the Universe: Adaptive Mesh Refinement in Cosmology,” *Computing in Science and Engineering*, Vol. 1, No. 2, 1999, pp. 46–53.

- [21] Y. Yang, M. N. Wernick, and J. G. Brankov, "A Fast Algorithm for Accurate Content-adaptive Mesh Generation," Proceedings of IEEE International Conference on Image Processing (ICIP), 2001.
- [22] S. R. Kohn and S. B. Baden, "A Parallel Software Infrastructure for Structured Adaptive Mesh Methods," ACM/IEEE SC Conference, 1995.
- [23] A. Jameson and J. Vassberg, "Computational Fluid Dynamics (CFD) for Aerodynamic Design: Its Current and Future Impact," AIAA Paper 2001-0538, 2001.
- [24] S. Kohn, J. Weare, M. Ong, and S. Baden, "Software Abstractions and Computational Issues in Parallel Structured Adaptive Mesh Methods for Electronic Structure Calculations," In Proceedings of the Workshop on Structured Adaptive Mesh Refinement Grid Methods, 1997.
- [25] S. R. Kohn, "Parallel Software Abstractions for Structured Adaptive Mesh Methods," *Journal of Parallel and Distributed Computing*, Vol. 61, 2001, pp. 713–736.
- [26] M. Parashar and J. C. Browne, "Object-Oriented Programming Abstractions for Parallel Adaptive Mesh-Refinement," In POOMA, 1996.
- [27] M. Parashar and J. C. Browne, "Distributed Dynamic Data-structures for Parallel Adaptive Mesh Refinement," In HiPC, 1995.
- [28] J. Nieplocha, R. J. Harrison, and R. J. Littlefield, "The Global Array: Non-Uniform-Memory-Access Programming Model for High-Performance Computers," *The Journal of Supercomputing*, Vol. 10, 1996, pp. 197–220.
- [29] K. Yelick, L. Semenzato, G. Pike, C. Miyamoto, B. Liblit, A. Krishnamurthy, P. Hilfinger, S. Graham, D. Gay, P. Colella, and A. Aiken, "Titanium: A High-Performance Java Dialect," Workshop on Java for High-Performance Network Computing, 1998.
- [30] W. W. Carlson, J. M. Draper, D. E. Culler, K. Yelick, E. Brooks, and K. Warren, "Introduction to UPC and Language Specification," CCS-TR-99-157, IDA/CCS, 1999.
- [31] S. J. Sherwin, O. Shah J. Peiro, G-S. Karamanos, and D. J. Doorly, "Computational Haemodynamics: Geometry and Non-Newtonian Modelling using

- Spectral/hp Element Methods,” *Computing and Visualization in Science*, Vol. 3, 2000, pp. 77–83.
- [32] P. C. Walsh and D. W. Zingg, “Solution Adaptation of Unstructured Grids for Two-Dimensional Aerodynamic Computations,” *AIAA Journal*, Vol. 39, No. 5, 2001, pp. 831–837.
- [33] V. Dolean and S. Lanteri, “Parallel Multigrid Methods for the Calculation of Unsteady Flows on Unstructured Grids: Algorithmic Aspects and Parallel Performances on Clusters of PCs,” *Parallel Computing*.
- [34] W. C. Huang and D. K. Tafti, “A Parallel Adaptive Mesh Refinement Algorithm for Solving Nonlinear Dynamical Systems,” *The International Journal of High Performance Computing Applications*, Vol. 18, No. 2, 2004, pp. 171–181.
- [35] Y. Le Moigne, “Adaptive Mesh Refinement Sensors for Vortex Flow Simulations,” European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS, 2004.
- [36] A. M. Roma, C. S. Peskin, and M. J. Bergery, “An Adaptive Version of the Immersed Boundary Method,” *Journal of Computational Physics*, Vol. 153, 1999, pp. 509–534.
- [37] M. J. Aftosmis, M. J. Berger, and G. Adomavicius, “A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries,” AIAA Paper 2000–0808, 2000.
- [38] F. Alauzet, P. J. Frey, and P. L. George, “Anisotropic Mesh Adaptation for Rayleigh-taylor Instabilities,” European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS, 2004.
