

**The Pennsylvania State University  
The Graduate School**

**HIGH ORDER UNSTRUCTURED GRID METHODS FOR COMPUTATIONAL  
AEROACOUSTICS**

A Thesis in  
Aerospace Engineering  
by  
Preetham P. Rao

© 2004 Preetham P. Rao

Submitted in Partial Fulfillment  
of the Requirements  
for the Degree of

Doctor of Philosophy

May 2004

The thesis of Preetham P. Rao was reviewed and approved\* by the following:

Philip J. Morris  
Boeing/A.D. Welliver Professor of Aerospace Engineering  
Thesis Advisor  
Chair of Committee

Lyle N. Long  
Professor of Aerospace Engineering

Kenneth S. Brentner  
Associate Professor of Aerospace Engineering

Paul E. Plassmann  
Associate Professor of Computer Science and Engineering

Dennis K. McLaughlin  
Professor of Aerospace Engineering  
Head of the Department of Aerospace Engineering

\*Signatures are on file in the Graduate School.

# Abstract

Practical problems concerning the aeroacoustics of aircraft demand the use of unstructured grids, to handle the complex geometry boundaries. The work presented in this thesis develops, implements and optimizes high order accurate algorithms on unstructured grids using Finite Element Methods. The focus of the applications is on problems involving the propagation of sound in sheared mean flows. Methods are developed for both time and frequency domain solutions of the linearized Euler equations in two spatial dimensions. In the time domain, a solver using the Discontinuous Galerkin method is developed. A novel way to approximate the flux expressions in the linearized Euler equations is implemented. This reduces the computational cost considerably relative to existing Discontinuous Galerkin implementations. The two dimensional solver is designed to run on parallel computing platforms, and is validated by solving several benchmark problems. The parallel Discontinuous Galerkin solver is then used to simulate the acoustic propagation in the shear layer of a realistic aircraft jet engine exhaust configuration, modeled as a rectangular nozzle. In the frequency domain, the suppression of the instability wave solution of the linearized Euler equations is sought on unstructured grids. The implementation and evaluation of the Discontinuous Galerkin method for frequency domain applications is carried out. A continuous, Streamline Upwind Petrov Galerkin method is then used to obtain only the acoustic solution for a benchmark problem involving the instability waves in two dimensions. Unlike other existing frequency domain solvers, no assumption is made regarding the nature of the mean flow. This feature assures that the methods developed in this study are better suited for problems of practical interest.

# Table of Contents

|  |            |
|--|------------|
| <b>List of Figures</b>   | <b>vii</b> |
| <b>List of Tables</b>  | <b>x</b>   |
| <b>List of Symbols</b>   | <b>xi</b>  |
| <b>Acknowledgments</b>   | <b>xiv</b> |
| <b>1 Introduction</b>  | <b>1</b>   |
| 1.1 Computational Aeroacoustics . . . . .                              | 1          |
| 1.2 The DG method . . . . .  | 5          |
| 1.3 Calculations in the frequency domain . . . . .                     | 8          |
| 1.4 Motivation . . . . .   | 9          |
| 1.5 Organization of the thesis . . . . .                               | 10         |
| <b>2 The generalized DG method</b>                                     | <b>11</b>  |
| 2.1 Existing literature on the DG method . . . . .                     | 11         |
| 2.2 Formulation of the DG method . . . . .                             | 13         |
| 2.3 Application to the advection equation . . . . .                    | 18         |
| 2.3.1 One dimensional calculations . . . . .                           | 18         |
| 2.3.2 Two dimensional calculations . . . . .                           | 20         |
| 2.4 The choice of polynomials for the basis set . . . . .              | 22         |
| 2.4.1 Element volume matrices . . . . .                                | 25         |
| 2.4.2 Expansion of flux expressions . . . . .                          | 25         |
| The original quadrature-free method. . . . .                           | 27         |
| 2.4.3 Element edge integral calculation . . . . .                      | 28         |
| 2.5 A general polynomial basis set in quadrature-free method . . . . . | 30         |
| 2.5.1 Use of Lagrange polynomials for flux interpolation . . . . .     | 33         |

|          |   |           |
|----------|---|-----------|
| <b>3</b> | <b>Time domain calculations</b>   | <b>35</b> |
| 3.1      | DG method applied to the LEE in the time domain . . . . .               | 35        |
| 3.2      | Boundary conditions . . . . .   | 37        |
| 3.2.1    | Characteristic boundary conditions . . . . .                            | 38        |
| 3.2.2    | The PML technique . . . . .   | 39        |
|          | Background . . . . .  | 39        |
|          | Stabilized PML . . . . .  | 40        |
| 3.3      | Applications . . . . .  | 43        |
| 3.3.1    | Uniform mean flow - propagation of an initial pulse . . . . .           | 44        |
|          | $h - p$ refinement . . . . .  | 45        |
|          | Efficiency improvements from Lagrange interpolation . . . . .           | 49        |
|          | Dissipation parameter . . . . .   | 52        |
| 3.3.2    | Non-uniform mean flow - acoustic propagation in a shear layer . . . . . | 55        |
|          | Oscillations in higher order . . . . .                                  | 56        |
| 3.4      | Future directions . . . . .   | 57        |
| 3.4.1    | Interpolation error: the choice of nodes . . . . .                      | 57        |
| 3.4.2    | Extension of the algorithm to three dimensions . . . . .                | 58        |
|          | Preprocessing . . . . .   | 59        |
|          | Element volume matrices . . . . .                                       | 60        |
|          | Element surface integrals . . . . .                                     | 61        |
|          | PML calculations and postprocessing . . . . .                           | 61        |
| 3.4.3    | A note on the efficiency of the $3d$ solver . . . . .                   | 61        |
| <b>4</b> | <b>Parallel implementation</b>  | <b>64</b> |
| 4.1      | Existing literature on parallel-DG implementation. . . . .              | 65        |
| 4.2      | Domain decomposition . . . . .  | 66        |
| 4.3      | The algorithm . . . . .   | 66        |
| 4.4      | Pre-processor for parallel implementation . . . . .                     | 68        |
| 4.5      | Parallel implementation with PML . . . . .                              | 70        |
| 4.5.1    | Preprocessing for PML . . . . .   | 74        |
| 4.6      | Application issues for real flow problems . . . . .                     | 75        |
| 4.7      | Results from parallel implementation . . . . .                          | 76        |
| 4.7.1    | Acoustic scattering from the fuselage of an aircraft . . . . .          | 76        |
| 4.7.2    | Parallel performance . . . . .  | 79        |
| 4.7.3    | Acoustic propagation in the shear layer of a realistic jet . . . . .    | 82        |
| <b>5</b> | <b>Frequency domain calculations</b>                                    | <b>90</b> |
| 5.1      | Literature on the DG method for elliptic equations . . . . .            | 91        |
| 5.2      | DG method for the LEE in the frequency domain . . . . .                 | 92        |
| 5.2.1    | Boundary conditions . . . . .   | 94        |

|          |   |            |
|----------|---|------------|
| 5.2.2    | The direct solver . . . . .                                   | 95         |
| 5.3      | Applications . . . . .  | 96         |
| 5.3.1    | One dimensional calculations . . . . .                        | 96         |
| 5.3.2    | Two dimensional calculations . . . . .                        | 97         |
| 5.3.3    | Evaluation of the DG method . . . . .                         | 101        |
| 5.4      | Evaluation conclusions . . . . .                              | 102        |
| 5.5      | The Streamline Upwind Petrov Galerkin method . . . . .        | 103        |
| 5.6      | Literature on the SUPG method . . . . .                       | 106        |
| 5.7      | SUPG method for the LEE in the frequency domain . . . . .     | 107        |
| 5.7.1    | The discretized equation . . . . .                            | 109        |
| 5.7.2    | The boundary conditions . . . . .                             | 110        |
| 5.8      | Applications . . . . .  | 110        |
| 5.8.1    | One dimensional wave equation . . . . .                       | 110        |
| 5.8.2    | Two dimensional LEE . . . . .                                 | 112        |
| 5.9      | Observations and future directions . . . . .                  | 113        |
| <b>6</b> | <b>Conclusions</b>  | <b>116</b> |
| 6.1      | Suggestions for future work . . . . .                         | 118        |
|          | <b>Bibliography</b>   | <b>120</b> |
| <b>A</b> | <b>The DG formulation using a moment polynomial basis set</b> | <b>127</b> |
| A.1      | The mass matrix . . . . .                                     | 129        |
| A.2      | The stiffness matrix . . . . .                                | 129        |
| A.3      | The boundary integrals . . . . .                              | 130        |
| A.4      | The final equation . . . . .                                  | 132        |
| <b>B</b> | <b>Element vectors and matrices</b>                           | <b>133</b> |
| B.1      | The stiffness vector . . . . .                                | 133        |
| B.2      | The reaction vector . . . . .                                 | 134        |
| B.3      | The edge flux vector . . . . .                                | 135        |
| B.3.1    | The edge flux matrix . . . . .                                | 136        |
| <b>C</b> | <b>The Kronecker product and the row multiple</b>             | <b>139</b> |
| C.1      | The Kronecker product . . . . .                               | 139        |
| C.2      | The Row multiple . . . . .                                    | 140        |

# List of Figures

|      |   |    |
|------|---|----|
| 1.1  | The configuration of an internal mixer in a jet engine. (courtesy Rolls Royce Allison), from Mengle <i>et al.</i> [2]. . . . .                    | 4  |
| 1.2  | A conceptual diagram of the lobed internal mixer. . . . .   | 4  |
| 1.3  | Conforming and non-conforming elements. . . . .   | 7  |
| 2.1  | The mapping from the real coordinates to the transformation coordinates. . . . .  | 16 |
| 2.2  | One dimensional calculations: The linear advection equation. . . . .  | 19 |
| 2.3  | One dimensional calculations : The nonlinear Burger's equation. . . . .   | 19 |
| 2.4  | The initial condition in two dimensional advection equation: The sine wave pulse with the unit maximum value at the center of the domain. . . . . | 20 |
| 2.5  | The initial condition in two dimensional advection equation: A contour map of the initial condition with the direction of advection. . . . .      | 21 |
| 2.6  | Advection of the initial pulse. (a) $t = 0.2$ (b) $t = 0.8$ (c) $t = 1.4$ (d) $t = 2$ , advection through one cycle. . . . .                      | 22 |
| 2.7  | Error in the advection solution after a cycle: Global error $u - u_0$ . . . . .   | 23 |
| 2.8  | Error in the advection solution after a cycle: A slice of the solutions at $y_{x=0}$ at the end of one and two cycles of advection. . . . .       | 23 |
| 2.9  | The three parts of the computation time using the DG method . . . . .   | 24 |
| 2.10 | Transformation to the edge coordinates. . . . .   | 29 |
| 2.11 | Lagrange polynomials, $p = 2$ . . . . .   | 31 |
| 3.1  | Damping parameters in the PML. . . . .  | 40 |
| 3.2  | The time required using the PML. . . . .  | 43 |
| 3.3  | The computation domain: the grid for increasing degree of polynomials. . . . .  | 45 |
| 3.4  | The acoustic propagation after 100 time units. . . . .  | 46 |
| 3.5  | Propagation of the pulse: (a) with and (b) without the PML. . . . .   | 47 |
| 3.6  | Contour plot of the calculated and the exact pressure . . . . .   | 48 |
| 3.7  | Variation in the error between the calculated and the exact pressure . . . . .  | 49 |
| 3.8  | The solution. (a) Along the line $x = y$ . (b) the first peak magnified. . . . .  | 50 |
| 3.9  | The performance. (a) $L^\infty$ norm of the error. (b) The computation time. . . . .  | 51 |
| 3.10 | Ratio of computational times using Lagrange to moment polynomials. . . . .  | 52 |

|      |  |     |
|------|--|-----|
| 3.11 | The effect of the dissipation parameter $\alpha$ on the solution. (a) One dimensional calculations (b) $h$ -convergence in two dimensional calculations. | 54  |
| 3.12 | (a) The grid for the 5th benchmark problem domain. (b) The profile of mean flow in the clustered region. . . . .   | 56  |
| 3.13 | Solution to the 5th benchmark problem: Instantaneous pressure at $y = 1$ at the end of a source cycle. . . . .   | 57  |
| 3.14 | Solution to the 5th benchmark problem: $\overline{p^2}$ at $y = 10$ . . . . .  | 58  |
| 3.15 | Oscillations in the non-uniform mean flow region with a 2nd order basis set. (a) At the end of a cycle (b) after two cycles. . . . .                     | 59  |
| 4.1  | An example of a partition by METIS. . . . .  | 69  |
| 4.2  | A typical partition configuration. . . . .   | 70  |
| 4.3  | An example of partitions with PML. . . . .   | 72  |
| 4.4  | Preprocessing for PML partitions. . . . .  | 74  |
| 4.5  | Acoustic scattering by the fuselage - problem configuration. . . . .   | 76  |
| 4.6  | Acoustic scattering by the fuselage - the mesh. . . . .  | 77  |
| 4.7  | Acoustic scattering - the partitions from METIS. . . . .   | 78  |
| 4.8  | Pressure contour at $t = 10$ . . . . .   | 79  |
| 4.9  | Pressure history at points A(top), B(center), and C(bottom). . . . .   | 80  |
| 4.10 | A typical partition used for the parallel performance study. . . . .   | 81  |
| 4.11 | Parallel speedup for the pulse propagation problem. . . . .  | 82  |
| 4.12 | Parallel efficiency for the pulse propagation problem. . . . .   | 83  |
| 4.13 | The structured grid used in the mean flow solution. Courtesy Steve Zygmunt [53]. . . . .   | 84  |
| 4.14 | Contour plot of the mean flow velocity in the $x$ direction. . . . .   | 85  |
| 4.15 | The unstructured grid for the splitter plate nozzle problem. . . . .   | 86  |
| 4.16 | Contour plot of the acoustic pressure at the end of a cycle of oscillation of a source inside the nozzle. . . . .  | 88  |
| 4.17 | Contour plot of the acoustic pressure at the end of a cycle of oscillation of a source in the far field. . . . .   | 89  |
| 5.1  | The structure of the global matrix and the vector of variables in the frequency domain DG method. . . . .  | 94  |
| 5.2  | The strength of the source for the $1d$ wave equation. . . . .   | 97  |
| 5.3  | Comparison of the DG solution with the exact solution for the $1d$ wave equation in the frequency domain. . . . .  | 98  |
| 5.4  | Convergence of the DG solution for the $1d$ wave equation. . . . .   | 98  |
| 5.5  | The instability wave and the acoustic solution from Agarwal <i>et al.</i> [6] . . . . .  | 99  |
| 5.6  | Comparison of the DG solution with the exact solution for the two dimensional frequency domain problem. . . . .  | 100 |

|      |  |     |
|------|--|-----|
| 5.7  | The one dimensional basis functions for $p = 1$ .  | 105 |
| 5.8  | The element lengths in a triangle.   | 109 |
| 5.9  | Comparison of the exact and SUPG solution for the $1d$ wave equation.  | 111 |
| 5.10 | Convergence of the SUPG solution for the $1d$ wave equation.   | 111 |
| 5.11 | The SUPG method in $2d$ LEE. (a) The $177 \times 97$ grid used. (b) A contour plot of the acoustic pressure. | 112 |
| 5.12 | The pressure values at $y = 15$ : The $89 \times 49$ grid.   | 113 |
| 5.13 | the $177 \times 97$ grid.  | 114 |
| 5.14 | The pressure values at $y = 50$ .  | 114 |
| A.1  | The transformation.  | 127 |
| B.1  | Node numbering in a triangular element for $p = 3$ .   | 136 |
| B.2  | Node numbering in a tetrahedral element for $p = 2$ .  | 137 |

# List of Tables

|     |  |    |
|-----|--|----|
| 2.1 | The three parts of the computation time (seconds) using the DG method. | 24 |
| 4.1 | The arrangement array for the partition A of figure 4.2 . . . . .      | 70 |
| 4.2 | The arrangement array for the partition B of figure 4.2 . . . . .      | 71 |
| 4.3 | Scaled speed up for the parallel solver. . . . .                       | 82 |

# List of Symbols

- $( )^e$  Value, or approximation of a quantity in  $\Omega^e$ , page 14
- $( )^{ej}$  Values on the side/surface numbered  $j$  of the element  $e$ , page 93
- $\alpha$  The dissipation parameter in the Lax-Friedrich's expression, page 15
- $\chi, \hat{\chi}$  Moment polynomials, in original coordinates and in transformed coordinates, page 17
- $\partial\Omega^e, \Gamma$  Boundary of  $\Omega^e$  and the coordinate(s) along the boundary, page 15
- $\partial\Omega_I$  Element edges lying on partition boundary, page 66
- $\partial\Omega_P$  Element edges not lying on partition boundary, page 66
- $\gamma, \mathcal{R}$  Ratio of specific heat coefficients, and universal gas constant, page 36
- I** A  $4 \times 4$  identity matrix, page 37
- A<sub>r</sub>** Coefficient matrix in the  $r^{th}$  dimension, page 36
- B** Elemental edge flux matrix on the side  $j$ , page 93
- C** Reaction matrix, page 36
- K** Stiffness matrix, page 17
- M** Mass matrix, page 17
- T** Edge transformation matrix, page 29
- $N_{el}$  Number of elements in the computational domain
- $N_{nod}$  Number of nodes in the computational domain
- ⊙ Row multiple, page 133

- $\overline{(\quad)}$  The value of a quantity in the reduced dimension, or the trace of the quantity , page 15
- $\Omega$  A bounded computational domain, page 14
- $\omega$  Frequency of the acoustic solution
- $\Omega^e, \Delta$  Elements which span  $\Omega$ , and their mapping in transformed coordinates,  $\xi$ ,  $\eta$ , and  $\zeta$ , page 14
- $\otimes$  Kronecker product, page 133
- $\phi, \hat{\phi}$  Lagrange polynomials, in original coordinates and in transformed coordinates, page 30
- $\psi, \hat{\psi}$  Basis set polynomials, in original coordinates  $x, y$  and  $z$  and in transformed coordinates, page 14
- $\rho, u, v, p$  Acoustic density,  $x$  - velocity component,  $y$  - velocity component, and pressure , page 36
- $\rho_o, P_o$  Non dimensional mean flow quantity, page 36
- $\sigma_x, \sigma_y, \sigma$  Damping parameters in the PML, page 39
- $\tilde{(\quad)}$  Spatial component of the acoustic solution
- $\vec{n}^{\partial\Omega^e}$  Unit outward normal to  $\partial\Omega^e$ , page 15
- $\vec{(\quad)}$  A physical vector quantity
- $\vec{F}$  Flux, a function of  $q$ , page 14
- $D$  Diameter of the nozzle at exit, page 84
- $d$  Spatial dimension, page 14
- $h$  Mesh size
- $J, J_\Gamma$  Jacobian of coordinate transformation from  $\Omega^e$  to  $\Delta$ , and the Jacobian of the edge coordinate transformation, page 16
- $M_x, M_y$  Mach number of the mean flow in  $x$  and  $y$  directions, page 36
- $N$  Size of the basis set in an element, page 14
- $N_{nz}$  Number of non-zeros in the global matrix, page 101

$p$  Degree of the basis set polynomials, page 14

$q$  Dependent variable(s), page 14

$t$  time, page 14

$T_s$  Non dimensional time period of oscillation, page 87

$x_1, x_2$  Coordinates,  $x_1 \equiv x$ ,  $x_2 \equiv y$ , page 36

$z$  PML variable, page 41

$s$  Acoustic source term, page 36

Tensor Notation: In the thesis, vectors and matrices are represented by bold type face. In places where additional clarity is necessary, vectors and matrices are enclosed by  $\{ \}$  and  $[ ]$  respectively. The elements of vectors and matrices are represented in the subscript, or the superscript, and are enclosed in round braces, as  $( )^{(kj)}$ , or  $( )_{(j)}$ .

# Acknowledgments

I am extremely grateful to my advisor Dr. Philip J. Morris, who has guided and helped me at every instant of my studies at Penn State. His encouragement, support and patience have nurtured my intellectual and personal character growth. I am honored to have known, worked and interacted with him. I thank him very much for all the help and consideration.

I would like to thank my committee members, Dr. Lyle N. Long, Dr. Kenneth S. Brentner and Dr. Paul E. Plassmann for reviewing my thesis, and providing valuable suggestions in the technical discussions that we had. I am grateful to my friend Anurag Agarwal for all the help, and also for providing me the results from his calculations on the benchmark problems for comparison. I would also like to thank my friend Steve Zygmunt for giving me the mean flow solution to the splitter plate nozzle configuration. I thank my colleagues Yih-Pin Liew, Umesh Paliath, Vernecia S. McKay, Debbie Sas-trapradja, Said Boluriaan and Anupam Sharma for making my stay very pleasant and memorable.

Finally I would like to express my gratitude to my family. It is beyond my ability to put into words the love, support and guidance that my parents have given me in my life. I am very grateful to my wife Sowmya for her unfailing love, support and understanding during my studies.

*Dedicated to my parents  
Meenakshi and Prahallad Rao*

# Chapter 1

## Introduction

This thesis describes the development, implementation and optimization of high order accurate algorithms on unstructured grids. Methods are developed for both time and frequency domain solutions. The focus of the application is on problems in aeroacoustics: specifically, the propagation of sound in sheared mean flows and in the presence of complex geometry boundaries. Both continuous and discontinuous Galerkin methods are used. The primary contributions of this thesis are: the optimization of time domain discontinuous Galerkin methods, the implementation and evaluation of discontinuous Galerkin methods in the frequency domain, and the use of stabilized, continuous Galerkin methods in the frequency domain.

### 1.1 Computational Aeroacoustics

Aerodynamic noise is a major concern in the design of modern aircraft. The control and minimization of noise are significant engineering design issues in today's environmentally conscious society. Some of the primary generators of noise encountered in aeronautic applications are jet engines, propellers, airframe components such as high

lift systems, and helicopter rotors. A key goal of the science of aeroacoustics is the development of effective modeling techniques for the analysis and control of radiated noise. Experimental studies in aeroacoustics are very expensive. They require anechoic tunnels and delicate instruments able to measure the high-frequency, low amplitude pressure fluctuations. The exponential growth of computing power in today's world has increased the emphasis on computational aeroacoustics (CAA), the branch of computational fluid dynamics that deals with the numerical simulation of generation and propagation of sound waves through a moving medium. CAA is a powerful tool to simulate the aerodynamic noise from complex turbulent flow fields.

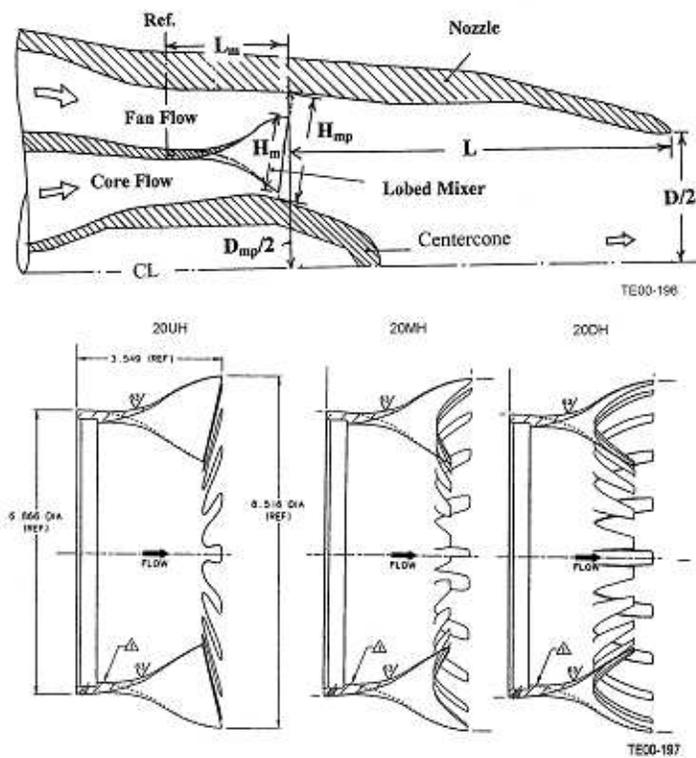
Aerodynamic noise in a turbulent flow originates from the unsteady motion of turbulent structures in the flow, usually in a shear layer or near a solid surface. The sound generated then propagates to the far field through the non-uniformly moving air. The pressure perturbations caused by the sound are very small compared to the value of the mean flow pressure. Now, computational fluid dynamics (CFD) codes based on some form of turbulence modeling can be used to characterize the sources of noise in the flow field. Since acoustic perturbations are small, linear methods could be used to simulate the propagation of sound to far field observer locations. Such methods include Ffowcs Williams-Hawkings (FW-H) method, the Kirchoff method [1] and methods using the Linearized Euler Equations (LEE). The FW-H method involves a rearrangement of the Navier-Stokes equations for the calculation of sound propagation at a particular far field location, given the perturbations on a surface. The sources outside the surface are usually neglected. The Kirchoff method assumes the wave equation as a model for the propagation of the sound waves outside the surface on which the perturbation values are given. The LEE, on the other hand, do not involve the specification of a volume or a surface which bounds the sources. While the FW-H method and the Kirchoff method are suited for the calculation of noise at a particular observer location, the LEE are suited

for a obtaining a contour map of the noise surrounding the acoustic sources. The noise map can be of importance in certain situations, such as the community noise levels near an airport. In this work, the LEE are used for the simulation of acoustic propagation, with a particular focus on the jet noise.

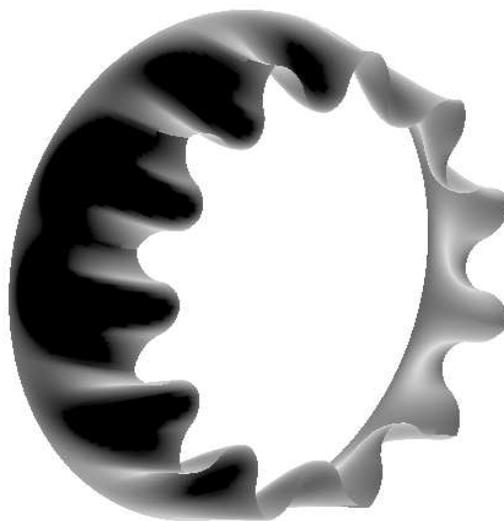
The LEE describe the propagation of sound waves in a mean flow. Euler equations are used to describe the sound propagation since the sound perturbations are only weakly affected by viscosity. The solution methods for the LEE, which try to resolve the perturbations in the flow pressure, have to be orders of magnitude more accurate compared to the solution schemes used for traditional CFD. The primary requirements of numerical methods for CAA can be summarized as:

1. High order of accuracy.
2. Low dispersion and dissipation errors.
3. Ability to propagate the sound perturbations to large spatial and time scales without a prohibitive computational cost.
4. Applicability to complex geometries.

Devising robust, accurate, and efficient numerical methods for the solution of problems in CAA is a challenge. Many traditional approaches based on finite difference methods (FDM) are relatively easy and straightforward to implement. However, they suffer from a major drawback, that is, their inability to handle complex geometries. As can be seen in Fig. 1.2, a typical configuration of a forced mixer of an aircraft jet engine is so complex that it is best represented using an unstructured grid. The use of FDM in such complex geometry is very difficult. Finite volume methods, which are very popular in CFD, are an alternative. However, finite volume methods do not provide high orders of accuracy in the solution and so require a very fine grid.



**Figure 1.1.** The configuration of an internal mixer in a jet engine. (courtesy Rolls Royce Allison), from Mengle *et al.* [2].



**Figure 1.2.** A conceptual diagram of the lobed internal mixer.

The finite element method (FEM) is another natural alternative for unstructured grids. Conventional finite element methods involve an assembly of elements, and subsequent solution of a global matrix. As the order of the approximation polynomials is increased, the size of the global matrix increases rapidly. One of the requirements of CAA is the accurate simulation of sound propagation over large time and spatial scales. This would involve a solution of the global matrix at each time step. The procedure would consume a prohibitive amount of time since a global matrix has to be inverted at each time step, for millions of time steps in engineering applications. However, as described in this thesis, there are situations in which a finite element approach can be valuable.

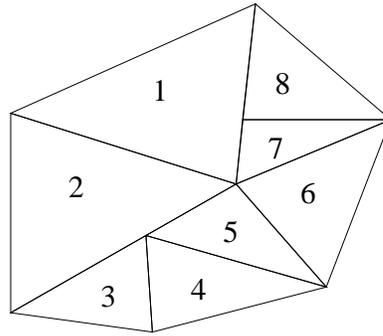
A class of finite element methods, known as the discontinuous Galerkin finite element method, (DG method), has been developed specifically for the solution of convection dominated flows. The DG method has many properties that are desirable not only in CAA, but in many other engineering fields as diverse as gas dynamics, meteorology, oceanography, turbulent flows, viscoelastic flows, magneto hydrodynamics and electromagnetism [3]. These properties are briefly described in the following section. Most of the work in this thesis concentrates on the development of an optimal DG method solver for aeroacoustic applications in aerospace engineering. The interest in the DG method has been growing in the aeroacoustic community in recent years. A brief review of the existing literature on the DG method applied to aeroacoustics is given in section 2.1.

## **1.2 The DG method**

The DG method was first developed in 1973, and enhancements and further development of the method have taken place in the last decade. The method has been shown to be well suited for high order accurate long time scale simulations. A key distinction

between the DG method and the conventional Galerkin FEM is that the resulting formulation is local to an element. Each element in the DG method can be thought of as an independent entity, merely requiring the boundary data, that is, flux, from the surrounding elements. Since the DG method incorporates numerical fluxes and discontinuous elements, it can be considered as a generalization of finite volume methods. The DG method has been classified also as one among the spectral  $h - p$  finite element methods, where convergence is achieved by a combination of both grid refinement ( $h$ -type) and polynomial order increase ( $p$ -type). Owing to its unique finite element nature, the DG method has numerous advantages over classical finite volume and finite element methods [3, 4]. These include:

- The method can be used to obtain uniformly high order accurate solutions. The rate of increase in the computational cost with the order of accuracy is considerably less compared to other numerical methods.
- It is well suited for complex geometries since it can be applied on unstructured grids. In addition the method can also handle non-conforming elements, as shown in Fig. 1.3, where the elements are seen to have “hanging” nodes. In the figure, elements (1,8,7), and (2,3,5) are non-conforming. The rest of the element pairs are conforming.
- The method is highly parallelizable. The method is compact, since each element is independent. Since the elements are discontinuous, and the inter-element communications are minimal, domain decomposition can be efficiently employed. The compactness also allows for a structured and simplified coding for the method.
- Implementation of various boundary conditions is straight forward and more natural in the DG method since the flux needs to be specified at the boundaries. This



**Figure 1.3.** Conforming and non-conforming elements.

is especially useful in the area of CFD where the boundary conditions are usually specified in terms of a flux.

- It can easily handle adaptive strategies, since refining or coarsening the grid can be achieved without considering the continuity restrictions typical to the conforming elements. The method allows for heterogeneity in the elements - for example, the order of accuracy, or shape, can vary from element to element.
- It has several useful mathematical properties with regards to stability and convergence. A brief discussion of some of the studies on the numerical properties of the method is given in Chapter 2.

Although the DG method is less susceptible to the problems that are common in finite difference schemes, it is not without its own weaknesses. The method has been recognized as expensive, in terms of both computational operation count and storage requirements. Although theoretically the method can be applied to an element of any shape, the requirement of numerical quadrature for the integrals in the formulation has restricted the applications to hexahedral and quadrilateral elements. The recently developed quadrature-free DG method [5] tries to resolve these problems by using simple polynomial basis functions, the products of which can be integrated without numerical

quadrature.

### 1.3 Calculations in the frequency domain

The linearized Euler equations, which are the standard equations in aeroacoustics, support spatially growing solutions. However, these instabilities, which are known as the Kelvin-Helmholtz instabilities, are limited in a real flow by the nonlinear effects present in the full governing equations, Navier-Stokes equations. Often, the acoustic solution is obscured by the non-physical instability solution. One approach to filter out the instabilities is to assume a time harmonic solution and solve the LEE in the frequency domain [6]. This transportation to the frequency domain modifies the originally hyperbolic LEE to an elliptic system of equations. The resulting algebraic system of equations has to be solved using a non-iterative method to eliminate the instability solution. The solution process for a hyperbolic problem involves marching in time, and at each time step, the domain is swept element after element - separate linear systems are solved for each element. But, for an elliptic problem, the solution is globally dependent, and hence an assembly of elements is necessary. In the DG method, since each element has a separate set of variables, the size of the resulting global matrix is large compared to the size of an equivalent matrix obtained using a continuous Galerkin FEM or FDM. It is shown in Chapter 5 that the DG method requires a very large usage of memory compared to a continuous Galerkin method when an assembly of elements is performed. The memory requirement is critical, especially for the case of the LEE in the frequency domain, where the problem involves multiple variables. The reduction of the system of equations to a single equation in pressure is analytically impossible in practical problems with spatially non-uniform mean flow quantities.

In the present work, as an alternative to the memory-expensive DG method for fre-

quency domain problems, the streamline upwind Petrov Galerkin (SUPG) method is considered. It is well known that the conventional Galerkin methods are unstable for convection-dominated problems [7]. In the SUPG method, stabilization is achieved by the addition of an artificial dissipation function to the trial space functions of the Galerkin formulation.

## 1.4 Motivation

The work presented in this thesis describes development of efficient tools for aeroacoustic applications in aerospace engineering, in particular, jet engine noise. The solution to the LEE is sought on unstructured grids, as the typical engine components have complex geometry. The first goal is to improve the efficiency of the DG method for applications in the time domain. The efficiency of the DG method is increased in two ways. First, new approaches are implemented in the algorithm, by making use of a nodal polynomial basis with group flux interpolation in a quadrature-free framework. Second, an implementation of the DG method on parallel computing platforms using domain decomposition is carried out. Another goal of the work is to develop and demonstrate a solver for the LEE in the frequency domain, to suppress the Kelvin-Helmholtz instabilities. Again, the solution is sought on an unstructured mesh. Two versions of FEM, namely, the DG method and the SUPG method are considered. Although all the developments in the thesis are carried out in two spatial dimensions, a discussion of the extension of the algorithm to three dimensions is also provided. It is expected that in three dimensions also, the approaches developed in this thesis improve the efficiency of the solvers.

## 1.5 Organization of the thesis

This thesis is organized as follows. In Chapter 2, a general formulation of the DG method for a hyperbolic conservation law is presented, followed by a detailed discussion of the choice of polynomials for the basis set expansion. An efficient DG formulation is developed in a quadrature-free environment using a general polynomial basis set, and is applied to one and two dimensional advection equations. In Chapter 3, the DG formulation is applied to the LEE in the time domain. The implementation of boundary conditions in two dimensions is discussed. The DG solver is validated using benchmark problems involving uniform and non-uniform mean flows. A parallel implementation of the DG algorithm is presented in Chapter 4. The parallel time domain DG solver is then used to predict the propagation of sound waves in the shear layer of a realistic jet. In Chapter 5, an evaluation of both the DG method and the SUPG method for solutions to the LEE in the frequency domain is conducted. The suppression of the instability wave solution for a benchmark problem is demonstrated using both methods. All the algorithms are coded using the programming language Fortran 90, except the preprocessor for the parallel solver, which is developed using C++. Finally, in Chapter 6, conclusions, a summary of the results and suggestions for future work are given.

# Chapter 2

## The generalized DG method

In this chapter, the development of the DG method for a hyperbolic conservation law is presented using a general polynomial basis set in a quadrature-free set up. Various methods to improve the efficiency of a DG solver are presented next, with particular emphasis on aeroacoustic applications involving non-uniform mean flows. It is shown that the choice of the basis set and the method of evaluation of the element boundary matrices are significant factors in the optimization of the algorithm. Results from the application of the DG algorithm are presented for one and two dimensional advection equations.

### 2.1 Existing literature on the DG method

Many researchers have worked on the theoretical and computational development, the mathematical properties and applications of the DG method since its introduction. The quantity of existing literature pertaining to the DG method is substantial. Here, only a brief note about some of the important works in its development and application is presented. The reviews by Cockburn, Karniadakis and Shu [3] provide an exhaustive

literature survey, and Cockburn [8, 9] give the general development and analysis of the DG method.

The DG method was originally introduced in 1973 by Reed and Hill [10] for the neutron transport equation. In 1974, LeSaint and Raviart [11] performed the first analysis of the DG method and proved a rate of convergence of  $(\Delta x)^{p+1}$  in the  $L^2$  norm for tensor products of polynomials of order  $p$  defined on Cartesian grids. Later, Johnson and Pitkäranta [12] proved a rate of convergence of  $(\Delta x)^{p+1/2}$  for a general triangulation. A wave propagation analysis of the DG method was performed by Rasetarinera *et al.* [13]. An eigensolution analysis of the DG method on non-uniform grids in one and two dimensions was carried out by Hu and Atkins [14], [15]. They calculated numerical dispersion and dissipation errors for various orders of the method and directions of wave propagation. A dispersion analysis of spectral element methods such as the DG method for various basis functions such as Legendre and Chebyshev polynomials was conducted by Stanescu *et al.* [16]. Although implicit time discretizations were initially studied by many researchers, explicit time discretization became popular due to the development of Total Variation Diminishing Runge Kutta (TVDRK) schemes by Cockburn and Shu [17] for hyperbolic equations using generalized slope delimiters.

The choice of the basis set and the expansions of the nonlinear flux terms in the DG method have been areas of study in recent years. Sherwin and Karniadakis [18] discuss general choices for the basis set polynomials for spectral element methods such as the DG method. Atkins and Shu [5] developed the quadrature-free method in 1996 and introduced a novel way to reduce the storage and computational time using a basis set consisting of simple polynomials from Taylor series expansions of the variables about the centroids of the elements. They also proved that the DG method is insensitive to mesh discontinuities, that is, sudden changes in the mesh size. Lockard and Atkins [19] conducted a study on ways to implement the quadrature-free DG method

of [5] efficiently. Use of the sparsity of matrices resulting from a rearrangement of basis set polynomials and truncation of the flux expansions were suggested as primary tools for reducing the computational cost.

The DG method has been applied for the solution of the governing equations in convection and diffusion dominated phenomena such as gas flows, electromagnetics, viscous magneto hydrodynamics, shallow water transport, and aeroacoustics [3]. Only some of the studies that have dealt with the application of the DG method in time-domain aeroacoustics are noted below. Recently, Stanescu *et al.* [20] applied a collocation form of the DG method to the aircraft acoustic scattering problem. Remaki *et al.* [21] used a three dimensional DG method for multiple pure tone noise problems in the presence of shocks and rarefactions. Atkins [22, 23] solved several benchmark problems using the quadrature free version of the DG method. Lockard and Atkins [24] applied the DG method to an aeroacoustic analysis of realistic aircraft configurations, such as a two dimensional slat and a three dimensional blended wing-body configuration. It can be seen that the DG method has gained interest among researchers owing to its high order of accuracy and compact nature. Although many researchers have used the DG method to solve practical problems in aeroacoustics, the development of an optimized DG solver is still an issue that is being explored extensively.

## **2.2 Formulation of the DG method**

The Runge-Kutta DG method, which was developed initially by Cockburn and Shu [17] can be applied to discretize a system of first order conservation equations, such as the linearized Euler equations in aeroacoustics. Here, for the sake of simplicity, the governing equation is chosen to be the conservation equation of a scalar variable in a bounded

domain  $\Omega \in \mathbb{R}^d$ :

$$\frac{\partial q}{\partial t} + \nabla \cdot \vec{F} = s, \quad (2.1)$$

where  $q$  is the conserved variable,  $\vec{F}$  is the flux, which is a function of  $q$ , and  $s$  is a source term independent of  $q$ . Let the domain be divided into a number of non-overlapping elements  $\Omega^e$  that span the entire  $\Omega$ . In the work of this thesis, the spatial dimension  $d \leq 2$  and  $\Omega^e$  is assumed to be a straight line in one dimension and an unstructured triangle with straight edges in two dimension, and the coordinate system is chosen to be Cartesian, unless stated otherwise. Let  $q^e$  represent an approximation to the variable  $q$  in  $\Omega^e$  using functions of a basis set as:

$$q^e = \sum_{j=1}^N q_j^e \psi_j \quad (2.2)$$

where  $\psi_k$ ,  $1 \leq k \leq N$  are polynomials of degree  $p$  in a basis set given by

$$\mathcal{B} \Rightarrow \{\psi_k\} \in \left\{ \mathbb{P}^p(x, y) \equiv (x^m y^n) : p \leq m + n \right\}. \quad (2.3)$$

The size of the basis set,  $N = N^e(p, d)$  depends on the degree of the polynomials of expansion, and the number of spatial dimensions, as:

$$N = \frac{(p+1)(p+2) \dots (p+d)}{d!} \quad \text{for } d = 1, 2, 3, \dots \quad (2.4)$$

Applying the traditional Galerkin method to equation (2.1) using (2.2) and (2.3) yields, for  $k = 1 \dots N$ ,

$$\int_{\Omega^e} \psi_k \left( \frac{\partial q^e}{\partial t} + \nabla \cdot \vec{F}^e - s^e \right) d\Omega^e = 0, \quad (2.5)$$

Using integration by parts, equation (2.5) can be written as,

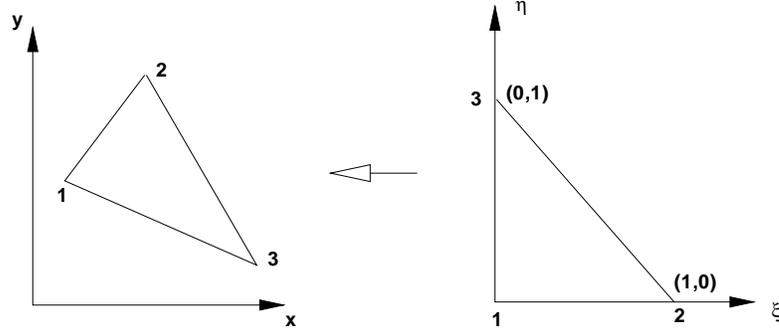
$$\int_{\Omega^e} \psi_k \sum_{j=1}^N \psi_j \frac{\partial q_j^e}{\partial t} d\Omega^e - \int_{\Omega^e} \nabla \psi_k \cdot \vec{F}^e d\Omega^e + \int_{\partial\Omega^e} \psi_k \bar{F}_R^e d\Gamma = \int_{\Omega^e} \psi_k s^e d\Omega^e \quad (2.6)$$

for  $k = 1 \dots N$ . Here  $d\Gamma$  is the surface area element (or the line element in 2D), and  $\bar{F}_R$  is the Riemann flux through the surface or the line element. The Riemann flux is approximated using the traces of the flux values of the element and the neighboring element on the common surface or edge. In the present work, the Riemann flux is approximated using the Lax-Friedrich's form:

$$\bar{F}_R^e = \frac{1}{2} \left\{ \left( \vec{F}_+^e + \vec{F}_-^e \right) \cdot \vec{n}^{\partial\Omega^e} - \alpha \left( \bar{q}_+^e - \bar{q}_-^e \right) \right\} \quad (2.7)$$

where  $\vec{F}_-^e$  is the trace of the flux of the element to the left of the edge, and  $\vec{F}_+^e$  is that of the element to the right of the edge.  $\vec{n}^{\partial\Omega^e}$  is the unit normal to the edge from left to right, and  $\alpha$  is a smooth positive parameter, chosen for the upwind bias. The value of  $\alpha$  is chosen to be close to the convection speed, given by  $\frac{dF}{du}$ . The choice for the value of  $\alpha$  is discussed further in section 3.3.1. It is convenient to carry out the integrations in the formulation (2.6) in transformed coordinates. The integrals are calculated on a mapped element, denoted by  $\Delta$ , but the final equation is evaluated and assembled in the real coordinates of  $\Omega$ . Let the coordinates in the transformed space be  $\xi$ ,  $\eta$  and  $\zeta$ , as shown in Fig. 2.1, with  $(0 < \xi, \eta, \xi + \eta < 1)$ . Then, for  $k = 1 \dots N$ ,

$$\int_{\Delta} \psi_k \sum_{j=1}^N \psi_j \left( \frac{\partial q_j^e}{\partial t} \right) |J| d\Delta - \int_{\Delta} J^{-1} \nabla \psi_k \cdot \vec{F}^e |J| d\Delta + \int_{\partial\Delta} \psi_k \bar{F}_R^e |J_{\Gamma}| d\Gamma^{\Delta} = \int_{\Delta} \psi_k s^e |J| d\Delta \quad (2.8)$$



**Figure 2.1.** The mapping from the real coordinates to the transformation coordinates.

$J$  is the Jacobian of the transformation, given by

$$J(\Delta \rightarrow \Omega^e) \equiv \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \quad (2.9)$$

and  $J_\Gamma$  is the Jacobian of the line coordinate transformation. In general, the flux can be a nonlinear function of  $q$ . Let us approximate the flux using the same basis set, as

$$\vec{F}(q) \approx \vec{F}^e = \sum_{j=1}^M \psi_j \vec{f}_j^e, \quad \text{for } M \geq N. \quad (2.10)$$

Let the source term also be approximated in fashion similar to  $q^e$  in equation (2.2). Then equation (2.8) is written as,

$$\mathbf{M}^e \frac{\partial \mathbf{q}^e}{\partial t} - \mathbf{K}^e \mathbf{f}^e + \int_{\partial \Delta} \psi_k \vec{F}_R^e |J_\Gamma| d\Gamma^\Delta = \mathbf{M}^e \mathbf{s}^e. \quad (2.11)$$

The notation  $\mathbf{q}^e = \{q_j^e\}$ ,  $j = 1 \dots N$  has been used, and the expression for the terms in  $\overline{F}_R^e$ , to be explained later, are omitted. Here the mass matrix,

$$\mathbf{M}_{kj}^e = \int_{\Omega^e} \psi_k \psi_j d\Omega^e = \int_{\Delta} \psi_k \psi_j |J| d\Delta, \quad \forall k, j = 1 \dots N, \quad (2.12)$$

and the stiffness matrix,

$$\mathbf{K}_{kj}^e = \int_{\Omega^e} \nabla \psi_k \psi_j d\Omega^e = \int_{\Delta} J^{-1} \nabla \psi_k \psi_j |J| d\Delta \quad \forall k = 1 \dots N, \forall j = 1 \dots M. \quad (2.13)$$

Due to the term  $J^{-1}$ , the matrix  $\mathbf{K}_{kj}^e$  is comprised of a sum of several matrices. The mass and stiffness matrices together are denoted as element volume matrices. The evaluation of the surface integral in equation (2.8) is discussed further later in this chapter. The above equation (2.8) for  $\mathbf{q}^e$  is solved using the Total Variation Bounded Runge-Kutta (TVBRK) scheme [17] to march in time, for each  $\Omega^e$  in the domain. It can be seen from the formulation that there is no global assembly of the elements involved in the DG method. The implementation of boundary conditions, which are discussed in detail further in Chapter 3 are applied by modifying the Riemann flux expression on the element edges that coincide with the computational boundary. A detailed formulation of the DG method using the moment polynomials basis set

$$\hat{\chi} = (1, \xi, \eta, \xi^2, \xi\eta, \eta^2 \dots) \quad (2.14)$$

for the linear advection equation is presented in the Appendix A, to provide a basic and more detailed understanding of the transformations and calculations involved.