- [39] M. J. Berger and M. J. Aftosmis, “Multilevel Error Estimation and Adaptive h -Refinement for Cartesian Meshes with Embedded Boundaries,” AIAA Paper 2002–0863, 2002.
- [40] A. Kurganov, S. Karni, and G. Petrova, “A Smoothness Indicator for Adaptive Algorithms for Hyperbolic Systems,” *Journal of Computational Physics*, Vol. 178, 2002, pp. 323–341.
- [41] X. Huang and X. Zhang, “Adaptive Mesh Refinement for Computational Aeroacoustics,” AIAA Paper 2005–2873, 2005.

- [42] J. Ray, C. A. Kennedy, S. Lefantzi, and H. N. Najm, “Using High-order Methods on Adaptively Refined Block-structured Meshes - Discretizations, Interpolations, and Filters,” SAND2005–7981, Sandia National Laboratories, 2006.
- [43] D. C. Arney and J. E. Flaherty, “An Adaptive Mesh-Moving and Local Refinement Method for Time-Dependent Partial Differential Equations,” *ACM Transactions on Mathematical Software*, Vol. 16, No. 1, 1990.
- [44] J. J. Quirk, “An Adaptive Grid Algorithm for Computational Shock Hydrodynamics,” PhD thesis, College of Aeronautics, Cranfield Institute of Technology, 1991.
- [45] W. J. Coirier, “An Adaptively-Refined, Cartesian, Cell-Based Scheme for the Euler and Navier-Stokes Equations,” PhD dissertation, Aerospace Engineering, The University of Michigan, 1994.
- [46] A. S. Almgren, J. B. Bell, L. H. Howell, and P. Colella, “An Adaptive Projection Method for the Incompressible Navier-Stokes Equations,” Proceedings of the 14th IMACS World Congress, 1994.
- [47] A. S. Almgren, T. Buttke, and P. Colella, “A Fast Adaptive Vortex Method In Three Dimensions,” *Journal of Computational Physics*, Vol. 113, No. 2, 1994, pp. 177–200.
- [48] P. Colella and W. Y. Crutchfield, “A Parallel Adaptive Mesh Refinement Algorithm on the C-90,” Energy Research Power Users Symposium, July 12, 1994.
- [49] J. A. Greenough, B. de Supinski, R. K. Yates, C. Rendleman, D. Skinner, V. Beckner, M. Lijewski, and J. Bell, “Performance of a Block Structured, Hierarchical Adaptive Mesh Refinement Code on the 64K Node IBM BlueGene/L Computer,” LBNL Report LBNL-57500, 2005.
- [50] M. Welcome, C. Rendleman, L. Okiker, and R. Biswas, “Performance Characteristics of an Adaptive Mesh Refinement Calculation on Scalar and Vector Platforms,” International Conference on Computing Frontiers, 2006.
- [51] J. B. Bell, M. S. Day, A. S. Almgren, M. J. Lijewski, and C. A. Rendleman, “A Parallel Adaptive Projection Method for Low Mach Number Flows,” *International Journal for Numerical Methods in Fluids*, Vol. 40, 2002, pp. 209–216.
- [52] C. A. Rendleman, V. E. Beckner, and M. J. Lijewski, “Parallelization of an Adaptive Mesh Refinement Method for Low Mach Combustion,” Proceedings Computational Science – ICCS 2001, San Francisco, CA, 2001.

- [53] M. J. Berger and R. J. LeVeque, “Adaptive Mesh Refinement Using Wave-Propagation Algorithms for Hyperbolic Systems,” *SIAM Journal on Numerical Analysis*, Vol. 35, No. 6, 1998, pp. 229–231.
- [54] J. B. Bell, M. S. Day, C. A. Rendleman, S. E. Woosley, and M. A. Zingale, “Adaptive low Mach Number Simulations of Nuclear Flame Microphysics,” *Journal of Computational Physics*, Vol. 195, 2004, pp. 677–694.
- [55] J. B. Bell, “AMR for low Mach number reacting flows,” Proceedings of the Chicago Workshop on Adaptive Mesh Refinement Methods, 2003.