## 2.3 Application to the advection equation

In this section, the DG method developed in section 2.2 is applied to the simple linear scalar advection equation in one and two dimensions using both Lagrange polynomials and moment polynomials. The nonlinear Burger's equation in one dimension is also solved using the generalized quadrature-free approach. These applications have proved helpful in understanding the basic concepts of the DG method and aspects of its implementation.

### 2.3.1 One dimensional calculations

The test equations are the advection equation,

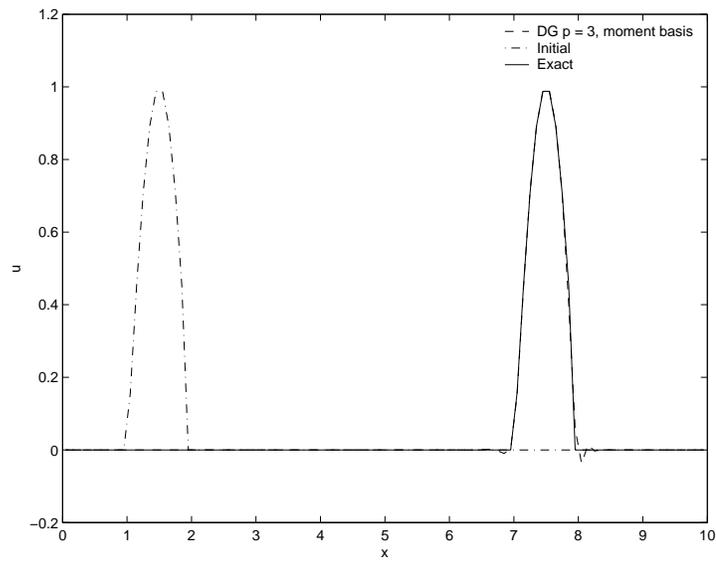
$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0 \quad (2.15)$$

and the nonlinear Burger's equation

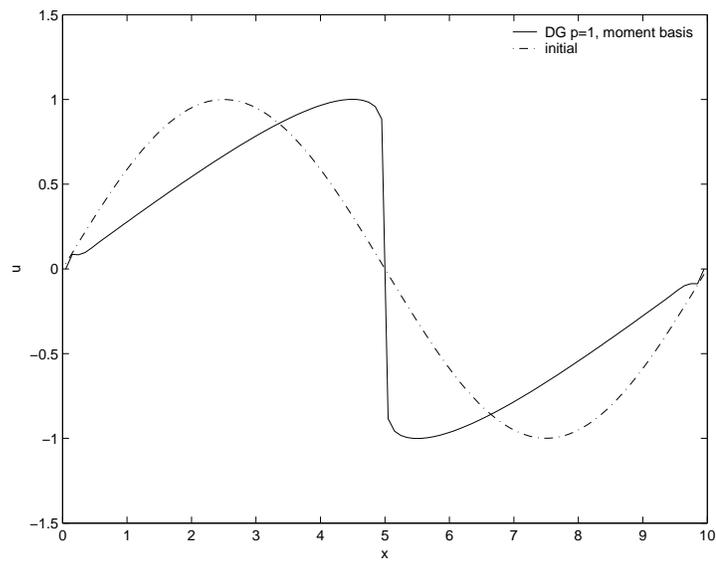
$$\frac{\partial u}{\partial t} + \frac{\partial(u^2/2)}{\partial x} = 0 \quad (2.16)$$

on the domain  $0 \leq x \leq 10$  with periodic boundary conditions.

The solution to the linear advection problem is obtained with an initial condition of a half sine wave over a unit interval. When using the moment polynomials, the initial condition is expanded as a Taylor series about the center of each element. The domain from  $x = (1, 10)$  is divided uniformly into 100 parts, that is, the value of  $\Delta x^e = 0.1$ . The value of the advection speed is  $a = 4.0$ . Fig. 2.2 shows the advected pulse after time  $t = 1.5$  using a moment basis of degree  $p = 3$  and a fourth order Runge Kutta method. The solution to the nonlinear equation is also shown in figure 2.3. The initial condition is a sine wave, which transforms into an 'N' wave after sufficient time.



**Figure 2.2.** One dimensional calculations: The linear advection equation.



**Figure 2.3.** One dimensional calculations : The nonlinear Burger's equation.

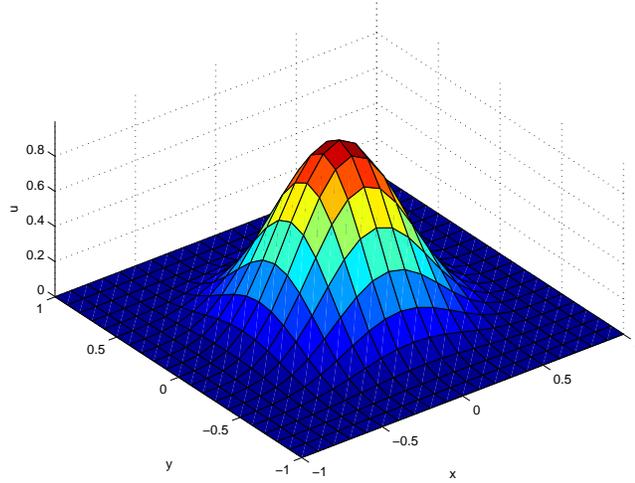
### 2.3.2 Two dimensional calculations

The linear two dimensional advection equation considered here is given by:

$$\frac{\partial u}{\partial t} + a_x \frac{\partial u}{\partial x} + b_y \frac{\partial u}{\partial y} = 0, \quad \text{with} \quad (2.17)$$

$$u(x, y, 0) = u_0 = [\sin(\pi x) \sin(\pi y)]^4,$$

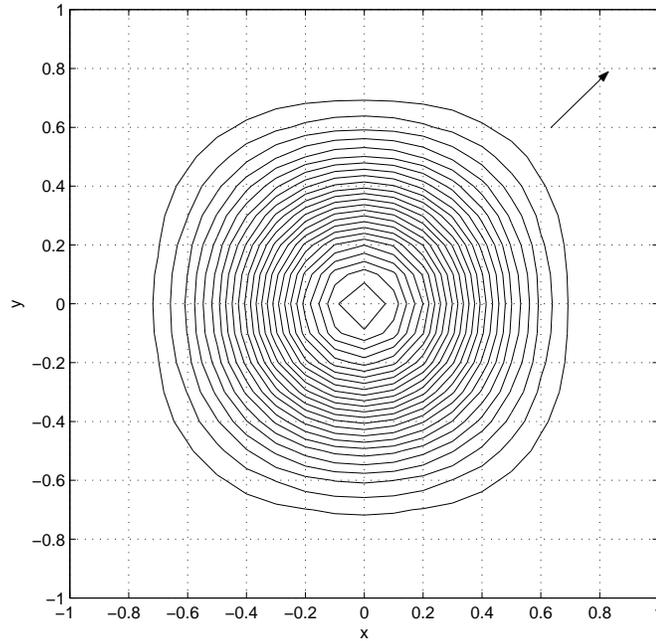
on a periodic square domain  $-1 < x, y < 1$ . The value of the advection speeds are  $a_x = b_y = 1.0$ . The grid is made up of 800 similar right angle triangles. The size of the triangles is constant, with  $\Delta_x^e = \Delta_y^e = 0.1$ . The basis set is made up of Lagrange polynomials of up to second order and moment polynomials of up to fourth degree. The initial condition is shown in Fig. 2.4 and Fig. 2.5.



**Figure 2.4.** The initial condition in two dimensional advection equation: The sine wave pulse with the unit maximum value at the center of the domain.

Figure 2.6 shows the results for the case with the basis set being comprised of cubic moment polynomials ( $p = 3$ ), using the three stage TVBRK method. The advected pulse is shown at various time intervals within a cycle of advection in the periodic domain. A

similar result is obtained with Lagrange polynomials in the basis set. Figure 2.7 and

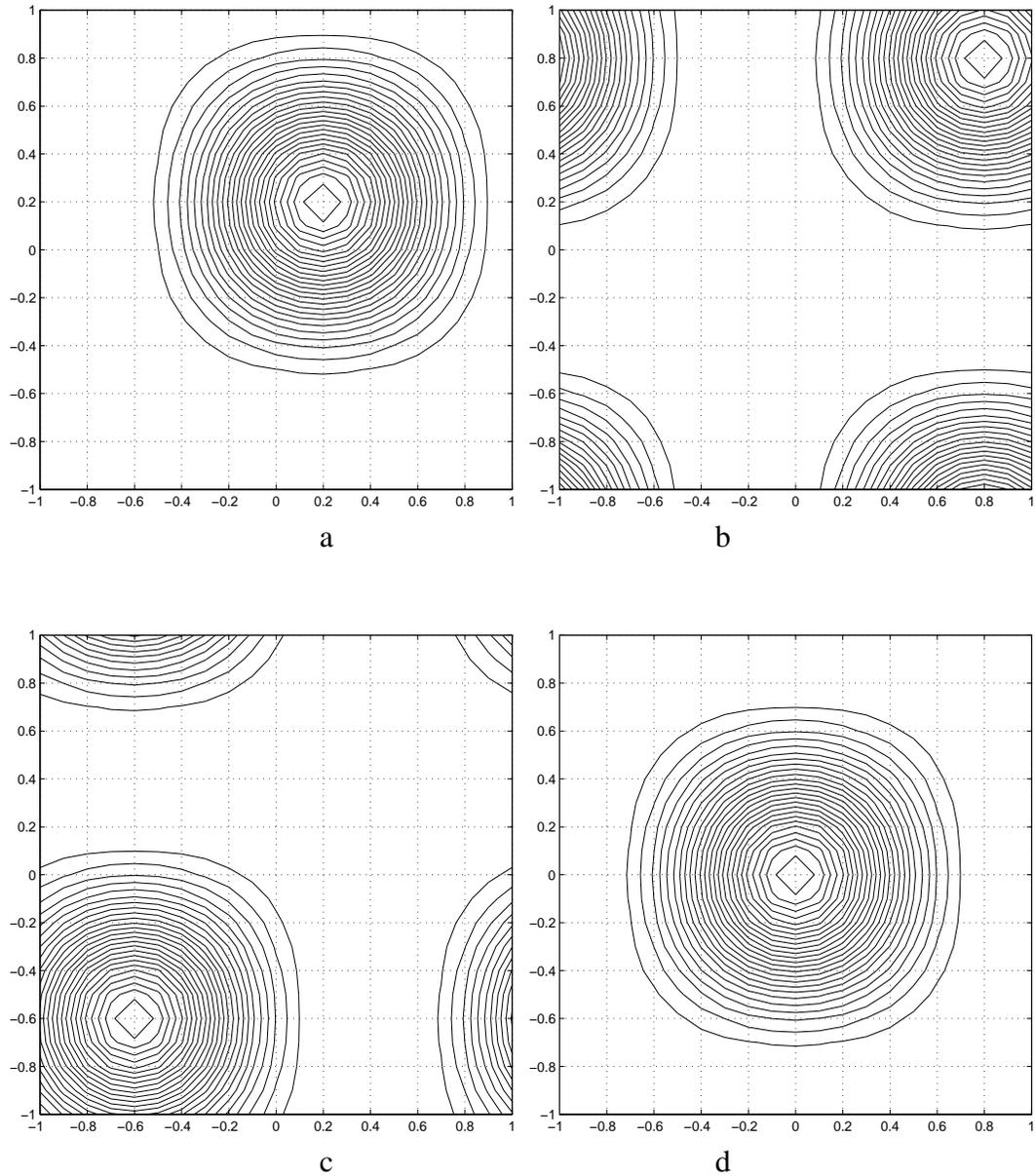


**Figure 2.5.** The initial condition in two dimensional advection equation: A contour map of the initial condition with the direction of advection.

Fig. 2.8 show the error in the solution after one cycle.

The  $L^2$  norm of the global error in the solution after a cycle is found to be

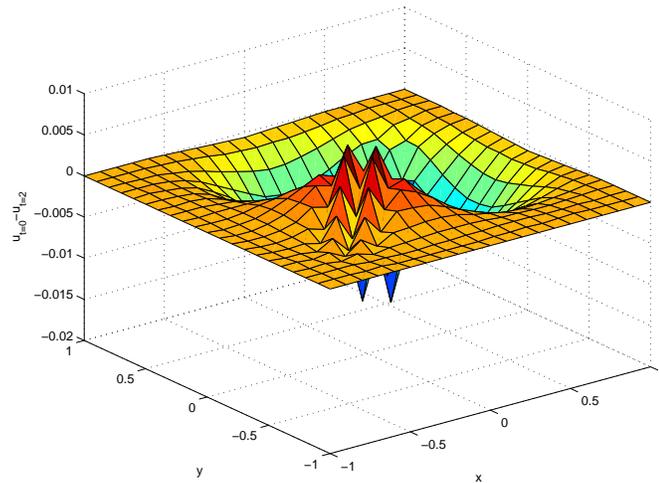
$$\|u - u_o\|_2 = \sqrt{\frac{1}{N_{nod}} \sum_{i=1}^{N_{nod}} (u_i - u_{0i})^2} = 3.7 \times 10^{-3} = O(h^3) \quad (2.18)$$



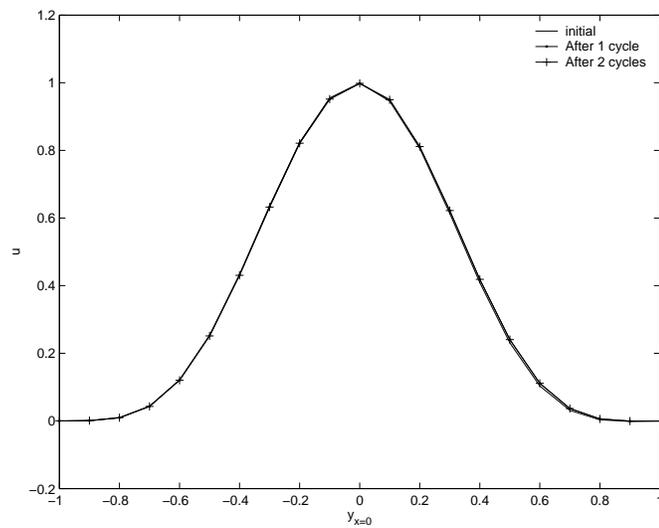
**Figure 2.6.** Advection of the initial pulse. (a)  $t = 0.2$  (b)  $t = 0.8$  (c)  $t = 1.4$  (d)  $t = 2$ , advection through one cycle.

## 2.4 The choice of polynomials for the basis set

The formulation, as given in the previous section, is independent of the choice of polynomials for the basis set and the transformation, as long as the Jacobian of the trans-



**Figure 2.7.** Error in the advection solution after a cycle: Global error  $u - u_0$ .



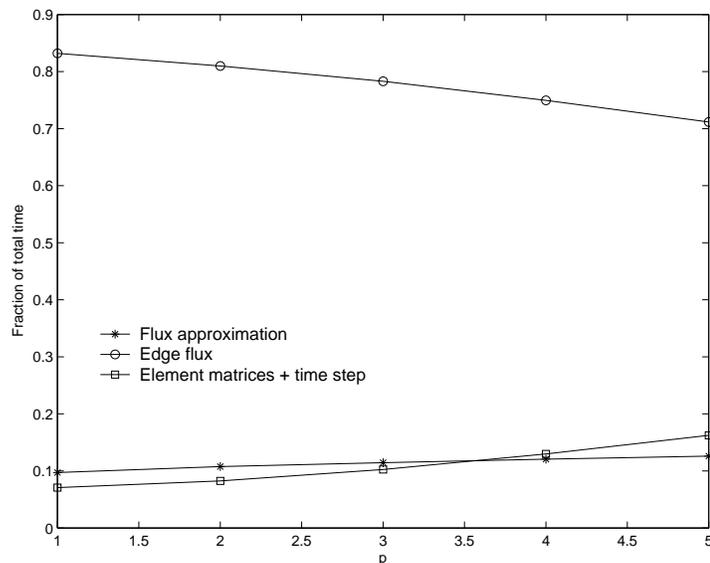
**Figure 2.8.** Error in the advection solution after a cycle: A slice of the solutions at  $y_{x=0}$  at the end of one and two cycles of advection.

formation does not become singular. With a judicious choice of the basis set and the transformation space, the number of numerical operations becomes substantially less, and the algorithm is less cumbersome to program [25]. There are at least three aspects to be considered for the choice of the basis set polynomials in the DG method: The calculation of the element volume matrices, the expansion of flux expressions and the

calculation of edge integrals [26]. The calculation times using the DG method for the solution of the LEE in a uniform mean flow is shown in table 2.1, and Fig. 2.9, for increasing degree of the basis set polynomials. In this case, the basis set polynomials are chosen as the Lagrange interpolants, for reasons which are elaborated further in the following sections. The problem considered is explained in section 3.3.1. As it can be seen, the edge integral calculations take up most of the computation time compared to the other two, for lower orders of approximation. For higher order accurate calculations, the fraction of the calculations involved in element matrices and the flux approximations also increase.

| $p$ | Total time | Flux approximation | Edge flux | Element matrices and time step |
|-----|------------|--------------------|-----------|--------------------------------|
| 1   | 1059       | 103                | 881       | 75                             |
| 2   | 1188       | 128                | 962       | 98                             |
| 3   | 1327       | 152                | 1039      | 136                            |
| 4   | 1557       | 188                | 1167      | 202                            |
| 5   | 1897       | 239                | 1350      | 308                            |

**Table 2.1.** The three parts of the computation time (seconds) using the DG method.



**Figure 2.9.** The three parts of the computation time using the DG method

The three aspects of the DG calculations are discussed in detail in the next sections.

### 2.4.1 Element volume matrices

The element matrices in equations (2.12) and (2.13) are formed by the sum of several matrices that are common to all the elements in the computation domain, multiplied by the components of the Jacobian matrix. In the DG method, there is no assembly of elements required. Hence it is reasonable to calculate and store the element volume matrices prior to marching in time. It is favorable to use a basis set containing orthogonal or nodal polynomials, such as Jacobi, Legendre or Lagrange polynomials. This makes the mass and stiffness matrices sparse and easily invertible. However, with the use of orthogonal or nodal polynomials, there is a need to use quadrature in the evaluation of the integrals.

### 2.4.2 Expansion of flux expressions

Usually, as in the case of Navier-Stokes equations, the flux is nonlinear, or as the in LEE with non-uniform mean flows, the components of the flux expression are spatially varying. Then, the products of variables need to be approximated up to an equal or higher order compared to the variables themselves, with the same set of polynomials as used for the variables. For example, consider a flux component expression in the LEE of the form

$$F_u = \rho_o u \tag{2.19}$$

where  $\rho_o$  is the spatially varying mean flow density, and  $u$  is a component of the acoustic velocity. Let us approximate  $\rho_o$  and  $u$  by,

$$\begin{aligned}\rho_o^e &= \sum_{j=1}^N \Psi_j \rho_{oj}^e \\ u^e &= \sum_{j=1}^N \Psi_j u_j^e.\end{aligned}\tag{2.20}$$

Then the approximate flux takes the form

$$F_u^e = \sum_{j=1}^N \sum_{k=1}^N \Psi_j \rho_{oj}^e \Psi_k u_j^e\tag{2.21}$$

If the above expression is not truncated, the corresponding second term in the DG formulation in equation (2.8) takes the form:

$$\int_{\Omega^e} \nabla \Psi_{ui} (\Psi_j \Psi_k \rho_{oj}^e u_j^e) d\Omega^e, \quad \forall \quad i, j, k = 1 \dots N.\tag{2.22}$$

We have denoted the component of  $\vec{\nabla} \Psi$  in the direction of  $u$  as  $\nabla \Psi_u$ . This integral contains triple products of basis set polynomials, which need to be either evaluated and stored prior to matching in time, or evaluated at each time step. Either way it is computationally expensive, for two reasons. Since  $N$  is large for higher orders, (see equation (2.4)) and there are  $N^3$  terms in the tensor formed by these integrals. Further, the use of orthogonal or nodal polynomials requires the use of quadrature. This was the main reason behind the DG method being recognized as expensive, until the development of the quadrature-free DG method by Atkins and Shu [5].

### The original quadrature-free method.

Atkins and Shu [5] demonstrated the use of the moment basis, equation (2.14) on triangular elements with straight edges in most of the computational domain. The moment basis can be recognized as the terms in the Taylor series expansion of the variables about the centroid of the element. Using the moment basis, the integrals in the formulation are calculated analytically and stored for a certain class of elements before the main computation, using mathematical packages such as Maple or Mathematica. The use of elements with straight edges for most of the domain means that the Jacobian is a constant for each element. Under these conditions, the mass and stiffness matrices are merely additions of matrices that are common for all the elements of a particular type. Using the moment basis, the flux expression in equation (2.21) is further approximated as

$$F_u^e \approx \sum_{j=1}^M \chi_j f_j^e, \quad \text{for } M \geq N. \quad (2.23)$$

This is accomplished relatively easily using a moment basis by either direct truncation, or by projection.