- [56] J. B. Bell, M. S. Day, J. F. Grcar, M. J. Lijewski, J. F. Driscoll, and S. F. Filatyev, “Numerical Simulation of a Laboratory-Scale Turbulent Slot Flame,” LBNL Report LBNL-59245, 2005.
- [57] P. MacNeice, K. M. Olson, C. Mobarry, R. de Fainchtein, and C. Packer, “PARAMESH: A Parallel Adaptive Mesh Refinement Community Toolkit,” *Computer Physics Communications*, Vol. 126, 2000, pp. 330–354.
- [58] A. Wissink and R. Hornung, “Parallel AMR Application Development with the SAMRAI Library,” 2005, LLNL technical report UCRL-PRES-209469.
- [59] Jürgen Dreher and R. Grauer, “Racoon: A parallel Mesh-adaptive Framework for Hyperbolic Conservation Laws,” *Parallel Computing*, Vol. 31, No. 8–9, 2005, pp. 913–932.
- [60] Stéphane Popinet, “Gerris: A Tree-based Adaptive Solver for the Incompressible Euler Equations in Complex Geometries,” *Journal of Computational Physics*, Vol. 190, No. 2, 2003, pp. 572–600.
- [61] Wikipedia, “Granularity,” <http://en.wikipedia.org/wiki/Granularity>.
- [62] J. S. Sachdev, C. P. T. Groth, and J. J. Gottlieb, “A Parallel Solution-Adaptive Scheme for Predicting Multi-Phase Core Flows in Solid Propellant Rocket Motors,” *International Journal of Computational Fluid Dynamics*, Vol. 19, No. 2, 2005, pp. 157–175.
- [63] J. Ray, C. Kennedy, J. Steensland, and H. Najm, “Advanced Algorithms for Computations on Block-structured Adaptively Refined Meshes,” *Institute of Physics Conference Series*, Vol. 16, 2005, pp. 113–118.

- [64] M. T. Jones and P. E. Plassmann, “Parallel Algorithms for Adaptive Mesh Refinement,” *SIAM Journal on Scientific Computing*, Vol. 18, No. 3, 1997, pp. 686–708.
- [65] M. Parashar, J. C. Browne, C. Edwards, and K. Klimkowski, “A Common Data Management Infrastructure for Adaptive Algorithms for PDE Solutions,” IEEE SC97 Conference, 1997.
- [66] D. De Zeeuw and K. G. Powell, “An Adaptively Refined Cartesian Mesh Solver for the Euler Equations,” *Journal of Computational Physics*, Vol. 104, 1993, pp. 56–68.
- [67] D. Edelsohn, “Hierarchical Tree-structures as Adaptive Meshes,” *International Journal of Modern Physics C*, Vol. 4, No. 5, 1993.
- [68] J. E. Flaherty, R. M. Loy, M. S. Shephard, B. K. Szymanski, J. D. Teresco, and L. H. Ziantz, “Adaptive Local Refinement with Octree Load Balancing for the Parallel Solution of Three-Dimensional Conservation Laws,” *Journal of Parallel and Distributed Computing*, Vol. 47, No. 2, 1997, pp. 139–152.
- [69] B. Hariharan and S. Aluru, “Efficient Parallel Algorithms and Software for Compressed Octrees with Applications to Hierarchical Methods,” *Parallel Computing*, Vol. 31, 2005, pp. 311–331.
- [70] W. Y. Crutchfield and M. Welcome, “Object Oriented Implementation of Adaptive Mesh Refinement Algorithms,” *Scientific Programming*, Vol. 2, No. 4, 1993, pp. 145–156.
- [71] M. Parashar and J. C. Browne, “System Engineering for High Performance Computing Software: The HDDA/DAGH Infrastructure for Implementation of Parallel Structured Adaptive Mesh Refinement,” Vol. 117: Structured Adaptive Mesh Refinement Grid Methods, IMA Volumes in Mathematics and its Applications, Springer-Verlag, 2000.
- [72] Q. L. Dan, “AMR++: Object-Oriented Design for Adaptive Mesh Refinement,” 1998, HPC 98, Boston, Massachusetts.