**Truncation:** (for  $d = 1$  and  $p = 2$ )

$$\begin{aligned} F_u^e &= \left( \rho_{o1} + \rho_{o2}\xi + \rho_{o3}\xi^2 \dots \right) \left( u_1 + u_{o2}\xi + u_3\xi^2 \dots \right) \quad (2.24) \\ &= \underbrace{\rho_{o1}u_1}_{f_1^e} + \underbrace{\left( \rho_{o2}u_1 + \rho_{o1}u_2 \right)}_{f_2^e} \xi + \underbrace{\left( \rho_{o3}u_1 + \rho_{o1}u_3 \right)}_{f_3^e} \xi^2 + \dots \text{ neglected terms} \\ &\approx f_1^e + f_2^e \xi + f_3^e \xi^2 \quad (2.25) \end{aligned}$$

This is equivalent to minimizing the truncation error at the centroid of the element.

**Projection:** This involves minimizing the truncation error over the whole element.

$$\int_{\Omega^e} \Psi_k \left( F_u^e - \rho_o^e u^e \right) d\Omega^e = 0, \quad \text{or,}$$

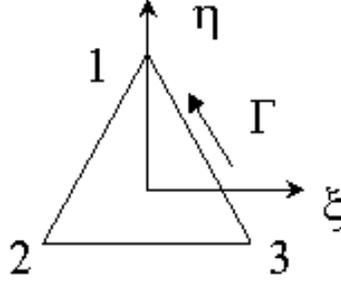
$$\left[ \int_{\Omega^e} \Psi_k \Psi_l d\Omega^e \right] \{f_j^e\} = \left\{ \int_{\Omega^e} \Psi_k \left( \Psi_i \rho_{oi}^e \Psi_j u_j^e \right) d\Omega^e \right\}. \quad (2.26)$$

Despite these simplifications, the use of moment basis has several serious disadvantages:

1. There is a need for additional computation in the formation of the coefficients of equation (2.25) or in the calculation of the right hand side of equation (2.26). These have to be performed at each time step for every element.
2. The coefficients  $\rho_{o1}$ ,  $u_1$ ,  $\rho_{o2}$ ,  $u_2$  in equation (2.24) are the derivatives of the variables  $\rho_o$  and  $u$  at the centroid of the element. These need to be calculated separately. This means an additional step in post-processing the solution, where, usually, the values of the variables at the vertices of the elements are sought.
3. The boundary matrices are dense and the coding for the boundary matrices is cumbersome, as explained below.

### 2.4.3 Element edge integral calculation

In the DG method, the flux needs to be computed on the edge of every element. This involves a transformation of coordinates with a reduction in dimension, as shown in Fig. 2.10. Using a moment basis, it is convenient to transform to an equilateral triangle with the origin at the centroid and the length of the edges equal to unity. Consider the



**Figure 2.10.** Transformation to the edge coordinates.

edge 3 – 1. Using the moment basis with  $p = 1$ , the transformation is given by,

$$\chi = \begin{Bmatrix} 1 \\ \xi \\ \eta \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{4} & -\frac{1}{2} \\ \frac{1}{4\sqrt{3}} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{Bmatrix} 1 \\ \Gamma \end{Bmatrix} = \mathbf{T}_3^e \chi_\Gamma \quad (2.27)$$

Consider a term corresponding to the variable  $\bar{q}_-^e$  of the Riemann flux 2.7, given by equation (2.7) in the edge flux integral, in equation (2.8). Using (2.27), the value of the integral on the side 3 – 1 is

$$\begin{aligned} \int_{3-1} \{\chi\} \bar{q}_-^e |J_\Gamma| d\Gamma &= |J_\Gamma| \int_{3-1} \mathbf{T}_3^e \{\chi_\Gamma\} \{\mathbf{q}^e\}^T \mathbf{T}_3^e \{\chi_\Gamma\} d\Gamma \\ &= |J_\Gamma| \mathbf{T}_3^e \left[ \int_{3-1} \{\chi_\Gamma\} \{\chi_\Gamma\}^T d\Gamma \right] [\mathbf{T}_3^e]^T \mathbf{q}^e \end{aligned} \quad (2.28)$$

Thus the numerical edge integral consists of a matrix vector operation, with the matrices  $\mathbf{T}^e$  being dense for higher  $p$ , and different for different edges.

If the polynomials chosen reduce to zero, or to simple polynomials along the edges of the computational domain, then the operation count for the boundary integral calculations are reduced significantly. On an unstructured grid made of triangles, orthogonal and nodal polynomials possess this property. Considering Lagrange polynomials, the

transformation on any edge of the triangle is given by their equivalents of the same degree along the edge. This is shown for  $p = 2$  for an arbitrary triangle in Fig. 2.11a and Fig. 2.11b.

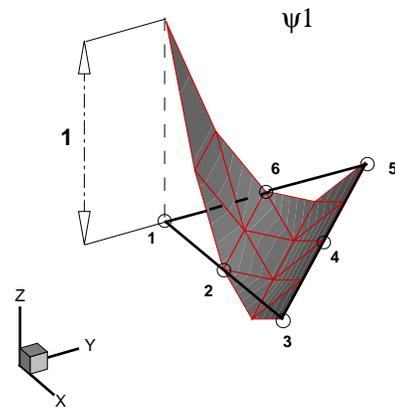
For  $p = 1$ , the transformed polynomials on side 3 – 1 in Fig. 2.10 are  $[1 - \Gamma, 0, 1 + \Gamma]^T$ , and the permutations of this vector on the other sides. The edge integral on side 3 – 1 is,

$$\int_{3-1} \{\phi\} \overline{q^e} |J_\Gamma| d\Gamma = \frac{1}{6} \begin{Bmatrix} 2q_1^e + q_3^e \\ 0 \\ q_1^e + 2q_3^e \end{Bmatrix} = \frac{1}{6} \begin{bmatrix} 2 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 2 \end{bmatrix} \mathbf{q}^e \quad (2.29)$$

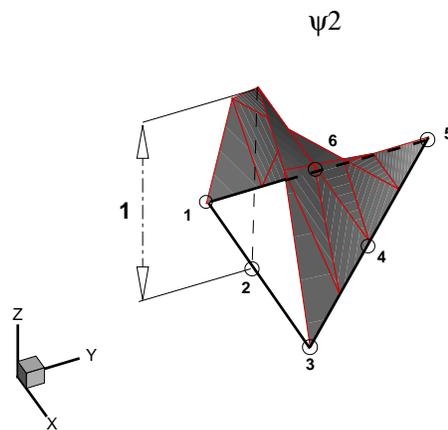
Note that the second row and column corresponding to the second node is zero in the above matrix. The rest of the elements are simply the rearrangements of the elements of the mass matrix corresponding to  $d = 1$ . This means that the boundary integrals can be computed by simple dot products of the rows of the mass matrix in one (or two) dimensions and the nodal variables on that boundary. So, even the sparsity in the matrix above, which is substantially high for higher  $p$  and  $d$  need not be separately included in the algorithm, saving complex routines or calls to sparse matrix packages. It is extremely important to cut down on the operation count in the edge flux calculations, since they constitute a major part of the total computation, as noted by Lockard and Atkins [19]. Hence the use of nodal or orthogonal polynomials is advantageous.

## 2.5 A general polynomial basis set in quadrature-free method

From the discussions in the previous sections, it is evident that it is important to use a nodal or orthogonal polynomial basis, without using quadrature for evaluation of the



(a)



(b)

**Figure 2.11.** Lagrange polynomials,  $p = 2$ .

integrals. To use any general polynomial basis set of order  $p$ , without using quadrature, a simple transformation from  $\chi^p$  to  $\psi^p$  is sufficient:

$$\begin{aligned}\psi^p &= \mathbf{Z}^p \chi^p \\ \frac{\partial \psi^p}{\partial \xi} &= \mathbf{Z}^p \frac{\partial \chi^p}{\partial \xi} \dots\end{aligned}\quad (2.30)$$

$\mathbf{Z}^p$  is a matrix containing only numerical constants, depending on the basis set  $\psi^p$ . This makes the evaluation of integrals of the mass and stiffness matrices extremely simple, for any basis set. As an example, in the transformed space, the Lagrange interpolants for  $p = 1$  are related to the moment basis as,

$$\psi^1 = \begin{Bmatrix} 1 - \xi - \eta \\ \xi \\ \eta \end{Bmatrix} = \mathbf{Z}^1 \chi^1 = \begin{bmatrix} 1 & -1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} 1 \\ \xi \\ \eta \end{Bmatrix}\quad (2.31)$$

For  $p > 1$ , the numerical values of elements in each row of  $\mathbf{Z}$  are evaluated by using the values of  $\psi$  corresponding to that row at the nodes of interpolation. That is, for the  $i^{th}$  row of  $\mathbf{Z}$ , solve

$$\sum_{j=1}^N Z_{ij} \chi_j(\xi_k, \eta_k) = \psi_i(\xi_k, \eta_k), \quad \forall k = 1 \dots N. \quad (2.32)$$

Using  $\psi^p$  and  $\chi^p$ , the equations (2.12) and (2.13) for  $N = M$  are written as,

$$\mathbf{M}^e = \int_{\Delta} \psi^p \psi^{pT} d\Delta = \mathbf{Z}^p \left( \int_{\Delta} [\chi^p \chi^{pT}] d\Delta \right) \mathbf{Z}^{pT} \quad (2.33)$$

and,

$$\mathbf{K}^e = \int_{\Delta} J^{-1} \nabla \psi^p \psi^{pT} |J| d\Delta = \mathbf{Z}^p \left( \int_{\Delta} [\nabla \chi^p \chi^{pT}] |J| d\Delta \right) \mathbf{Z}^{pT} \quad (2.34)$$

Using the definition of  $\chi^p$  in two dimensions, the terms in the integrals in equations (2.33) and (2.34) are evaluated as:

$$\int \xi^k \eta^l d\Delta = \frac{k!l!}{(k+l+2)!} \quad \text{for } k, l \geq 0 \quad (2.35)$$

over the right angle triangle element in the transformation space.

### 2.5.1 Use of Lagrange polynomials for flux interpolation

In this work, Lagrange polynomials are used in the basis set. Lagrange polynomials have been extensively used in FEM, because they are nodal basis functions. Nodal basis functions interpolate the actual values of the flow variables over the nodes of an element, thus allowing for global assembly of elements. Under the DG framework, Lagrange polynomials have been rarely used, not only because of the absence of global assembly in the DG method, but also due to the difficulty encountered in integrating the products of Lagrange polynomials arising out of the nonlinear flux expansion.

However, since Lagrange polynomials interpolate the values of the variables, they can as well be used to interpolate the products of variables. The flux in equation (2.21) is approximated as,

$$F_u^e \approx \sum_{j=1}^N \phi_j f_j = \sum_{j=1}^N \phi_j (\rho_{oj} u_j) \quad (2.36)$$

This makes the evaluation of integrals of the mass and stiffness matrix extremely simple. The integrals are precomputed and stored without much storage requirement.

**Remark 1.** Lagrange polynomials used in this fashion can be seen as an automatic truncation of the flux expansion. Although this implies a decrease in the accuracy, it can be overcome by using higher order elements. It has to be also noted that truncation of terms is indeed necessary to reduce the time of computation. For

example, Lockard and Atkins [19] report a 75% reduction in computational time when the product of terms in the flux expressions are approximated by truncation, while using the moment polynomials in the basis set.

**Remark 2.** Although the idea of using Lagrange polynomials in this fashion was proposed independently in this study, it was later found that this had already been employed by Fletcher [27] in his studies, and referred to as “Group interpolation”.

Having selected the Lagrange polynomials as the basis set functions, the development of the DG method for the LEE in the time domain is considered next. In the next chapter, the application of a serial DG algorithm to solve several benchmark problems in aeroacoustics is discussed.

# Chapter 3

## Time domain calculations

In this chapter, the generalized DG method from Chapter 2 is applied to the linearized Euler equations, with uniform and non-uniform mean flows in two dimensions. A polynomial basis set comprising of Lagrange polynomials is used. The effective implementation of boundary conditions is discussed. Some recommendations are made for improvements in the algorithm and extensions to three spatial dimensions.

### 3.1 DG method applied to the LEE in the time domain

The governing equations are the linearized Euler equations in two dimensions, written in non-dimensional, conservation form:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{A}_r \mathbf{q}}{\partial x_r} + \mathbf{C} \mathbf{q} = \mathbf{s} \quad \text{for } r = 1, 2 \quad (3.1)$$

with  $x_1 \equiv x$ ,  $x_2 \equiv y$ ,

$$\mathbf{q} = \begin{Bmatrix} \rho \\ u \\ v \\ p \end{Bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} M_x & \rho_o & 0 & 0 \\ 0 & M_x & 0 & 1/\rho_o \\ 0 & 0 & M_x & 0 \\ 0 & \gamma P_o & 0 & M_x \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} M_y & 0 & \rho_o & 0 \\ 0 & M_y & 0 & 0 \\ 0 & 0 & M_y & 1/\rho_o \\ 0 & 0 & \gamma P_o & M_y \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{\rho_o} \left( M_x \frac{\partial M_x}{\partial x} + M_y \frac{\partial M_x}{\partial y} \right) & -\frac{\partial M_y}{\partial y} & \frac{\partial M_x}{\partial y} & -\frac{\partial(1/\rho_o)}{\partial x} \\ \frac{1}{\rho_o} \left( M_x \frac{\partial M_y}{\partial x} + M_y \frac{\partial M_y}{\partial y} \right) & \frac{\partial M_y}{\partial x} & -\frac{\partial M_x}{\partial x} & -\frac{\partial(1/\rho_o)}{\partial y} \\ 0 & (1-\gamma) \frac{\partial P_o}{\partial x} & (1-\gamma) \frac{\partial P_o}{\partial y} & (\gamma-1) \left( \frac{\partial M_x}{\partial x} + \frac{\partial M_y}{\partial y} \right) \end{bmatrix} \quad (3.2)$$

and  $\mathbf{s}$  is a vector of time varying acoustic source terms. Here  $\rho, u, v, p$  are the acoustic density, velocity components and pressure fluctuation respectively, and  $\rho_o, M_x, M_y, P_o$  are the values of the mean density, velocity components and pressure respectively, nondimensionalized using appropriate reference values. The mean flow quantities can be spatially varying.  $\mathbf{A}_r$  denotes the coefficient matrices. The matrix  $\mathbf{C}$ , made up of the derivatives of the mean flow quantities, is called as the reaction matrix. The DG approximation in each element gives:

$$\frac{\partial \mathbf{q}^e}{\partial t} + [\mathbf{I} \otimes \mathbf{M}^e]^{-1} \{ \mathbf{k}^e + \mathbf{b}^e \} = -\mathbf{c}^e + \mathbf{s}^e \quad (3.3)$$

The number of discretized equations above is  $4N$ , and corresponds to the values of each variable in  $\mathbf{q}$  at all the nodes in an element. Here  $\mathbf{q}^e$  is a  $4N$  vector of the nodal values for all variables in  $\mathbf{q}$ :

$$\mathbf{q}^e = \{ \rho_j^e, u_j^e, v_j^e, p_j^e \}^T, \quad \text{for } j = 1 \dots N. \quad (3.4)$$

$\mathbf{s}^e$  is the vector of the nodal acoustic source values for each equation in (3.1).  $\mathbf{I}$  is a  $4 \times 4$  identity matrix,  $\mathbf{M}^e$  is given by equation (2.12), and  $\otimes$  represents the Kronecker product, defined in Appendix C. The mass matrix is easily invertible, and the inverse of the mass matrix is computed analytically prior to the main computation. The calculation of the stiffness vector,  $\mathbf{k}^e$ , and the reaction vector  $\mathbf{c}^e$  involve flux interpolation. The edge flux vector,  $\mathbf{b}^e$  also involves flux interpolation, along with the calculation of the Riemann flux. A full description of the evaluation of the terms  $\mathbf{k}^e$ ,  $\mathbf{b}^e$  and  $\mathbf{c}^e$  is provided in Appendix B.

## 3.2 Boundary conditions

An effective implementation of the boundary conditions is crucial for obtaining an accurate solution in aeroacoustics. The common types of boundary conditions encountered in aeroacoustics are the reflecting boundary condition, the non-reflecting boundary condition, and the inflow-outflow boundary condition. The last two are implemented when a finite computational domain has to represent an infinite or semi-infinite physical domain that is often encountered in CAA. In this case, precise boundary conditions are not known at the computational boundaries. The numerical conditions imposed at the computational boundaries must make the flow field behave as if the computational domain were larger: the waves must be allowed to exit the domain without any reflections. Boundary conditions used in CFD are usually based on characteristic formulations [22]. But these formulations lead to spurious reflections as the computational domain gets reduced, and as the angle of incidence of the waves departs from the direction normal to the boundary. One of the ways to reduce the reflections from the boundaries has been to incorporate damping layers [28] around the physical domain. The Perfectly Matched Layer (PML) technique is one such method, and has become popular among the aero-

coustic community in recent years.

The implementation of boundary conditions in the DG algorithm is simple, since the algorithm for the interior elements can be easily modified to incorporate the boundary conditions. On the edges of the elements coinciding with the computational boundary, only the expression for the Riemann flux is modified. A discussion of the characteristic and PML boundary conditions is given below. Atkins [22] presents a comprehensive report on the implementation of reflective and characteristic boundary conditions using the DG method.

### 3.2.1 Characteristic boundary conditions

The characteristic formulation is obtained by splitting the normal component of the flux at the boundary into components that are leaving and entering the domain. Let  $\vec{n}_r$  represent the normal to the computational boundary at a node on the edge of an element that lies on the computational boundary. Then, the characteristic formulation for the nodes on the boundary is written as [22],

$$\mathbf{F}_n = [\mathbf{A}_r \cdot \vec{n}_r] \mathbf{q} = \mathbf{P}[\lambda] \mathbf{P}^{-1} \mathbf{q} = \underbrace{\mathbf{P}[\lambda^+] \mathbf{P}^{-1} \mathbf{q}}_{interior} + \underbrace{\mathbf{P}[\lambda^-] \mathbf{P}^{-1} \mathbf{q}}_{exterior}. \quad (3.5)$$

Here,  $\mathbf{P}$  is the eigenvector matrix and  $[\lambda]$  is the matrix composed of the eigenvalues of  $\frac{\partial \mathbf{F}_n}{\partial \mathbf{q}} = [\mathbf{A}_r^e \cdot \vec{n}_r]$ .  $[\lambda^{+-}]$  is the matrix containing positive or negative eigenvalues only. The negative eigenvalues correspond to the exterior solution, and are usually set to zero. A way of implementing the characteristic boundary condition is to decompose the nodal values themselves into interior and exterior components, and to set the exterior component to zero, as:

$$\mathbf{q}_{interior} = \mathbf{P}[I^+] \mathbf{P}^{-1} \mathbf{q}, \quad (3.6)$$

where  $I^+$  is a diagonal matrix containing ones only in places that correspond to the positive eigenvalues,  $\lambda^+$ . This value of  $\mathbf{q}$  is used in the expression for the Riemann flux in equation (2.7). The Riemann flux is one sided on a boundary. That is, equation (2.7) is re-written using (3.6) as:

$$\bar{\mathbf{F}}_R^e = \bar{\mathbf{F}}_-^e = [\mathbf{A}_r^e \cdot \vec{n}_r] \mathbf{q}_{interior}^e = \mathbf{P}[I^+] \mathbf{P}^{-1} \mathbf{q}^e \quad (3.7)$$

Since the LEE can not be decoupled into separate equations, the splitting in the equation (3.5) is not accurate for waves that are not normal to the boundary. Hence characteristic boundary conditions result in spurious reflections for such waves.

### 3.2.2 The PML technique

#### Background

The Perfectly Matched Layer technique was originally introduced by Berenger [29] in the field of electromagnetics. It was later improved by many other researchers, such as Hu [30], for applications in aeroacoustics. The original PML technique consists of splitting the LEE in the normal and tangential directions to the physical domain boundary, in a buffer layer surrounding the physical domain. For the homogeneous LEE, the splitting can be written as:

$$\frac{\partial \mathbf{q}_1}{\partial t} + \sigma_x \mathbf{q}_1 + \mathbf{A}_1 \frac{\partial \mathbf{q}}{\partial x} = 0 \quad (3.8)$$

$$\frac{\partial \mathbf{q}_2}{\partial t} + \sigma_y \mathbf{q}_2 + \mathbf{A}_2 \frac{\partial \mathbf{q}}{\partial y} = 0 \quad (3.9)$$

with  $\mathbf{q}_1$  and  $\mathbf{q}_2$  being the split variables. Damping terms are added to the split equations such that only the waves corresponding to the normal direction to the physical domain

boundary are damped, as illustrated in Fig. 3.1. Also, the damping is zero at the

|                                  |                                  |                                  |
|----------------------------------|----------------------------------|----------------------------------|
| $\sigma_x > 0$<br>$\sigma_y > 0$ | $\sigma_x = 0$<br>$\sigma_y > 0$ | $\sigma_x > 0$<br>$\sigma_y > 0$ |
| $\sigma_x > 0$<br>$\sigma_y = 0$ |                                  | $\sigma_x > 0$<br>$\sigma_y = 0$ |
| $\sigma_x > 0$<br>$\sigma_y > 0$ | $\sigma_x = 0$<br>$\sigma_y > 0$ | $\sigma_x > 0$<br>$\sigma_y > 0$ |

**Figure 3.1.** Damping parameters in the PML.

beginning and increases smoothly in the PML. The damping of waves in this manner assure that there are no reflections at the interface of the PML and the physical domain, and hence the name “perfectly matched”. Because of this, the PML technique is very effective as an absorbing boundary condition with a relatively small number of grid points compared to other types of absorbing boundary conditions [31].

The initial version of the PML technique using split variables was reported to support unstable solutions [32]. Hu [33] improved on this unstable version and derived a stable version of the PML in unsplit variables. In the present work using the DG methods in the time domain, the stable version of the PML is used, with characteristic boundary conditions at the outer boundaries of the PML.

### **Stabilized PML**

The stable version of the PML in unsplit variables, as developed by Hu [33] corresponds to a uniform mean flow in one direction. For a uniform mean flow in the  $x$ -direction, the

PML formulation corresponding to the LEE (3.1) is:

$$\frac{\partial \mathbf{q}}{\partial t} + \mathbf{A}_r \frac{\partial \mathbf{q}}{\partial x_r} + \sigma_x \mathbf{A}_1 \frac{\partial \mathbf{z}}{\partial x} + \sigma_y \mathbf{A}_2 \frac{\partial \mathbf{z}}{\partial y} + (\sigma_x + \sigma_y) \mathbf{q} \quad (3.10)$$

$$+ \sigma_x \sigma_y \mathbf{z} + \sigma_x \frac{M_x}{(1 - M_x^2)} \mathbf{A}_1 (\mathbf{q} + \sigma_y \mathbf{z}) = \mathbf{s}$$

$$\frac{\partial \mathbf{z}}{\partial t} = \mathbf{q}, \quad (3.11)$$

where  $\mathbf{z}$  is the additional or PML variable. The above set of equations is obtained by transforming the coordinates and applying a change of variables in the frequency domain for the LEE (3.1). This change of variables is equivalent to splitting the LEE [33] as in equations (3.8) and (3.9). The formulation in (3.11) can be extended to a case with a uniform mean flow in both the directions by a coordinate rotation prior to the transformation and change of variables associated with the stabilized PML formulation. The stable PML equations for a mean flow in both the  $x$  and  $y$  directions can be written as:

$$\frac{\partial \mathbf{q}}{\partial t} + \mathbf{A}_r \frac{\partial \mathbf{q}}{\partial x_r} + \sigma_x \mathbf{A}_1 \frac{\partial \mathbf{z}}{\partial x} + \sigma_y \mathbf{A}_2 \frac{\partial \mathbf{z}}{\partial y} + (\sigma_x + \sigma_y) \mathbf{q} + \sigma_x \sigma_y \mathbf{z} \quad (3.12)$$

$$+ \frac{\sqrt{\sigma_x^2 + \sigma_y^2} Q}{1 - Q^2} (\beta_1 \mathbf{A}_1 \mathbf{q} + \beta_2 \mathbf{A}_2 \mathbf{q}) + \frac{\sigma_x \sigma_y Q}{1 - Q^2} (\beta_1 \mathbf{A}_1 \mathbf{z} + \beta_2 \mathbf{A}_2 \mathbf{z}) = \mathbf{s}$$

$$\frac{\partial \mathbf{z}}{\partial t} = \mathbf{q}, \quad (3.13)$$

$$\text{with } \beta_1 = \frac{M_x}{Q}, \quad \beta_2 = \frac{M_y}{Q}, \quad \text{and } Q = \sqrt{M_x^2 + M_y^2}$$

The damping parameters  $\sigma_x$  and  $\sigma_y$  are zero at the beginning of the PML, and are varied smoothly within the PML to reach a maximum value at the end of the computational domain. The variation could be chosen, for example, to be a polynomial function of the distance from the beginning of the PML [33]. In the implementation of the PML

using the DG method in this thesis, the damping parameters are treated as piecewise constants in the PML; they are taken as a constant over each element in the PML. Consider an element in the PML with  $x_{cen}$  and  $y_{cen}$  as the coordinates of the centroid of an element inside the PML. The damping parameters are chosen as a function of the non-dimensional distance from the beginning of the PML:

$$\sigma_x = \sigma \left( \frac{x_{cen}}{L_{xPML}} \right)^k \quad (3.14)$$

$$\sigma_y = \sigma \left( \frac{y_{cen}}{L_{yPML}} \right)^k \quad (3.15)$$

and  $k > 0$ . Here  $L_{xPML}$  and  $L_{yPML}$  are the thicknesses of the PML in the  $x$  and  $y$  directions respectively.  $\sigma$  is the maximum value of  $\sigma_x$  and  $\sigma_y$ , and is usually chosen close to unity. The DG formulation applied to equations (3.13) and (3.14) results in the following relations:

$$\begin{aligned} \frac{\partial \mathbf{q}^e}{\partial t} + [\hat{\mathbf{M}}^e]^{-1} \{ \mathbf{k}^e + \mathbf{b}^e \} + [\hat{\mathbf{M}}^e]^{-1} \{ \mathbf{k}^{ez} + \mathbf{b}^{ez} \} \\ + (\sigma_x + \sigma_y) \mathbf{q}^e + \sigma_x \sigma_y \mathbf{z}^e + \mathbf{g}^z = \mathbf{s}^e \end{aligned} \quad (3.16)$$

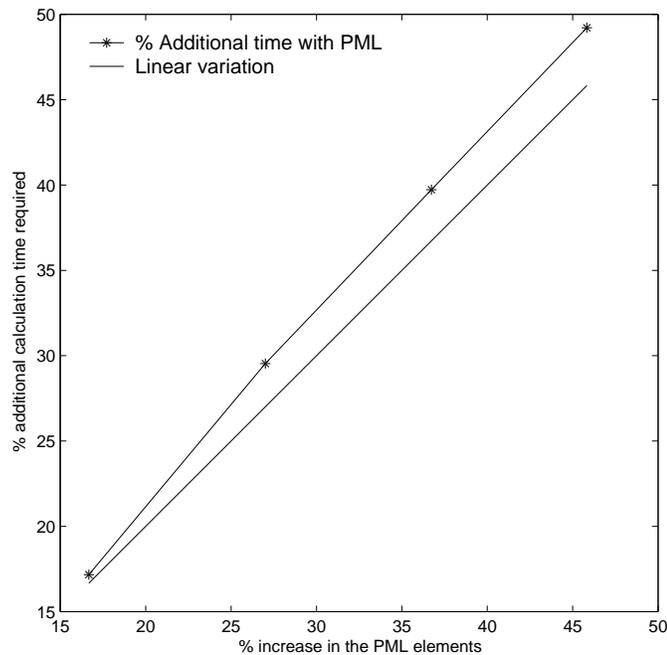
$$\frac{\partial \mathbf{z}^e}{\partial t} = \mathbf{q}^e, \quad (3.17)$$

$$\text{with } \mathbf{g}^z = \frac{\sqrt{\sigma_x^2 + \sigma_y^2} Q}{1 - Q^2} (\beta_1 \mathbf{A}_1 + \beta_2 \mathbf{A}_2) \otimes \mathbf{q}^e + \frac{\sigma_x \sigma_y Q}{1 - Q^2} (\beta_1 \mathbf{A}_1 + \beta_2 \mathbf{A}_2) \otimes \mathbf{z}^e,$$

$\mathbf{k}^{ez}$  and  $\mathbf{b}^{ez}$  are formed by replacing  $A_r$  with  $\sigma_x A_r$  in the formulation of  $\mathbf{k}^e$  and  $\mathbf{b}^e$ .

The additional computational time required using a PML in a problem is directly proportional to the number of elements in the PML. This can be understood by looking at equations (3.16) and (3.16). The additional calculations with the PML include the calculation of the edge flux, the volume flux and three vectors for the PML variable, and the solution of another simple differential equation. Figure 3.2 shows the addi-

tional time taken for varying thickness of the PML, for a benchmark problem which is discussed in the next section, using a second order polynomial basis set. The fraction of the additional time is directly proportional to the fraction of the elements within the PML. The additional difference between the linear variation and the computation time is attributed to the time required for the three vectors and the solution of equation (3.16). For calculations in three dimensions also, the additional time required is expected to vary in a fashion similar to the Fig. 3.2.



**Figure 3.2.** The time required using the PML.

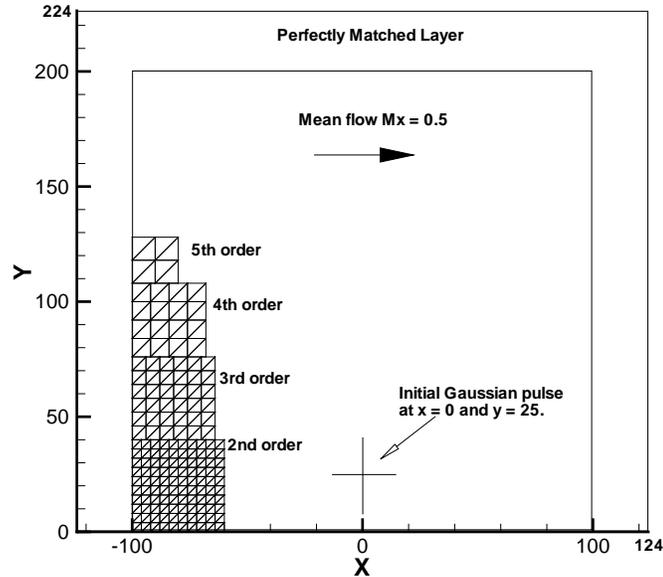
### 3.3 Applications

Here, an application of the DG method in the time domain for solving the LEE in uniform and non-uniform mean flow cases is considered. Results from a  $h - p$  refinement study for the uniform mean flow case are presented. A comparison of the computation

times for the algorithms using Lagrange and moment polynomials is made. Also, the variation of the error in the solution with the dissipation parameter in the Lax-Friedrich's flux is studied.

### 3.3.1 Uniform mean flow - propagation of an initial pulse

The 4th benchmark problem from the ICASE/LARC workshop [34] is considered. The problem concerns the propagation of an initial Gaussian pulse near a wall in a uniform mean flow of Mach number 0.5 in the positive  $x$  direction. The mean flow quantities in equation (3.1) are non dimensionalized with respect to the speed of sound and mean thermodynamic properties. The problem domain is  $(-100 < x < 100)$  and  $(0 < y < 200)$  with the initial pulse located at  $(0, 25)$ , as shown in Fig. 3.3. The lower boundary is treated as a rigid wall. The inflow and outflow boundary conditions are implemented using the PML. A constant damping parameter of value  $\sigma_x = \sigma_y = 0.2$  is used in the inlet, outlet and top PML, which is shown in the Fig. 3.3. A thickness of 8 to 10 elements in the PML is expected to reduce the the amplitude of the acoustic waves to zero at the outer edge of the PML [33]. In such a case, the value of the acoustic variables could be specified to be zero at the end of the PML. In this work, however, the thickness of the PML is taken to be 5 to 6 elements, to reduce the computational time. Characteristic boundary conditions [22] are used at the outer edge of the PML. Since the acoustic waves are reduced to very low values towards the end of the PML, the reflections caused by the characteristic boundary conditions would be weak. These weak reflections are in turn completely damped off in the PML. Basis sets made up of Lagrange polynomials of up to 5th degree are used on an unstructured grid consisting of uniform triangles. Fig. 3.4 shows the propagation of the pulse in the mean flow after 100 non-dimensional time units, using a basis set of degree one. The acoustic amplitudes are damped effec-

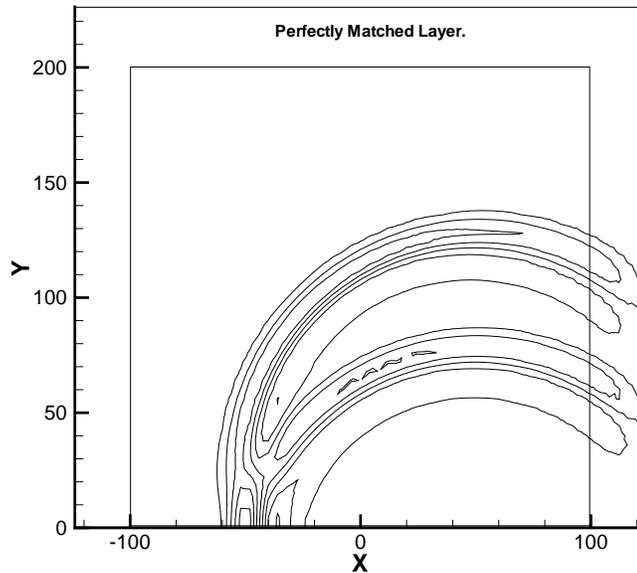


**Figure 3.3.** The computation domain: the grid for increasing degree of polynomials.

tively in the PML. The solution after the pulse has propagated for a time of 200 time units is shown in Fig. 3.5a. The solution obtained without using the PML is shown in Fig. 3.5b. The reflections from the outflow boundary due to the characteristic boundary condition are clearly seen. In this case, the PML is in  $x < -88$ ,  $x > 88$  and  $y > 172$ . The computational domain is from  $-108 < x < 108$  and  $0 < y < 192$ , and the element size  $h = 4$ . The degree of the basis set polynomials is  $p = 2$ . A time step of 0.1 non-dimensional units is used with a third order three stage total variation bounded Runge Kutta (TVBRK) scheme [17].

### ***h* – *p* refinement**

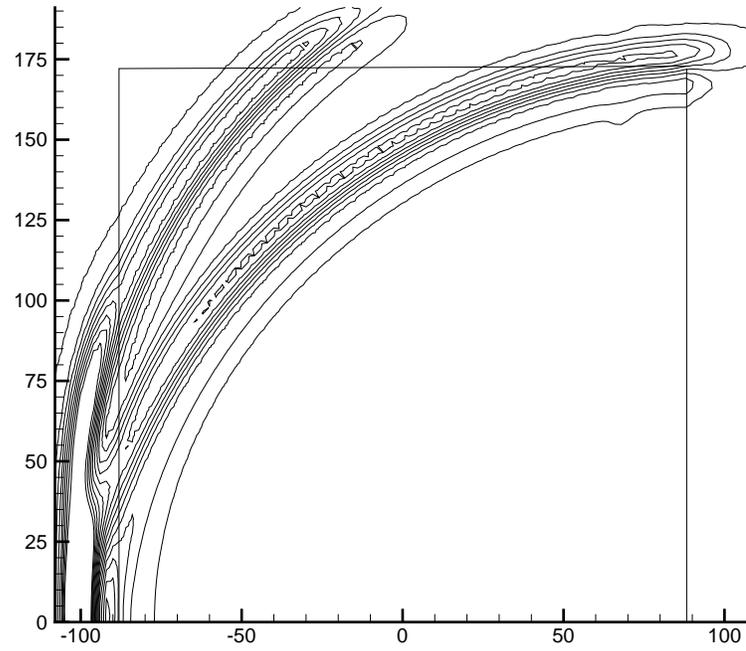
Decreasing the size of the elements in the domain, while keeping the degree of the approximation a constant, is termed as the  $h$  refinement. This is also equivalent to increasing the number of elements. On the other hand, increasing the degree of the



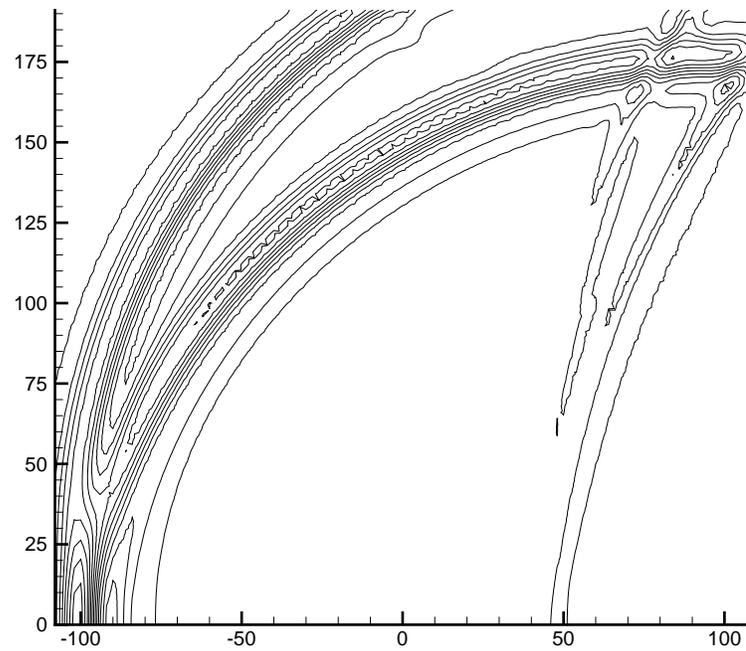
**Figure 3.4.** The acoustic propagation after 100 time units.

approximating polynomial without a change in the size of the elements is termed as the  $p$  refinement. For a study of the  $p$  refinement, the mesh size is increased with an increase in the degree of the basis set polynomials, so that the total number of nodes in the domain remains almost the same. For example, the element size in the case with 5th order polynomials is five times greater than the element size with first order. The grids for the 2nd, 3rd, 4th and the 5th order calculations are shown in Fig. 3.3a. The number of elements used in the computation are 27776, 6944, 3108, 1736 and 1056, for polynomial orders from one to five respectively. The time step for each order is kept the same, at  $\Delta t = 0.25$ , to keep the temporal error at the same level for all the five cases. The time marching scheme used is a third order TVBRK method for all the cases.

Figure 3.6a shows a contour plot of the computed solution and the exact solution [34] for  $p = 2$  at  $t = 100$ . The variation of the difference between the two solutions is shown in Fig. 3.6b. In the workshop on benchmark problems which included this pulse

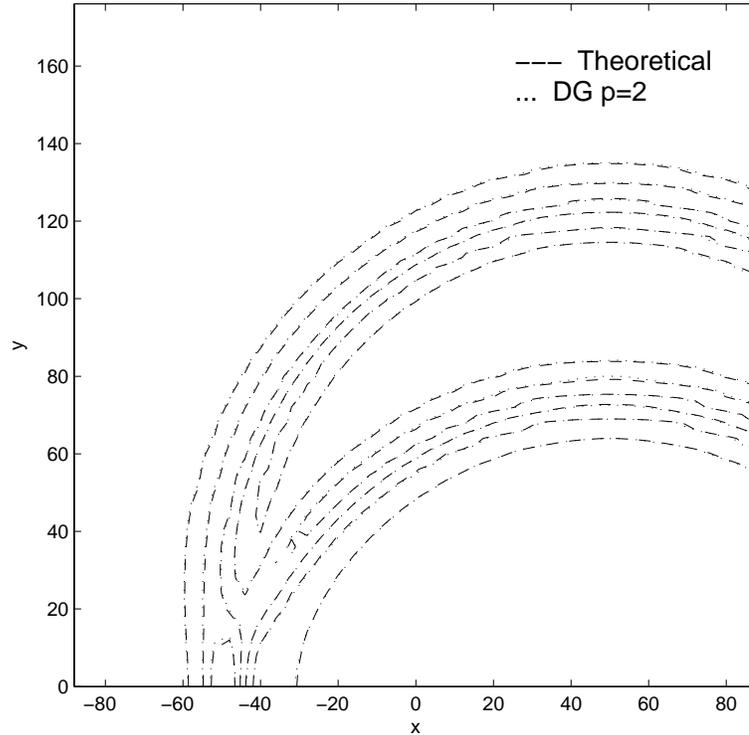


(a)



(b)

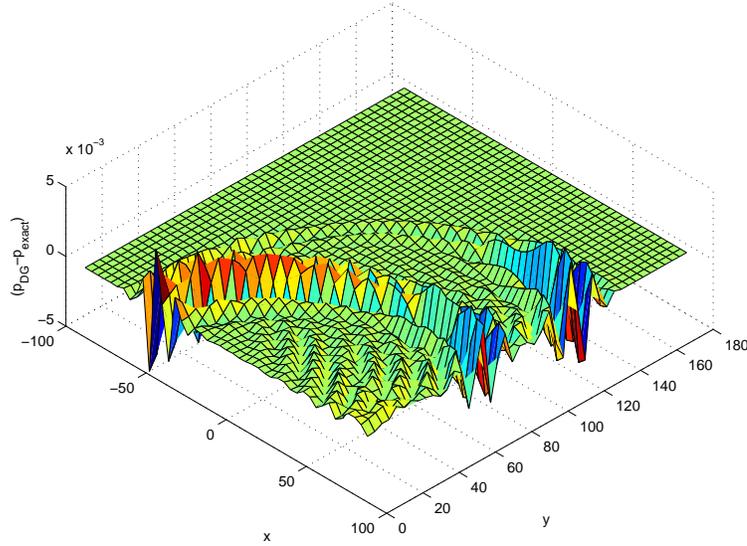
**Figure 3.5.** Propagation of the pulse: (a) with and (b) without the PML.



**Figure 3.6.** Contour plot of the calculated and the exact pressure

propagation problem, the solution along the line  $x = y$  is compared with the exact solution. Here, the values of pressure from the DG method are compared with the exact solution along the line  $x = y$  at  $t = 60$ , in Fig. 3.8. The values at the first peak, within the dashed box in Fig. 3.8a are magnified and shown in Fig. 3.8b. It is seen that the solution improves with the increase in order, even with a dramatic increase in element size.

The  $L^2$  norm of the difference between the numerical and exact solutions is shown in Fig. 3.9a, for varying degrees of the basis set polynomial. The error is computed as the difference between the computed and the exact solution at  $x = y$ . Finally, the computational time required for all the cases is shown in Fig. 3.9b. The calculations have been performed on a 800 MHz single processor. The time required drops by almost a factor of four from the  $p = 1$  to  $p = 5$  case. This is attributed to the elimination of boundary integrals inside the larger element in the latter case.