- [73] R. Deiterding, “Object-oriented Design of an AMR-algorithm for Distributed Memory Computers,” 8th International Conference on Hyperbolic Problems, 2000.

- [74] P. Colella, D. T. Graves, T. J. Ligocki, D. F. Martin, D. Modiano, D. B. Serafini, and B. Van Straalen, “Chombo Software Package for AMR Applications,” Lawrence Berkeley National Laboratory, 2003.
- [75] L. H. Howell and J. B. Bell, “An Adaptive Mesh Projection Method for Viscous Incompressible Flow,” *SIAM Journal on Scientific Computing*, Vol. 18, No. 4, 1997, pp. 996–101.
- [76] P. Mehrotra and H. Zima, “High Performance Fortran for aerospace applications,” *Parallel Computing*, Vol. 27, No. 4, 2001, pp. 477–501.
- [77] A. Legrand, H. Renard, Y. Robert, and F. Vivien, “Load-balancing iterative computations in heterogeneous clusters with shared communication links,” Research Report RR-2003-23, 2003.
- [78] J. E. Flaherty, R. M. Loy, C. Özturan, M. S. Shephard, B. K. Szymanski, J. D. Teresco, and L. H. Ziantz, “Parallel structures and dynamic load balancing for adaptive finite element computation,” *Applied Numerical Mathematics: Transactions of IMACS*, Vol. 26, No. 1–2, 1997, pp. 241–263.
- [79] M. Parashar and J. Browne, “Distributed Dynamic Data-Structures for Parallel Adaptive Mesh Refinement,” Proceedings of the International Conference on High Performance Computing, 1995.
- [80] Z. L. Lan, V. E. Taylor, and G. Bryan, “Dynamic Load Balancing for Structured Adaptive Mesh Refinement Applications,” IEEE SC2001 Conference, 2001.
- [81] J. Steensland, S. Chandra, and M. Parashar, “An Application-Centric Characterization of Domain-Based SFC Partitioners for Parallel SAMR,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 12, 2002.
- [82] A. Wissink, “Algorithmic Issues for Scaling Structured AMR Calculations to Thousands of Processors,” SIAM CSE05 meeting, also available as LLNL technical report UCRL-PRES-209446, 2005.
- [83] A. C. Calder, B. C. Curtis, L. J. Dursi, B. Fryxell, G. Henry, P. MacNeice, K. Olson, P. Ricker¹, R. Rosner, F. X. Timmes, H. M. Tufo, J. W. Truran, and M. Zingale¹, “High-Performance Reactive Fluid Flow Simulations Using Adaptive Mesh Refinement on Thousands of Processors,” IEEE SC2000 Conference, 2000.

- [84] J. J. Quirk, “A Cartesian Grid Approach with Hierarchical Refinement for Compressible Flows,” NASA CR-194938 ICASE Report No. 94-51, 1994.
- [85] Dae-II Choi, J. D. Brown, B. Imbiriba, J. Centrella, and P. MacNeice, “Interface Conditions for Wave Propagation through Mesh Refinement Boundaries,” *Journal of Computational Physics*, Vol. 193, No. 2, 2004, pp. 398–425.
- [86] M. Sun and K. Takayama, “Conservative Smoothing on an Adaptive Quadrilateral Grid,” *Journal of Computational Physics*, Vol. 150, 1999, pp. 143–180.
- [87] W. D. Henshaw and D. W. Schwendeman, “An Adaptive Numerical Scheme for High-speed Reactive Flow on Overlapping Grids,” *Journal of Computational Physics*, Vol. 191, 2003, pp. 420–447.
- [88] D. T. Thorne, “Cache Aware Multigrid on AMR Hierarchies,” The Virtual Proceedings of the Copper Mountain Conference on Multigrid Methods, 2003.
- [89] S. T. Li and H. Li, “A Novel Approach of Divergence-free Reconstruction for Adaptive Mesh Refinement,” *Journal of Computational Physics*, Vol. 199, 2004, pp. 1–15.