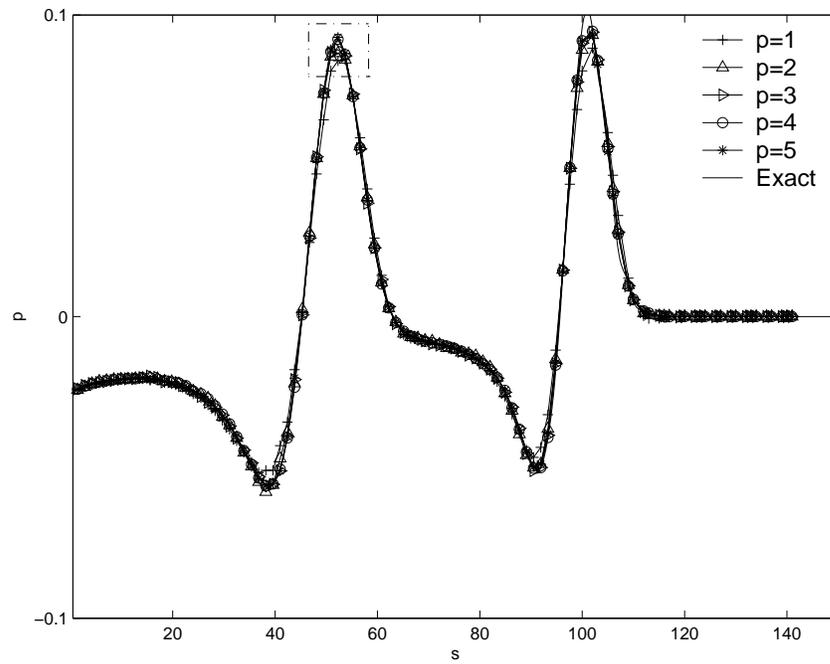


**Figure 3.7.** Variation in the error between the calculated and the exact pressure

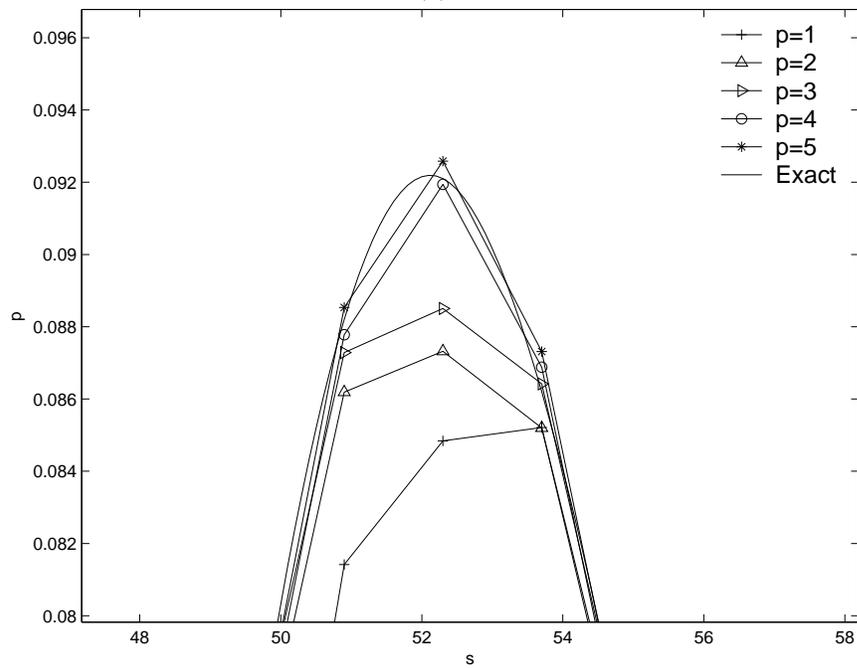
### Efficiency improvements from Lagrange interpolation

A comparison between the computation times for the quadrature-free methods using Lagrange and moment polynomials has been made for basis sets of up to 4th degree polynomials. The problem domain and the initial conditions are kept the same, but periodic boundary conditions are implemented along all the boundaries, in order to compare only the element calculations, but not the time required for the implementation of PML and characteristic boundary conditions. No algorithm optimizations have been performed and the sparsity of the matrices is not used in both the cases. The ratio of the computational time required using the Lagrange basis to that using the moment polynomial basis set is presented in Fig. 3.10. The use of Lagrange polynomials reduces the effort by about 36% when 4th degree polynomials are used in the basis set. This significant reduction is attributed to the absence of transformation matrices in the calculation of the inter-element boundary flux, using Lagrange polynomials.

The flux expressions in this benchmark problem are linear, of the form  $M_x \vec{u}$  where

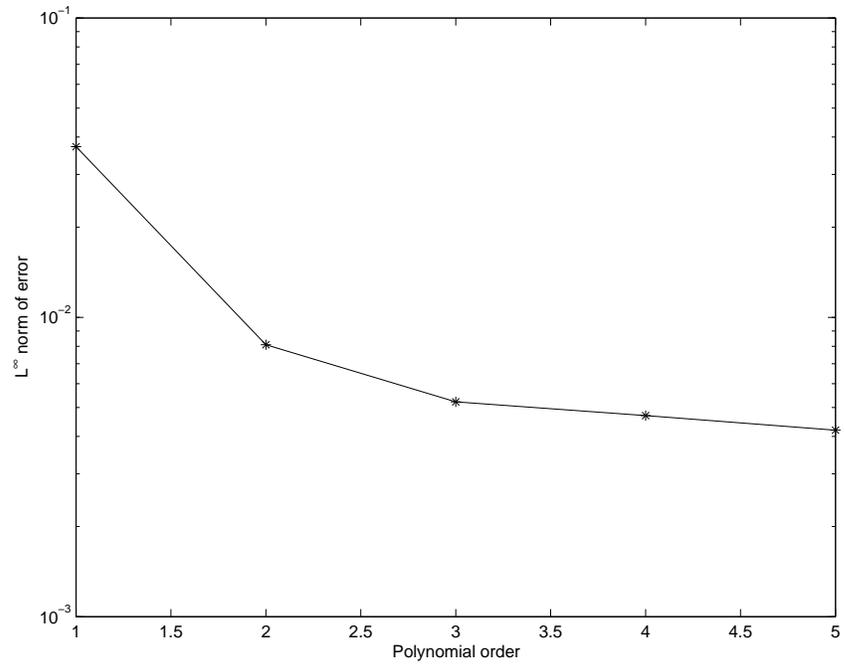


(a)

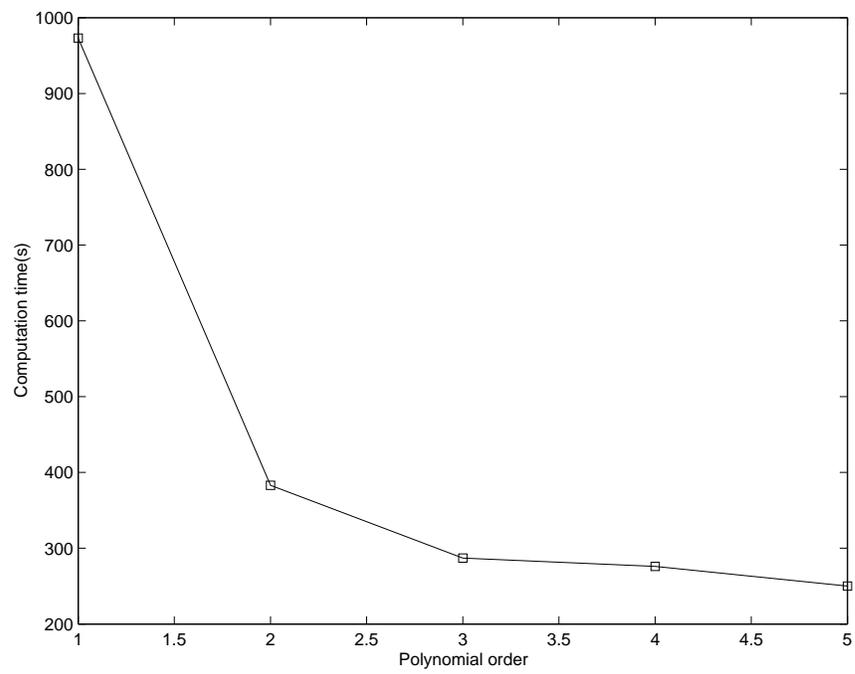


(b)

**Figure 3.8.** The solution. (a) Along the line  $x = y$ . (b) the first peak magnified.

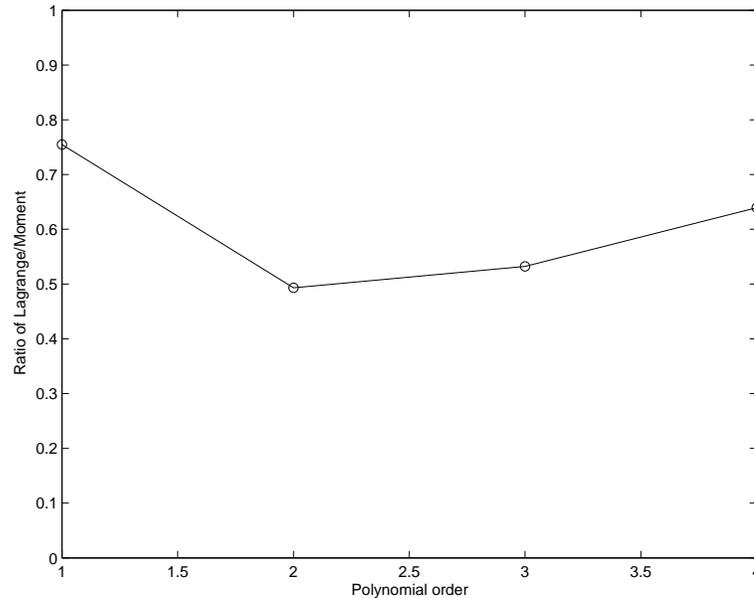


(a)



(b)

**Figure 3.9.** The performance. (a)  $L^\infty$  norm of the error. (b) The computation time.



**Figure 3.10.** Ratio of computational times using Lagrange to moment polynomials.

$M_x$  is a constant of value 0.5. In the case of moment polynomials, the flux terms are calculated by interpolating the acoustic variables and then multiplying the expansion coefficients by the constant mean flow Mach number. In the case of Lagrange polynomials, the flux terms are interpolated separately, as mentioned in the previous section. This explains the increase in the ratio of the computational times in Fig. 3.10 for higher order polynomials. However when the mean flow is non-uniform, the flux expressions have to be interpolated separately and this would increase the operation count in the case moment polynomials. Hence, for more realistic problems in aeroacoustics involving non-uniform mean flows, the reduction in computational effort is expected to be greater using Lagrange polynomials.

### **Dissipation parameter**

In the DG method, the discontinuities between the elements are “smoothed out” by adding the dissipation terms, in which the jumps across the edges of the element are

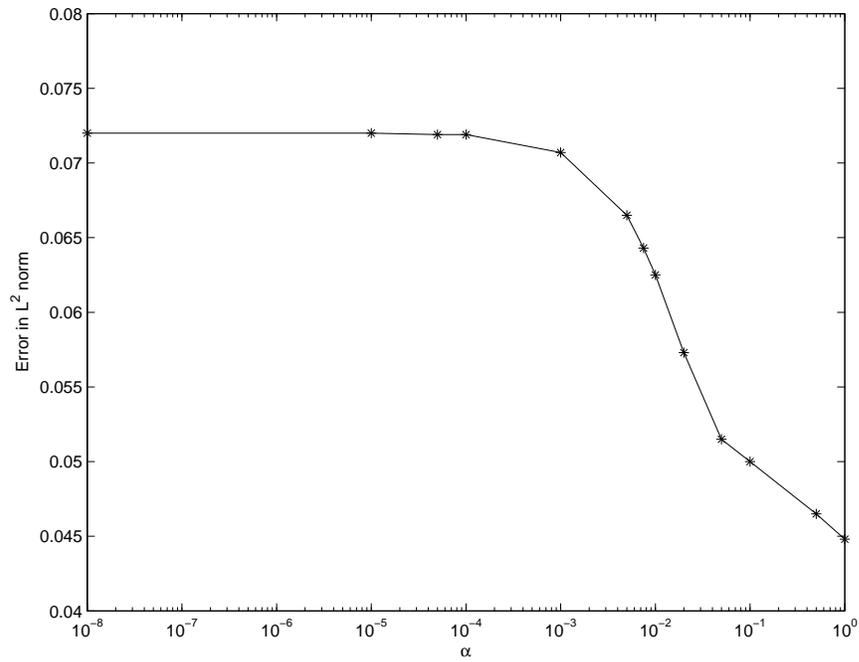
used. As explained in [8], the artificial dissipation is directly proportional to the residual inside the element. Hence as the polynomial degree of approximation increases, the artificial viscosity diminishes.

The selection of the dissipation parameter  $\alpha$  in the Lax-Friedrich's flux in equation (2.7) affects the stability of the scheme, and also the error in the solution. Consider the one dimensional advection equation in section 2.3, with an advection speed of unity. The variation of the error in the  $L^2$  norm of the solution with the choice of the dissipation parameter for  $p = 1$  is shown in Fig. 3.11a. The error decreases with an increase in the dissipation parameter, and the scheme becomes unstable for values of  $\alpha > a$ , the advection speed in equation (2.15). In this case,  $a = 1$ . The convergence of the solution for  $h$  refinement in two dimensional calculations for  $p = 1$  is shown in Fig. 3.11b. The problem is the LEE with uniform mean flow, with the configuration and grid as shown in Fig. 3.3a. The abscissa represents the number of nodes on the  $x$  axis in Fig. 3.3a, or essentially a decreasing size of the elements. The rate of convergence in the solution is  $h^p$  for  $\alpha = 0.02$ , and improves to  $h^{p+\frac{1}{2}}$  for  $\alpha = 0.8$ .

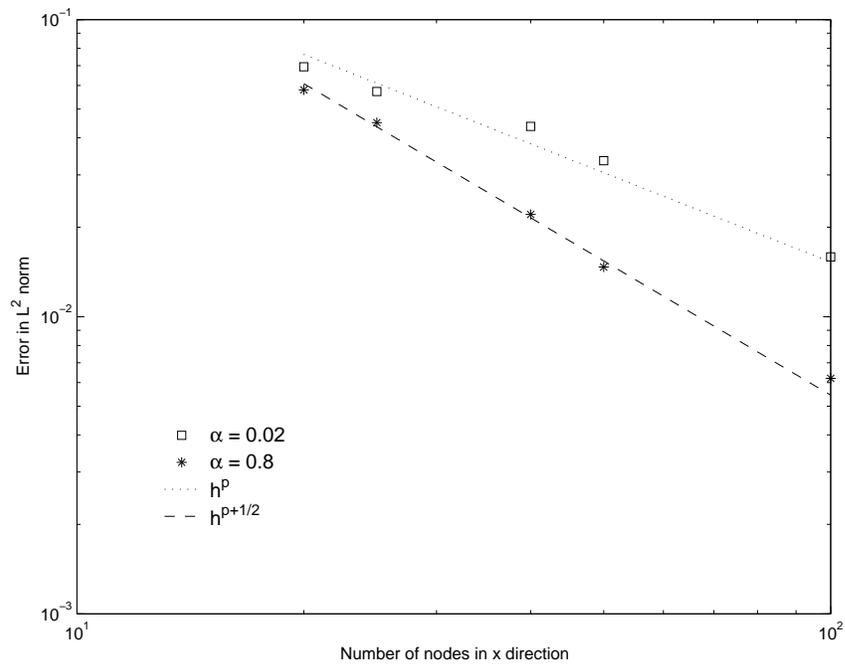
Conventionally, the value of  $\alpha$  is chosen [5] to be related to the eigenvalues of the trace of the average flux Jacobian,

$$\frac{1}{2} \left\{ \frac{\partial (\vec{F}_+ + \vec{F}_-)}{\partial q} \cdot \vec{n} \right\} \quad (3.18)$$

on the edges of the element. In a recent work, Hu and Atkins [14] have conducted an eigensolution analysis of the discontinuous Galerkin method in one dimension. They show that the spurious non physical modes in the one dimensional conservation equations are damped when the dissipation parameter in the Lax-Friedrich's flux is near the minimum eigenvalue of the flux Jacobian.



(a)



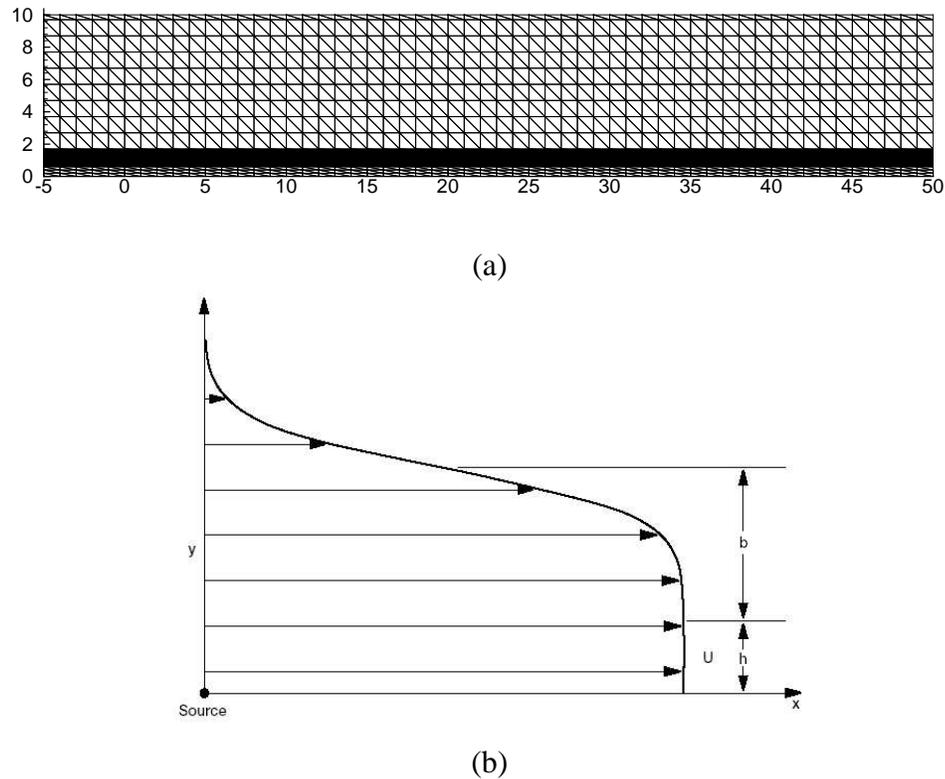
(b)

**Figure 3.11.** The effect of the dissipation parameter  $\alpha$  on the solution. (a) One dimensional calculations (b)  $h$ -convergence in two dimensional calculations.

### 3.3.2 Non-uniform mean flow - acoustic propagation in a shear layer

The 5th benchmark problem from the CAA workshop [35] is considered. It involves the excitation of instability waves in the shear layer of a two dimensional jet by an acoustic source. The mean flow is in the  $x$  direction only, but varies in the  $y$  direction in the shear layer. The problem domain is given by  $-5 < x < 50$  and  $0 < y < 10$ . Symmetry boundary conditions are imposed at the lower boundary. The computational region is extended in the  $x$  direction up to  $-15 < x < 60$  and the mean flow is accelerated to supersonic speed in the buffer region. This buffer region is not treated as a PML, since the mean flow is non-uniform. The top boundary is extended to  $y = 20$  and a PML is used from  $10 < y < 20$ . Radiating boundary conditions are used at the outer boundaries. A grid consisting of triangles, similar to the one shown in Fig. 3.12a is used. The grid is refined between  $y = 0.6$  to  $y \approx 1.7$ , where the gradient of the mean flow,  $\partial U_0/\partial y$ , is high, and the acoustic solution varies greatly. The variation of the mean flow from  $y = 0$  to  $y \approx 1.7$  is represented in Fig. 3.12b.

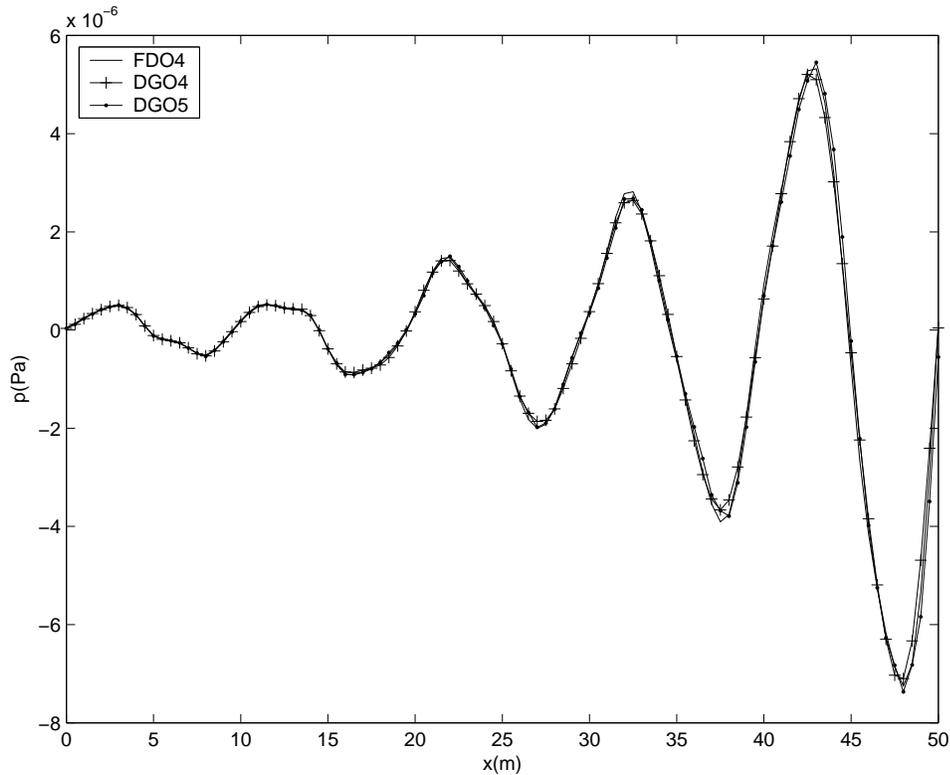
The high-gradient, non-uniform mean flow region in the  $y$  direction is divided into nine equal parts between  $0.6 < y < 1.7$ . The solution is computed using basis sets of Lagrange polynomials of up to 5th degree. A three stage third order Runge Kutta method is used to march in time, with the non-dimensional time step kept constant at  $\Delta t = 0.3333 \times 10^{-3}T$ , where  $T$  is the period of the source. The numerical solution using the DG method is compared to that using a fourth order finite difference method by Agarwal and Morris [36]. The instantaneous pressure values at the end of a cycle of oscillations at  $y = 1$  for the non dimensional frequency  $St = 0.14$  are plotted in Fig. 3.13 for polynomial orders 4 and 5. The variation of the mean square of the pressure at  $y = 10$  is plotted in Fig. 3.14. The solution is compared with the second order accurate finite difference solution obtained by Agarwal and Morris [36].



**Figure 3.12.** (a) The grid for the 5th benchmark problem domain. (b) The profile of mean flow in the clustered region.

### Oscillations in higher order

Using higher orders of the Lagrange polynomial basis set, oscillations in the solution, as shown in Fig. 3.15a, are observed. These oscillations are in turn amplified in the shear layer, as shown in Fig. 3.15b, after two periods of the acoustic source. If characteristic boundary conditions alone are used at the outflow boundaries, secondary oscillations reflected from these boundaries enter the domain, further contaminating the solution. To minimize the reflections from the boundaries, a buffer region in which the mean flow is accelerated to supersonic speeds is used at the right boundary, and a PML is used at the top boundary. By using a value for  $\alpha$  near the minimum of the eigenvalues of the



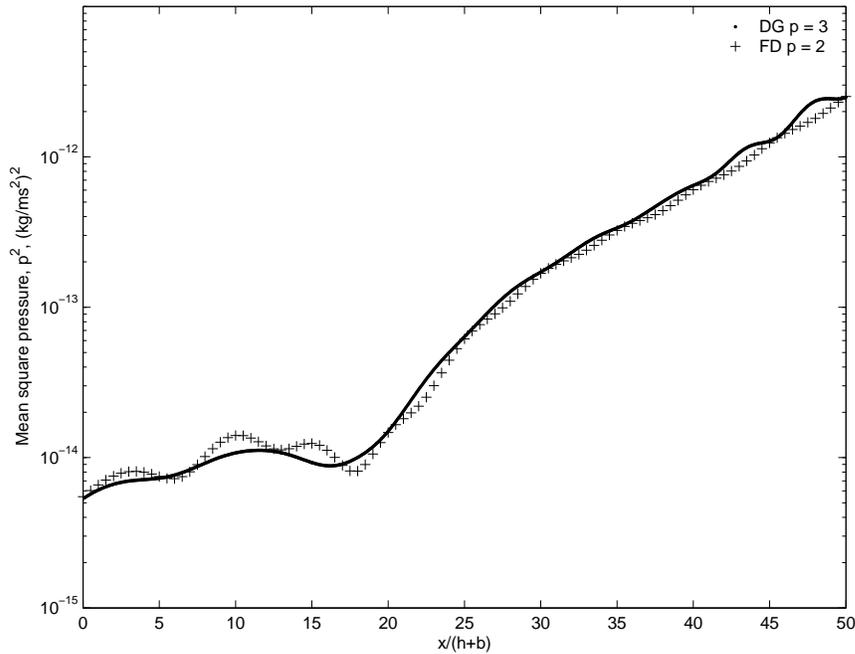
**Figure 3.13.** Solution to the 5th benchmark problem: Instantaneous pressure at  $y = 1$  at the end of a source cycle.

flux Jacobian on a refined grid, these oscillations are reduced.