- [90] R. B. Pember, J. B. Bell, W. Y. Crutchfield P. Colella, and M. L. Welcome, “An Adaptive Cartesian Grid Method for Unsteady Compressible Flow in Irregular Regions,” *Journal of Computational Physics*, Vol. 120, 1995, pp. 278–304.
- [91] R. Mittal and G. Iaccarino, “Immersed Boundary Methods,” *Annual Reviews of Fluid Mechanics*, Vol. 37, 2005, pp. 239–361.
- [92] F. E. Ham, F. S. Lien, and A. B. Strong, “A Cartesian Grid Method with Transient Anisotropic Adaptation,” *Journal of Computational Physics*, Vol. 179, 2002, pp. 469–494.
- [93] J. Mohd-Yusof, “Development of Immersed Boundary Methods for Complex Geometries,” Technical Rreport, Center for Turbulence Research Annual Research Briefs, Stanford University, 1998.
- [94] S. Majumdar, G. Iaccarino, and P. Durbin, “RANS Solvers with Adaptive Structured Boundary Non-conforming Grids,” Technical Report, Center for Turbulence Research Annual Research Briefs, Stanford University, 2001.
- [95] W. D. Henshaw and D. W. Schwendeman, “Moving Overlapping Grids with Adaptive Mesh Refinement for High-Speed Reactive and Non-reactive Flow,” *Journal of Computational Physics*, Vol. 216, 2006, pp. 744–779.

- [96] D. L. Brown, W. D. Henshaw, and D. J. Quinlan, “Overture: An Object Oriented Framework for Solving Partial Differential Equations,” ISCOPE conference, 1997.
- [97] X. Huang and X. Zhang, “Adaptive Mesh Refinement Computation of Acoustic Radiation from an Engine Intake,” AIAA Paper 2006-2694, 2006.
- [98] W. D. Henshaw, “A High-Order Accurate Parallel Solver for Maxwell’s Equations on Overlapping Grids,” UCRL-JRNL-215684, 2005.
- [99] S. K. Lele, “Compact Finite Difference Schemes with Spectral-Like Resolution,” *Journal of Computational Physics*, Vol. 103, 1992, pp. 16–42.
- [100] R. E. Gordnier and M. R. Visbal, “Compact Difference Scheme Applied to Simulation of Low-Sweep Delta Wing Flow,” *AIAA Journal*, Vol. 43, No. 8, 2005, pp. 1744–1752.
- [101] R. Hixon, “Prefactored Small-Stencil Compact Schemes,” *Journal of Computational Physics*, Vol. 165, No. 2, 2000, pp. 522–541.
- [102] G. Ashcroft and X. Zhang, “Optimized Prefactored Compact Schemes,” *Journal of Computational Physics*, Vol. 190, No. 2, 2003.
- [103] T. Beck, “Multigrid High Order Mesh Refinement Technique,” Preprint, Department of Chemistry, University of Cincinnati, 1997.
- [104] S. Lefantzi, C. A. Kennedy, J. Ray, and H. N. Najm, “A Study of the Effect of Higher Order Spatial Discretizations in SAMR (Structured Adaptive Mesh Refinement) Simulations,” Proceedings of the Fall Meeting of the Western States Section of the Combustion Institute, 2003.
- [105] R. J. LeVeque, “CLAWPACK Version 4.2 User’s Guide,” <http://www.amath.washington.edu/~claw/>.
- [106] C. A. Rendleman, V. E. Beckner, M. Lijewski, W. Y. Crutchfield, and J. B. Bell, “Parallelization of Structured, Hierarchical Adaptive Mesh Refinement Algorithms,” *Computing and Visualization in Science*, Vol. 3, 2000, pp. 147–157.
- [107] D. S. Balsara and C. D. Norton, “Highly Parallel Structured Adaptive Mesh Refinement Using Parallel Language-based Approaches,” *Parallel Computing*, Vol. 27, No. 1–2, 2001, pp. 37–70.

- [108] S. K. Das, D. J. Harvey, and R. Biswas, “Parallel Processing of Adaptive Meshes with Load Balancing,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 12, 2001, pp. 1269–1280.