## 3.4 Future directions

### 3.4.1 Interpolation error: the choice of nodes

The choice of the nodes on the elements plays an important role in determining the accuracy of the solution, as the polynomial order of approximation is increased. Consider the interpolation of a function in one dimension. A grid with equidistant nodes gives rise to oscillations at the end points, or the boundaries. This behavior is termed as the Runge phenomenon [37]. The nodes must be clustered near the end points to avoid such oscillations. Such nodal sets can be chosen, for example, to be the zeros of classical

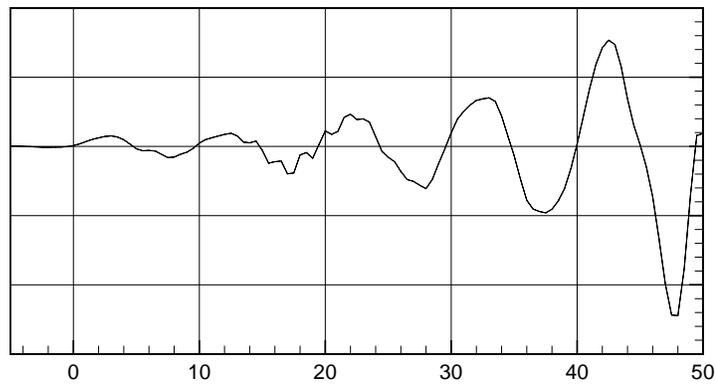


**Figure 3.14.** Solution to the 5th benchmark problem:  $\overline{p^2}$  at  $y = 10$ .

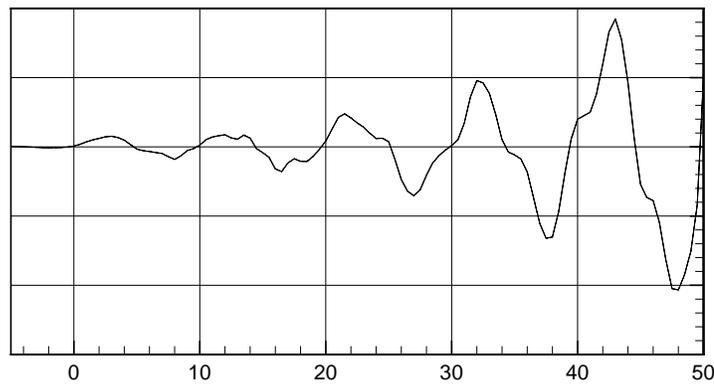
orthogonal polynomials and their derivatives. The choice is not so simple in the case of two dimensional and three dimensional elements. Chen and Babuska [38, 39] have developed nodal sets for minimizing interpolation errors on triangles and tetrahedrons. The use of such nodal sets in the DG method will further improve the accuracy of the solution.

### 3.4.2 Extension of the algorithm to three dimensions

The extension of the DG solver to three spatial dimensions is an obvious next step. The process of extension is straightforward, but demands substantial effort. Some of the issues to be addressed are the preprocessing, the analytical evaluation of element integrals, the implementation of the PML in three dimensions, and the postprocessing of the finite element solution. Further, the parallelization of a three dimensional solver, although identical to the two dimensional case, demands a significant effort on account



(a)



(b)

**Figure 3.15.** Oscillations in the non-uniform mean flow region with a 2nd order basis set. (a) At the end of a cycle (b) after two cycles.

of the additional details of the mesh and partitions. These aspects are discussed below, with an assumption that the three dimensional element is a tetrahedron, and the basis set is made up of Lagrange interpolation polynomials.

### Preprocessing

1. As in two dimensions, the three dimensional solver requires the information of the neighbors. The element numbers of the neighbors have to be extracted from the conventional FEM data, namely, the coordinates of the nodes and the connec-

tivity of the elements. However, in  $3d$ , in addition to the numbers of elements neighboring the four surfaces, information is also needed about the arrangement of the nodes on the surface of the neighboring element. This is necessary for the calculation of the edge flux, where the nodal values need to be multiplied with the rows of the mass matrix for  $2d$ .

2. The preprocessor which has to be developed for the parallel solver, discussed in Chapter 4, needs changes for a  $3d$  grid. Instead, the use of a readily available parallel partition-preprocessing software such as PARMETIS [40] could also be explored. However, with such a usage, the information for the PML domains and elements needs to be extracted.

### Element volume matrices

The element mass and stiffness matrices need to be computed for  $3d$  calculations. The methodology for these calculations is exactly the same as that given in section 2.5. The only difference is in equation (2.35) which now has to be expressed for a tetrahedron. Consider a tetrahedron as shown in Fig. B.2 in Appendix B with the origin at the right angle vertex, and with unit length edges. The equivalent of the integral in equation (2.35) for this tetrahedron is written as:

$$\int \int \int \xi^k \eta^l \zeta^m d\xi d\eta d\zeta = \frac{k!l!m!}{(k+l+m+3)!} \quad \text{for } k, l, m \geq 0 \quad (3.19)$$

The coordinates of the nodes in a tetrahedron need to be selected based on the optimal choices, which have been discussed in section 3.4.1.

### **Element surface integrals**

In  $3d$ , the calculation of element surface integrals is similar to that explained in section 2.4.3, with Lagrange polynomials. The elements in the rows of the surface integral vectors corresponding to each surface in a tetrahedron are calculated by a dot product of the rows of the mass matrix in two dimensions (corresponding to the surface) with the nodal values on the surface. This is explained in section B.3.

### **PML calculations and postprocessing**

The PML calculations in  $3d$  are similar to those in  $2d$ , but with an additional term corresponding to the mean flow in the additional direction. The stabilized version of the PML equations in  $3d$  are provided by Hu [33]. While the basic structure of the solver remains the same, some additions are necessary to run the solver in three dimensions. As with the two dimensional results, the solution obtained in three dimensions is difficult to plot and analyze, especially for high order elements. There is a need to triangulate or create tetrahedrons from arbitrary points in both cases. Technical software packages such as Tecplot [41] can triangulate a set of arbitrary points in two dimensions. However, in three dimensions, a postprocessing algorithm may be necessary to assist the creation of tetrahedrons and extraction of data on the vertices.

### **3.4.3 A note on the efficiency of the $3d$ solver**

The extension of the solver to three dimensions introduces an additional variable, and corresponding flux terms. These demand additional memory and computation resources. The DG solver developed in this study has been already shown to be more efficient in two dimensions compared to the original quadrature free (QF) method proposed by Atkins and Shu [5]. A comparison of the two methods in  $3d$  for a given order of approx-

imation and grid is given below.

- The calculation of the element volume matrices and stiffness matrices are done prior to the main computation. The time marching takes the same amount of time in both the original QF method and the method proposed here, since the size of the matrices and vectors,  $N$ , is the same for both the cases.
- In the original QF method, the approximation of flux expressions takes significant time in  $3d$  compared to the method proposed here. This conclusion can be drawn on the basis of the operation count required for the flux approximation, especially when the flux expression is nonlinear or contains spatially varying components. The operation count using the original QF method is provided in [5], and could be as high as  $[N(p, d)][N(p + 1, d)]$ . On the other hand, the operation count using flux interpolation described in section 2.5 is only  $N$ . The expression for  $N(p, d)$  is given in equation 2.4. The value of  $N$  increases greatly with  $p$  for  $d = 3$ , and hence the approach described here becomes more efficient compared to the original QF method.
- Using moment polynomials as in the original QF method, the computation of element surface integrals involves the calculation and multiplication of surface transformation matrices,  $\mathbf{T}$ , as given in equation 2.28. These matrices are not only dense for higher order basis sets, but also different for different surfaces of the tetrahedron. In the present method, the surface integrals are easily calculated by a product of the rows of the mass matrix in  $d - 1$  dimensions and the nodal values corresponding to the surface, as explained in section 2.4.3. There is no need to even calculate the transformation matrices. Hence the present method is expected to be more efficient for calculations in  $3d$ .

- Due to the introduction of the transformation matrices and additional calculations for the flux approximations, the structure of the original QF code is more complex compared to that of the present method. Details about the structure of the code using the original QF method can be found in the study by Lockard and Atkins [19]. Also, using the original QF method, there is a need for additional pre and postprocessing. The initial condition and the solution in any physical problem are expressed in terms of their values at specific physical points, or, the nodes of an element. Using the moment basis, the coefficients of expansion need to be computed from the nodal values. But using Lagrange functions, the nodal values can be used directly.

To be of practical value, it is important to develop the time domain DG algorithm to run on parallel computing platforms. In the following chapter, a parallel implementation of the DG method for unstructured two dimensional grids is presented.

# Chapter 4

## Parallel implementation

As discussed in Chapter 1, the primary requirements of CAA in aerospace engineering are high order of accuracy, the simulation of noise propagation over large distances and times, and applicability in complex geometries. To be of practical value, the usually massive computations must be performed quickly, and to do so, an efficient use of parallel computing platforms is necessary.

The requirement of a high order simulation in complex geometries complicates the design of a parallel implementation. Traditional high order finite difference methods are not compact, and the amount of data to be communicated across partition boundaries increases considerably with the order of the method [42]. Further, the application of high order finite difference methods on unstructured meshes for complex geometries is extremely difficult. The DG method, however, satisfies the requirement of high order accuracy and applicability to complex geometries. As noted previously, in the DG method, each element behaves as a separate entity, and the element communication involves only the trace of the flux at the surfaces/edges of the element. Hence the method is compact. This compactness allows for an easier implementation of parallel algorithms on unstructured grids. In the present work, a parallel version of the time domain solver using the

DG method is developed using domain decomposition. A detailed description of the parallelization approach is presented here. This is followed by performance results on a Beowulf cluster.

## **4.1 Existing literature on parallel-DG implementation.**

Several investigators have studied the parallel implementation of the DG method. Biswas *et al.* [43] applied a third order quadrature based DG method to the scalar wave equation on 256 processors. Bey *et al.* [44] applied a parallel  $h-p$  adaptive DG method on structured grids for hyperbolic conservation laws and obtained nearly optimal speedups. Baggag *et al.* [42] modified an object oriented DG solver with different parallelization strategies using MPI. They reported slightly superlinear speedup on an SGI Origin due to cache effects for large scale problems, using more than 60 processors. cache effects. Further improvements on the parallelization approach of [42] were implemented in [45], to minimize redundant computations on inter-partition boundary elements. The method was tested on various platforms such as SGI Origin, IBM SP2, and clusters of SGI and Sun workstations. Atkins and Lockard [46] implemented a parallel DG method for the aeroacoustic analysis of realistic aircraft configurations. They used fifth order triangular elements on over 100 processors in a study of acoustic scattering near a leading edge slat. Xu *et al.* [47] predicted the noise from the dominant spinning modes from the fan of an aircraft engine using a parallel DG method. They reported a computation time of 10 days on a 32 processor IBM SP4 machine using approximately 22 million nodes in the computation domain, with a sixth order Legendre polynomial basis.

## 4.2 Domain decomposition

The computational space is divided into a number of polygonal partitions, using the package METIS [40], and computation in each partition is assigned to a separate processor. METIS is a family of open-source programs for partitioning unstructured meshes. It attempts to balance the load and minimize the communication among the processors [40]. A parallel version of METIS, known as PARMETIS is also available, but it has not been used in the present work. This is to allow for additional modifications for complexities encountered in adding PML to the physical domain. Instead, a separate preprocessor, discussed further in this chapter, is developed for generating the finite element data for each partition. The partition data from METIS is a text file containing 2 columns, with the first column being a list of the element numbers, and the second, their corresponding partition numbers. A separate package called PMVIS [48] is used to visualize the partition configuration.

## 4.3 The algorithm

Following the same notation as before, the boundary of an element  $\Omega^e$  in a partition is denoted by  $\partial\Omega^e$ . The element edges lying on the partition boundary are denoted by  $\partial\Omega_I$ , and are termed inter-boundary (IB) edges. The communication between the processors involves the trace of the flux terms on these edges. All other element edges, which are not common to different processors, are denoted by  $\partial\Omega_P$ , and called interior edges.

Consider the scalar hyperbolic equation (2.1). The application of the DG method results in the set of equations for each element. This set of equations can be rewritten as:

$$\frac{\partial \mathbf{q}^e}{\partial t} + [\mathbf{M}^e]^{-1} \mathbf{K}^e \mathbf{q}^e + [\mathbf{M}^e]^{-1} \sum_{j=1}^3 \mathbf{B}_{(j)}^e \bar{\mathbf{F}}_{R(j)} = \mathbf{s}^e \quad (4.1)$$

$\mathbf{M}^e, \mathbf{K}^e$ , are defined by equations (2.12) and (2.13), and  $\mathbf{B}_{(j)}^e, j = 1 \dots 3$  are defined by equation (B.13) in Appendix B, and the Riemann flux for edges that are not on the computational boundary given by equation (2.7). The details of the edge flux term in the above equation can be found in Appendix B.

The Riemann flux for the edges on the computational boundary is expressed in terms of the trace of flux from the left side only, with appropriate modifications for boundary conditions. Note that the flux is discontinuous across an element boundary. However, the absolute value of the Riemann flux on an edge shared by two elements is the same for both the elements. Hence, it is not necessary to compute the Riemann flux for every edge of every element. The computation in a processor is carried out as follows.

1. Calculate the traces  $\bar{\mathbf{q}}$  and  $\bar{\bar{\mathbf{F}}}$  for the inter-boundary edges,  $\partial\Omega_I$ .
2. Pack the arrays of  $\bar{\mathbf{q}}$  and  $\bar{\bar{\mathbf{F}}}$ , into a single array. Send parts of the packed array to the corresponding neighbor processors.
3. Calculate  $\bar{\mathbf{q}}, \bar{\bar{\mathbf{F}}}$ , and  $\bar{\mathbf{F}}_R$  for all  $\partial\Omega_P$ .
4. Receive and unpack the data from the neighbors.
5. Calculate  $\bar{\mathbf{F}}_R$  for all  $\partial\Omega_I$ .
6. Compute the edge flux term  $\sum_{j=1}^3 \mathbf{B}_{(j)}^e \bar{\mathbf{F}}_{R(j)}$  in equation (4.1)  $\forall \partial\Omega^e$
7. Compute  $\mathbf{M}^e, \mathbf{K}^e$  and  $\mathbf{s}^e \forall \Omega^e$ . Advance in time.

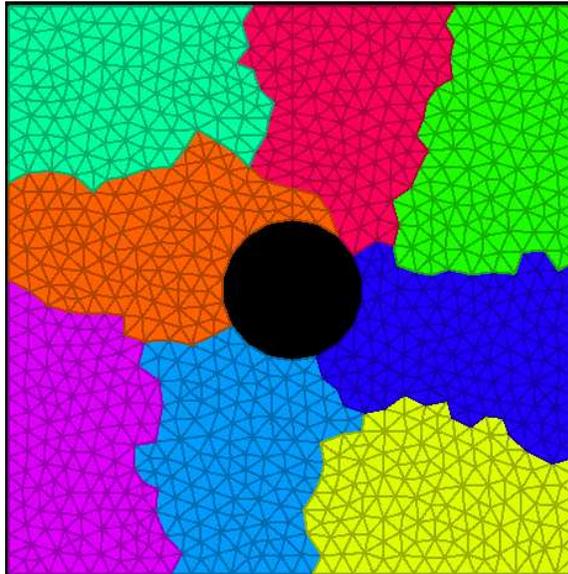
The parallel implementation is MPI [49] based. Non blocking send and receive are used, so that the communication is overlapped with computation in step 3. The period of overlap can be increased by breaking up the process in step 6 between  $\partial\Omega_P$  and  $\partial\Omega_I$  and computing the edge flux terms for  $\partial\Omega_P$  between non blocking send and receive. Further overlap can be achieved by breaking up the calculation of  $\mathbf{M}^e, \mathbf{K}^e$ , in step 7, between the

elements completely inside the domain,  $\Omega_P$  and the elements on the partition boundary,  $\Omega_I$ . The Riemann flux is computed only once for all  $\partial\Omega^e \in \partial\Omega_P$ . For element edges  $\partial\Omega^e \in \partial\Omega_I$ , it is computed once on each partition. However, the calculation of  $\bar{\mathbf{F}}_R$  on all edges in the partition consumes only 2% – 3% of the total time, as noted by Baggag *et al.* [45]. Several other strategies for parallel implementation of the DG method in the time domain can be found in the same reference. However, the complexity of the implementation increases with improvements in the algorithm.

#### 4.4 Pre-processor for parallel implementation

Every processor assigned to a partition must contain information about the number of neighbor partitions, corresponding elements and edges to communicate the data across the inter-partition boundaries. The partitions obtained using METIS are unstructured, meaning that the number of neighbor partitions of a partition under consideration is not constant for all the partitions. This can be seen in a sample unstructured triangular mesh that is used in the study, in Fig. 4.1. This is an important condition to be dealt with when developing the preprocessor. Further, the elements within each partition are numbered randomly. These features are depicted in Fig. 4.2. The common boundary between partitions ‘A’ and ‘B’ is shared by six elements in each of the partitions. These six elements themselves are numbered randomly. The communication across the inter partition boundary requires information about the exact arrangement of the 12 elements, that is, which element in partition A shares what side of which element in partition ‘B’. Such an arrangement information is required for all the neighbor partitions of a partition under consideration. The tasks of the preprocessor can be summarized as follows:

- Generate the coordinate, connectivity and neighbor arrays for all the nodes and elements inside each partition. In other words, number the nodes and elements

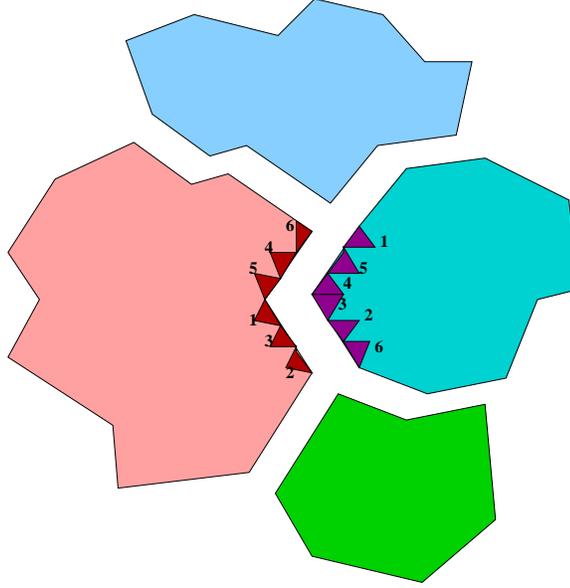


**Figure 4.1.** An example of a partition by METIS.

locally within each partition.

- Extract the number of neighbors and the number of element edges on the common boundary between each of the neighbors for each partition.
- Create the arrangement array for each of the neighbor in each partition. This array contains a list of the local number of elements on the inter-partition boundary, their edge number, the local number of the neighbor element in the neighbor partition, the neighbor element's edge member, and the index of the neighbor element in the neighbor partition's arrangement array. The parts of the arrangement arrays in 'A' and 'B' corresponding to one another are shown in table 4.1 and table 4.2 respectively.

In the table, the edge members on the IB edge for each element in partition 'A' and 'B' are assumed to be 2 and 3 respectively. The column of indices of elements is necessary in unpacking the data communicated across the IB edges.



**Figure 4.2.** A typical partition configuration.

| Element in A | Edge | Partition | Element in B | Edge | Index in B |
|--------------|------|-----------|--------------|------|------------|
| 12           | 2    | B         | 34           | 3    | 3          |
| 27           | 2    | B         | 68           | 3    | 6          |
| 38           | 2    | B         | 23           | 3    | 2          |
| 43           | 2    | B         | 57           | 3    | 5          |
| 52           | 2    | B         | 46           | 3    | 4          |
| 61           | 2    | B         | 10           | 3    | 1          |

**Table 4.1.** The arrangement array for the partition A of figure 4.2

## 4.5 Parallel implementation with PML

The implementation of the PML technique in the calculations introduces an additional variable and damping terms into the system of equations for elements in the buffer domain. The discretized LEE with the unsplit PML is rewritten here:

$$\begin{aligned}
 \frac{\partial \mathbf{q}^e}{\partial t} + [\mathbf{M}^e]^{-1} \mathbf{K}^e \mathbf{q}^e + [\mathbf{M}^e]^{-1} \mathbf{K}^{ez} \mathbf{z}^e + [\mathbf{M}^e]^{-1} \sum_{j=1}^3 \mathbf{B}_{(j)}^e (\bar{\mathbf{F}}_{R(j)} + \bar{\mathbf{F}}_{R(j)}^z) \\
 + (\sigma_x + \sigma_y) \mathbf{q}^e + \sigma_x \sigma_y \mathbf{z}^e + \mathbf{g}^z = \mathbf{s}^e
 \end{aligned} \tag{4.2}$$

| Element<br>in B | Edge | Partition | Element<br>in A | Edge | Index<br>in A |
|-----------------|------|-----------|-----------------|------|---------------|
| 10              | 3    | A         | 61              | 2    | 6             |
| 23              | 3    | A         | 38              | 2    | 3             |
| 34              | 3    | A         | 12              | 2    | 1             |
| 46              | 3    | A         | 52              | 2    | 5             |
| 57              | 3    | A         | 43              | 2    | 4             |
| 68              | 3    | A         | 27              | 2    | 2             |

**Table 4.2.** The arrangement array for the partition B of figure 4.2

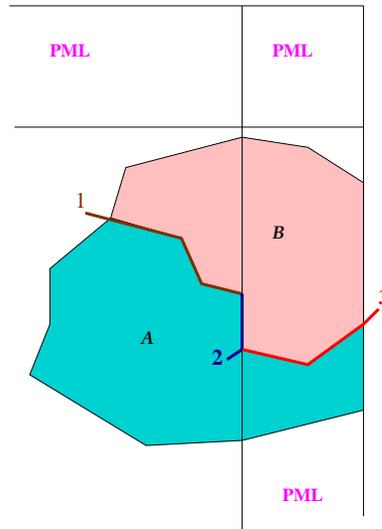
$$\frac{\partial \mathbf{z}^e}{\partial t} = \mathbf{q}^e \quad (4.3)$$

The flux terms from the PML variable have to be communicated across the IB edges of a partition in addition to the flux terms from the acoustic variables. The calculations are carried out as follows:

1. Calculate  $\bar{\mathbf{q}}$  and  $\vec{\mathbf{F}} \forall \partial \Omega_I$  in all partitions.
2. Calculate  $\bar{\mathbf{z}}$  and  $\vec{\mathbf{F}}^z \partial \Omega_I$  in partitions having elements in the PML.
3. Pack the arrays  $\bar{\mathbf{q}}$ ,  $\vec{\mathbf{F}}$ , and send data to neighbors from all partitions.
4. Pack the arrays  $\bar{\mathbf{z}}$ ,  $\vec{\mathbf{F}}^z$  and send to *neighbors with elements in the PML* from partitions having elements in the PML.
5. Calculate  $\bar{\mathbf{q}}$ ,  $\vec{\mathbf{F}}$ , and  $\vec{\mathbf{F}}_R \forall \partial \Omega_P$  in all partitions. Calculate  $\bar{\mathbf{z}}$ ,  $\vec{\mathbf{F}}^z$ , and  $\vec{\mathbf{F}}_R^z \forall \partial \Omega_P$  in partitions having elements in PML.
6. Receive and unpack the data from the neighbors.
7. Calculate  $\vec{\mathbf{F}}_R \forall \partial \Omega_I$ . Calculate  $\vec{\mathbf{F}}_R^z \forall \partial \Omega_I$  in partitions having elements in PML.
8. Compute the edge flux terms  $\forall \partial \Omega^e$

9. Compute  $\mathbf{M}^e$ ,  $\mathbf{K}^e$  and  $\mathbf{s}^e \forall \Omega^e$ . Compute the damping terms for PML. Advance in time.