- [109] C. K. W. Tam and J. C. Webb, “Dispersion-Relation-Preserving Finite Difference Schemes for Computational Acoustics,” *Journal of Computational Physics*, Vol. 107, No. 2, 1993, pp. 262–281.
- [110] C. K. W. Tam and K. A. Kurbatskii, “Multi-size-mesh Multi-time-step Dispersion-relation-preserving Scheme for Multiple-Scales Aeroacoustics Problems,” *International Journal of Computational Fluid Dynamics*, Vol. 17, No. 2, 2003, pp. 119–132.
- [111] F. Q. Hu, M. Y. Hussaini, and J. Manthey, “Low-Dissipation and Low-Dispersion Runge-Kutta Schemes for Computational Acoustics,” NASA CR-195022, 1994.
- [112] R. Hixon, “Prefactored Compact Filters for Computational Aeroacoustics,” AIAA Paper 1999-358, 1999.
- [113] J. von Neumann and R. D Richtmyer, “A method for the numerical calculation of hydrodynamic shocks,” *J. Appl. Phys.*, Vol. 21, 1950, pp. 232–247.
- [114] A. Jameson, W. Schmidt, and E. Turkel, “Numerical Simulation of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes,” AIAA Paper 1981-1259, 1981.
- [115] J. W. Kim and D. J. Lee, “Adaptive Nonlinear Artificial Dissipation Model for Computational Aeroacoustics,” *AIAA Journal*, Vol. 39, No. 5, 2001, pp. 810–818.
- [116] M. J. Lighthill, “On Sound Generated Aerodynamically, 1,” *In Proceedings of the Royal Society of London*, Vol. A221, 1952, pp. 564–587.
- [117] M. J. Lighthill, “On Sound Generated Aerodynamically, 2,” *In Proceedings of the Royal Society of London*, Vol. A222, 1954, pp. 1–32.
- [118] J. E. Ffowcs-Williams and D. L. Hawkings, “Sound Generation by Turbulence and Surfaces in Arbitrary Motion,” *Philosophical Transactions of the Royal Society of London*, Vol. A264, 1969, pp. 321–342.

- [119] G. B. Ashcroft, "A Computational and Experimental Investigation into the Aeroacoustics of Low Speed Flows," PhD thesis, School of Engineering Sciences, University of Southampton, 2003.
- [120] J. M. Tyler and T. G. Sofrin, "Axial Flow Compressor Noise Studies," *SAE Transactions*, Vol. 70, 1962, pp. 309–332.
- [121] A. M. Cargill, "The Radiation of High Frequency Sound out of a Jet Pipe," *Journal of Sound and Vibration*, Vol. 83, No. 3, 1982, pp. 313–337.
- [122] C. K. W. Tam and J. C. Hardin (Eds.), "Second Computational Aeroacoustics Workshop on Benchmark Problems," NASA CP-3352, 1997.
- [123] X. Zhang, X. X. Chen, and C. L. Morfey, "Acoustic Radiation from a Semi-infinite Duct with a Subsonic Jet," *International Journal of Aeroacoustics*, Vol. 4, No. 1-2, 2005, pp. 169–184.
- [124] S. K. Richards, X. X. Chen, and X. Zhang, "Computation of Mode Radiation From a Generic Aeroengine Intake," European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2004, 2004.
- [125] S. K. Richards, X. X. Chen, and X. Zhang, "Parallel Computation of 3D Acoustic Radiation from an Engine Intake," AIAA Paper 2005–2947, 2005.
- [126] X. Zhang, X. X. Chen, C. L. Morfey, and B. J. Tester, "Computation of Fan Noise Radiation through an Engine Exhaust Geometry with Flow," AIAA Paper 2003–3267, 2003.
- [127] A. Jameson, "Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils, Wings, and Helicopter Rotors," AIAA Paper 1991–1596, 1991.
- [128] G. C. Fox, R. D. Williams, and P. C. Messina, *Parallel Computing Works*, Morgan Kaufmann Publishers, Inc., 1994.
- [129] Message Passing Interface Forum, "MPI: A Message-Passing Interface standard (version 1.1)," Technical Report, <http://www.mpi-forum.org>, 1995.
- [130] R. Buyya (Ed.), *High-Performance Cluster Computing Volume 1: Architectures and Systems*, Prentice Hall PTR, Upper Saddle River, NJ,, 1999.
- [131] Beowulf.org, "Beowulf Cluster Site," <http://www.beowulf.org/>.

- [132] MPI Forum, “MPI Applications,” <http://www-unix.mcs.anl.gov/mpi/libraries.html>.
- [133] R. Varadarajan and I. Hwang, “An efficient dynamic load balancing algorithm for adaptive mesh refinement,” *Selected Areas in Cryptography*, 1994.
- [134] Wikipedia, “Space-filling curve,” http://en.wikipedia.org/wiki/Space-filling_curve.
- [135] Wikipedia, “Z-order (curve),” [http://en.wikipedia.org/wiki/Z-order_\(curve\)](http://en.wikipedia.org/wiki/Z-order_(curve)).
- [136] D. S. Wise and J. D. Frens, “Morton-order Matrices Deserve Compilers Support Technical Report 533,” Technical Report 533, 1999.
- [137] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in Fortran*, Cambridge University Press, 1993.
- [138] J. L. Hennessy and D. Goldberg, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann, 2002.
- [139] D. W. Zingg, “Aspects of linear stability analysis for higher-order finite-difference methods,” AIAA Paper 1997-1939, 1997.
- [140] M. Embree and L. N. Trefethen, “Pseudospectra Gateway,” Web site: <http://www.comlab.ox.ac.uk/pseudospectra>.
- [141] T. A. Driscoll and L. N. Trefethen, “Pseudospectra for the Wave Equation with An Absorbing Boundary,” *Journal of Computational and Applied Mathematics*, Vol. 69, 1996, pp. 125–142.
- [142] R. Hixon and R. R. Mankbadi, “Validation of a High-Order Prefactored Compact Scheme on Nonlinear Flows With Complex Geometries,” 3rd Computational Aeroacoustics(CAA) Workshop on Benchmark Problems, 2000.
- [143] C. A. Kennedy and M. H. Carpenter, “Comparison of Several Numerical Methods for Simulation of Compressible Shear Layers,” NASA Technical Paper 3484, 1997.
- [144] J. Fenlason and R. Stallman, “GNU gprof: The GNU Profiler,” <http://www.gnu.org/software/binutils/manual/gprof-2.9.1/>.
- [145] J. C. Hardin, J. R. Ristorcelli, and C. K. W. Tam (Eds.), “ICASE/LaRC Workshop on Benchmark Problems in Computational Aeroacoustics,” NASA CP-3300, 1995.

- [146] E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof, “Combined Immersed-Boundary Finite-Difference Methods Three-Dimensional Complex Flow Simulations,” *Journal of Computational Physics*, Vol. 161, 2000, pp. 35–60.
- [147] S. K. Richards, X. Zhang, X. X. Chen, and P. A. Nelson, “Evaluation of Non-Reflecting Boundary Conditions for Duct Acoustic Computation,” *Journal of Sound and Vibration*, Vol. 270, 2004, pp. 539–557.
- [148] C. K. W. Tam and Z. Dong, “Wall Boundary Conditions for High-Order Finite-Difference Schemes in Computational Aeroacoustics,” *Theoretical and Computational Fluid Dynamics*, Vol. 6, 1994, pp. 303–322.
- [149] G. F. Homicz and J. A. Lordi, “A Note on the Radiative Directivity Patterns of Duct Acoustic Modes,” *Journal of Sound and Vibration*, Vol. 41, 1975, pp. 283–290.
- [150] S. Lidoine, H. Batard, H. Troyes., S. Delnevo, and M. Roger, “Acoustic Radiation Modelling of Aeroengine Intake Comparison between Analytical and Numerical Methods,” AIAA Paper 2001–2140, 2001.
- [151] R. Astley, J. Hamilton, N. Baker, and E. Kitchen, “Modelling Tone Propagation from Turbofan Inlets - the Effect of Extended Lip Liners,” AIAA Paper 2002–2449, 2002.
- [152] S. K. Richards, X. Zhang, X. X. Chen, and P. A. Nelson, “Evaluation of Non-Reflecting Boundary Conditions for Duct Acoustic Computation,” *Journal of Sound and Vibration*, Vol. 270, 2004, pp. 539–557.
- [153] S. K. Richards, “Aeroacoustic Computation of Sound Radiation from Ducts,” PhD thesis, School of Engineering Sciences, University of Southampton, 2005.
- [154] C. Lorenz, “Complete System Performance Monitor HOWTO,” <http://www.tldp.org/HOWTO/archived/CSPM-HOWTO/>.
- [155] R. M. Munt, “The Interaction of Sound with a Subsonic Jet Issuing from a Semi-infinite Cylindrical Pipe,” *Journal of Fluids Mechanics*, Vol. 83, 1977, pp. 609–640.
- [156] R. Ewert and W. Schröder, “Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations,” *Journal of Computational Physics*, Vol. 188, 2003, pp. 365–398.

- [157] R. Ewert and W. Schröder, “On the Simulation of Trailing Edge Noise with a Hybrid LES/APE Method,” *Journal of Sound and Vibration*, Vol. 270, 2004, pp. 509–524.
- [158] X. X. Chen, “Numerical Prediction of Spinning Modal Radiation from A Bypass Duct,” Technical Rreport, ISVR, University of Southampton, 2002.
- [159] C. K. W. Tam, “Computational aeroacoustics: issues and methods,” *AIAA Journal*, Vol. 33, No. 10, 1995, pp. 1788–1796.
- [160] X. Huang and X. Zhang, “The Fourier Pseudospectral Time-Domain Method for Some Computational Aeroacoustics Problems,” *International Journal of Aeroacoustics*, Vol. 5, No. 3, 2006.
- [161] D. Kosloff and R. Kosloff, “A Fourier Method Solution for the Time Dependent Schrödinger Equation as a Tool in Molecular Dynamics,” *Journal of Computational Physics*, Vol. 52, 1983, pp. 35–53.
- [162] T. A. Driscoll and B. Fornberg, “A Block Pseudospectral Method for Maxwell’s Equations,” *Journal of Computational Physics*, Vol. 140, 1998, pp. 47–65.
- [163] Y. Wang and H. Takenaka, “A Multidomain Approach of the Fourier Pseudospectral Method using Discontinuous Grid for Elastic Wave Modeling,” *Earth Planets Space*, Vol. 53, 2001, pp. 149–158.
- [164] Q. H. Liu, “PML and PSTD Algorithm for Arbitrary Lossy Anisotropic Media,” *IEEE Microwave And Guided Wave Letters*, Vol. 9, No. 2, 1999, pp. 48–50.
- [165] B. Fornberg, *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, 1998.
- [166] Q. H. Liu and G. Zhao, “Review of PSTD Methods for Transient Electromagnetics,” *International J. Numerical Modeling*, Vol. 17, 2004, pp. 299–323.
- [167] Q. H. Liu, “The PSTD algorithm: A Time-domain Method Requiring Only Two Cells Per Wavelength,” *Microwave and Optical Technology Letters*, Vol. 15, No. 3, June 1997, pp. 158–165.
- [168] D. D. Milo, “Third Computational Aeroacoustics (CAA) Workshop on Benchmark Problems,” Tech. Rep. NASA CP-2000-209790, NASA, 2000.

- [169] J. P. He, L. F. Shen, Q. Zhang, and S. L. He, “A Pseudospectral Time-Domain Algorithm for Calculating the Band Structure of a Two-Dimensional Photonic Crystal,” *Chinese Phys. Lett.*, Vol. 19, No. 4, 2002, pp. 507–510.
- [170] J. Schulten, “On the Use of Characteristics in CAA,” AIAA Paper 2002–2584, 2002.
- [171] F. Farassat, “Generalized Functions and Kirchhoff Formulas,” AIAA Paper 1996–1705, 1996.
- [172] C. K. W. Tam, “Rotor Noise: Category 2 Analytical Solution,” *Third Computational Aeroacoustics (CAA) Workshop on Benchmark Problems*, Vol. 270, 2000, pp. 41–46.