There are two differences in the algorithm including the PML compared to that without the PML. The first is that the edge flux terms need to be computed for all the elements with the PML present. The second, which is particularly important in the parallel implementation, is that the trace of the PML flux terms,  $\overline{\mathbf{F}}^z$ , on the IB edges needs to be communicated only among partitions with elements in the PML. While some partitions in the computational domain may lie completely outside or inside the PML, others may have elements on both sides of the PML boundary, as illustrated in Fig. 4.3. This com-



**Figure 4.3.** An example of partitions with PML.

plicates the handling of accurate data communication. Three types of computations may be required among partitions that have elements on both sides of the PML boundary:

1. The computation for IB edges not in the PML, such as the IB edge 1 in Fig. 4.3. This involves calculation and communication of only  $\bar{\mathbf{q}}$  and  $\overline{\mathbf{F}}$  across 1.
2. The computation for IB edges coinciding with the PML boundary, such as the IB

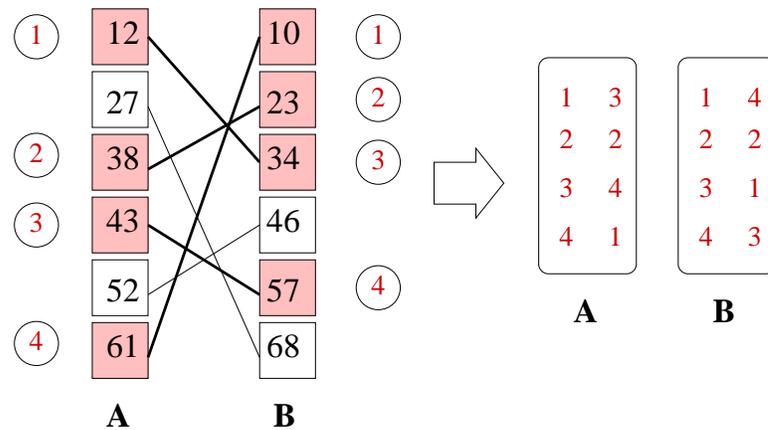
edge 2. This involves the communication of only  $\bar{q}$  and  $\vec{F}$  across 2. The traces  $\bar{z}$  and  $\vec{F}^z$  are calculated only for elements on the side of partition 'B'. The elements in partition 'A' lying on the IB edge 2 do not have a PML variable, and hence  $\bar{z}$  and  $\vec{F}^z$  are neither calculated nor communicated by the partition 'A'. The Riemann flux for the PML variable in partition 'B' on element edges coinciding with the IB edge 2 is one sided. Thus even if some of the elements lying on the IB edge in partition 'B' are in the PML, no communication of the PML flux is necessary for those elements.

3. The computation for IB edges inside PML, such as the IB edge 3. The traces from both the acoustic and PML variables are calculated and communicated across the IB edge.

Use of one or several processors only for the PML would indeed decrease the complications in the computation mentioned above. However, this is deemed to be inefficient for two reasons. Since the domain decomposition is accomplished using METIS, which tries to minimize the inter-processor communications, the addition of partitions in the PML could make the communications unequal and sub-optimal among the processors. Also, the PML calculations will consume additional computation time only in the PML partitions. This would mean a waiting period for processors in the physical domain, as the computations among the processors have to be synchronized. To implement the three cases of computations with PML, additional information is extracted from the pre-processor about the number of neighbors, number of element edges and the arrangement of elements in the neighbors which share the elements in the PML.

### 4.5.1 Preprocessing for PML

The information from the preprocessor for the partitions in Fig. 4.2 is shown in Fig. 4.4. The elements that communicate are connected by lines at the left. Let us assume that the



**Figure 4.4.** Preprocessing for PML partitions.

elements in the shaded boxes are the elements in the PML. These elements are connected by darker lines. Now, within each such arrangement between two processors, the PML-elements are numbered separately, as shown by the encircled numbers. Then the packing for the PML flux communication is accomplished using the PML arrangement array shown in the box at the right for each processor, corresponding to the neighbor. A summary of the steps taken in extracting the data needed for communication in partitions with elements in the PML is given below.

- Extract the ranks of the neighbor partitions with which the PML communication has to be carried out.
- Number the elements which need PML communication, that is, create the encircled list in Fig. 4.4.
- Exchange the list of encircled numbers between the neighbors.

- Generate the arrangement array for the elements in the PML.

## 4.6 Application issues for real flow problems

For real flow problems, such as the noise prediction from the jet engines, the mean flow data that is used in this code is obtained from CFD. The CFD data is more likely to be on a very fine grid, in order to resolve the high gradients in the shear layers and boundary layers. The size of the time step for higher orders  $p \geq 3$  has to be usually an order of magnitude less compared to the minimum element size in the domain, using the TVDRK schemes. As an example, if the CFD grid has a minimum grid size of  $\Delta x = 10^{-4}m$ , the acoustic code can have a maximum time step size of the order of  $\Delta t \approx 10^{-5}s$ . Hence, it is not practical to use the same grid for the time domain acoustic propagation code. This is further elaborated using a practical example in section 4.7. As a consequence, it is required to generate a separate coarser grid for acoustic propagation calculations, and the mean flow data needs to be interpolated onto the new grid. For simple problems, this interpolation could be accomplished using a commercial software such as Tecplot. For problems with a highly complex geometry, or when an already partitioned mean flow grid itself is to be used, a separate algorithm needs to be developed to interface the present solver with the output from the mean flow solver.

The DG code developed in this thesis can be used for the calculation of the propagation of noise from a single source in the shear layer at any observer location. In order to calculate the noise from all the sources in the flow at a particular frequency, a source can be located at the observer's location, and the acoustic reciprocity theorem could be used.

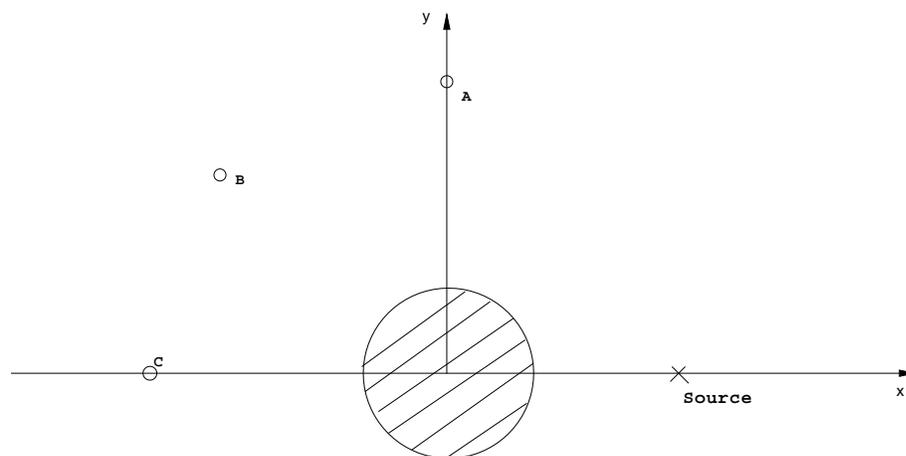
## 4.7 Results from parallel implementation

In this section the parallel DG solver is applied to two problems. The first is a benchmark problem modeling the acoustic scattering by the fuselage of an aircraft. The second is the acoustic propagation in the shear layer of the exhaust from a realistic aircraft jet engine. Results from the performance study of the parallel solver are presented.

The problems considered in this study are solved using a uniformly same order approximation through out the computational domain. The parallel solver developed in the study is not applicable to two domains or elements with different orders of approximation. Such elements or domains are non-conforming, as explained in Chapter 1. The calculation of flux over such elements require additional modifications and additions to the existing basic solver.

### 4.7.1 Acoustic scattering from the fuselage of an aircraft

The physical problem is to find the sound field generated by a propeller, scattered by the fuselage of an aircraft, as stated in the 2nd CAA workshop [50]. The fuselage is ideal-



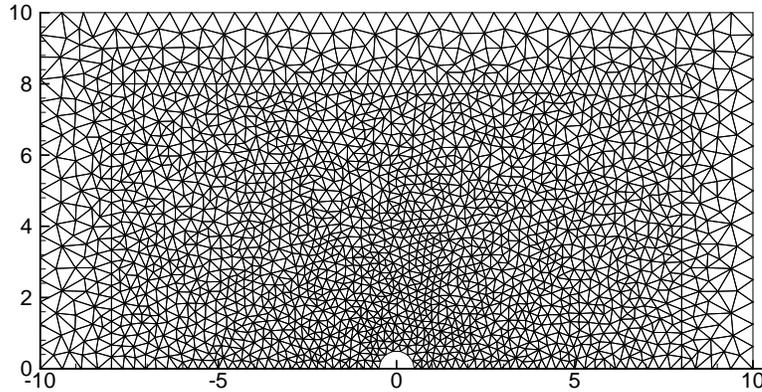
**Figure 4.5.** Acoustic scattering by the fuselage - problem configuration.

ized as a circular cylinder, and the propeller as a line source or an initial Gaussian pulse, so that the computation is in two dimensions, as shown in Fig. 4.5. The diameter of the cylinder is unity, and the source is located at  $(x, y) = (4, 0)$ . The initial value problem, with a zero mean flow and a Gaussian pulse at the source location is considered. The initial conditions are:

$$p = \exp \left[ -\ln(2) \left( \frac{(x-2)^2 + y^2}{0.2^2} \right) \right] \quad (4.4)$$

$$u = v = \rho = 0$$

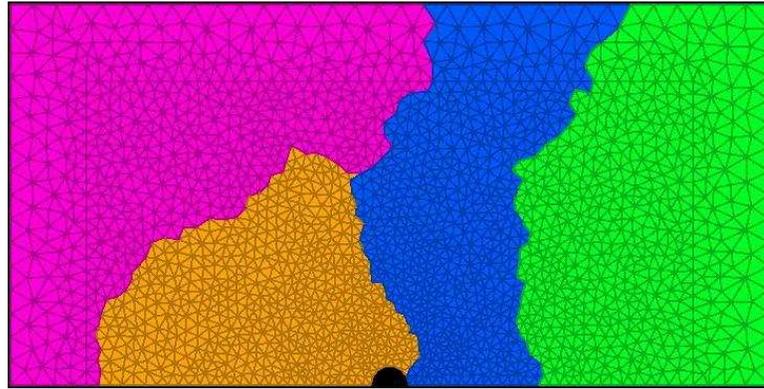
The measurement points are located at a radius  $r = 5.0$  and angles of 90, 135 and 180 degrees from the  $+x$  direction, as shown in Fig. 4.5. The computational domain is as shown in the Fig. 4.6. The unstructured mesh is generated using Gmsh [51], which is



**Figure 4.6.** Acoustic scattering by the fuselage - the mesh.

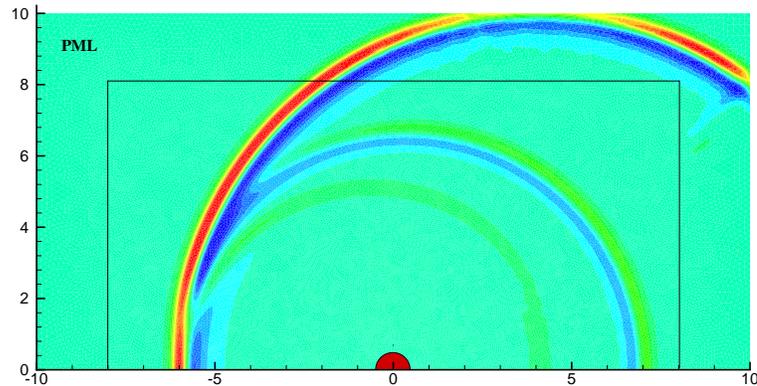
an excellent open source software for generating unstructured meshes. The triangulation consists of 4054 elements and 2100 nodes, with the cylinder being modeled by 8 nodes. The mesh is refined in the region of the cylinder to represent the curved wall. The computational domain is taken to be  $-10 < x < 10$  and  $0 < y < 10$ , so that any residual effects of reflection after damping in the PML are negligible at the measurement points

within the time span of interest,  $t \leq 10$ . The origin is at the center of the cylinder. The PML occupies a thickness of 2 units at the edge of the domain. Symmetry boundary condition is applied at the bottom boundary. The smallest characteristic length of an element in the domain is  $h \approx O(10^{-2})$ . The domain decomposition obtained by METIS on the computational domain for four partitions is shown in Fig. 4.7. The calculations



**Figure 4.7.** Acoustic scattering - the partitions from METIS.

are performed on a Beowulf cluster with Pentium III 800MHz processors. The pressure contour after  $t = 10$  is shown in Fig. 4.8. The initial pulse generates two reflections from the surface of the cylinder. The reflections from the computational boundary are effectively damped in the PML. The degree of the basis set used is  $p = 3$ . A time step of  $\Delta t = 0.0025$  is used with a third order three stage TVBRK method. Fig. 4.9 shows a comparison of the pressure history at the measurement points for  $6 \leq t \leq 10$ . The exact value of the pressure was obtained from Kurbatskii [52], Apart from the error inherent in the DG approximation, the error in the calculation is due to several other factors: the accuracy in the initial condition, the accuracy of the non reflecting boundary conditions and the post-processing error. The initial condition is represented by the grid at the source location,  $(x,y) = (4,0)$ . The non reflective boundary conditions are effective due to the presence of PML. The post processing involves interpolation of the acoustic



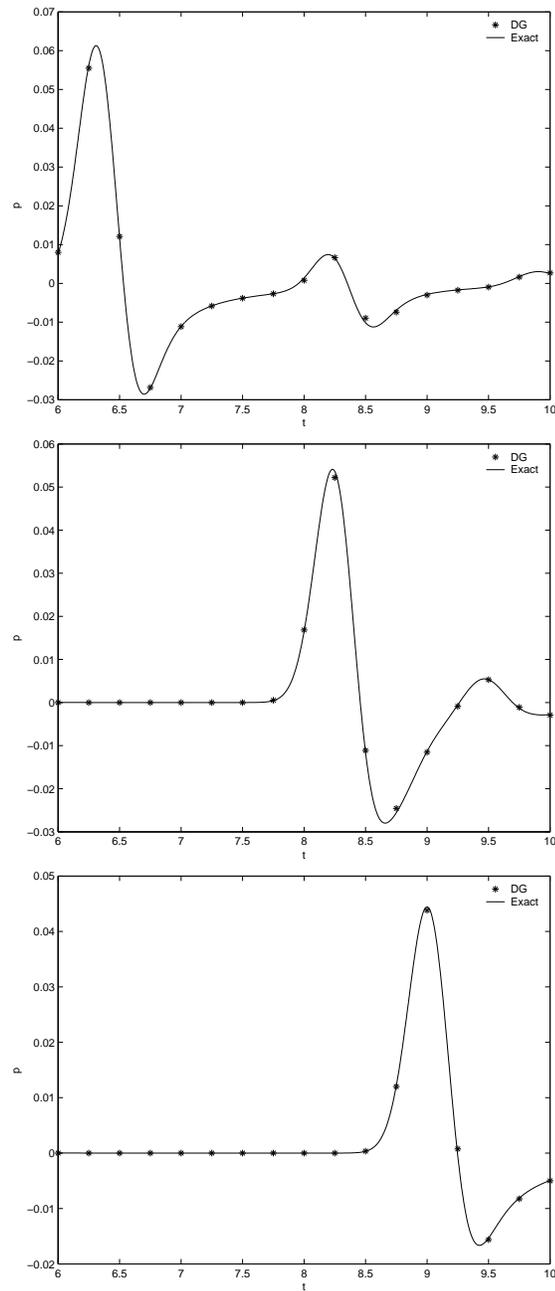
**Figure 4.8.** Pressure contour at  $t = 10$ .

pressure values at the three locations, which was carried out using Tecplot [41].

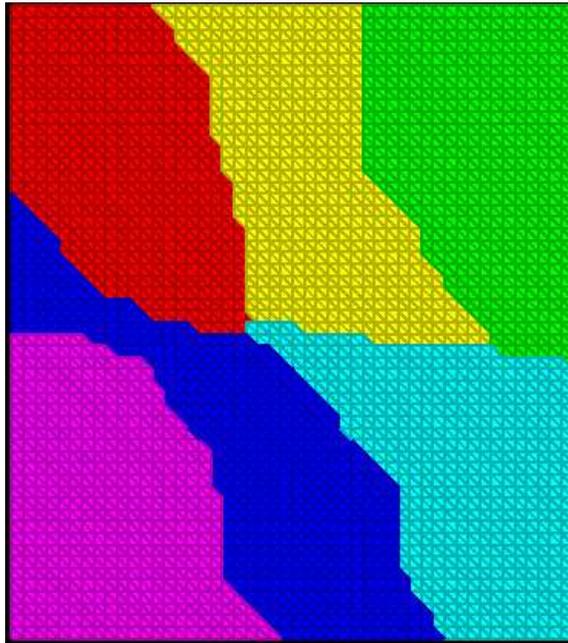
### 4.7.2 Parallel performance

The parallel DG solver is applied to the problem of propagation of an initial pulse in a uniform mean flow [34], which was described previously in Chapter 3. The computational domain is  $-108 \leq x \leq 8$  and  $0 \leq y \leq 196$ . The PML boundary is:  $x < -88$ ,  $x > 88$  and  $y > 172$ . The grid is made of similar right angle triangles of height  $h = 4$  units. There are 5184 elements and 2695 nodes in the domain. The performance of the parallel solver is tested by varying the number of partitions from 1 to 32. A typical partition from METIS is shown in Fig. 4.10. The calculations are performed on COCOA2, a Beowulf cluster in the Department of Aerospace Engineering, at Penn State. The cluster has 21 nodes with 2 Pentium III 800 MHz processor at each node. It has a dual fast-Ethernet per node and all the nodes are connected using two HP2524 switches with channel bonding.

The calculations are performed using a second order basis set, and marching in time up to  $t = 200$  with  $\Delta t = 0.1$ . The parallel speedup in computation is shown in Fig. figspeedup. The parallel speedup is calculated as the ratio of the computation time



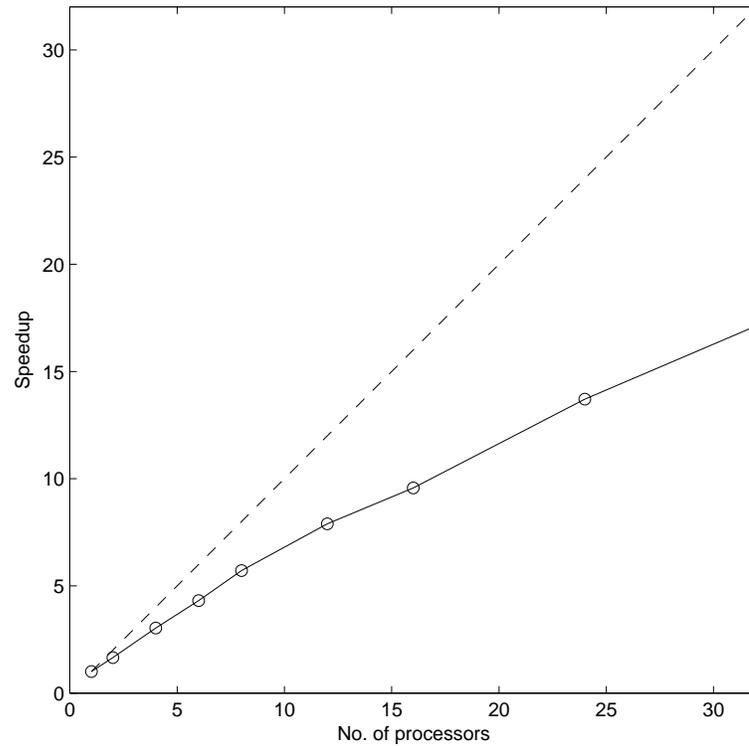
**Figure 4.9.** Pressure history at points A(top), B(center), and C(bottom).



**Figure 4.10.** A typical partition used for the parallel performance study.

consumed using a single processor with a serial algorithm to the time using a parallel algorithm for a particular number of processors. The parallel efficiency, defined as the percentage ratio of the speedup to the number of processors is shown in Fig. 4.12. The speedup and efficiency deteriorate with the increase in the number of processors. An explanation for this behavior lies in the load balance and the communication overlap period. As the number of processors is increased, the number of elements within a processor decreases, and at the same time, the number of elements on the IB edges varies. The latter may increase or decrease, depending on the configuration and the number of neighbor partitions for a given partition. However, too many partitions in a domain involves a substantial communication time. This implies a waiting period for each time step, during which only communications occur, hence a deterioration in the efficiency.

The scalability of the solver is tested by increasing the problem size with the increase in the number of processors. The scaled speed up is shown in table 4.3. The number



**Figure 4.11.** Parallel speedup for the pulse propagation problem.

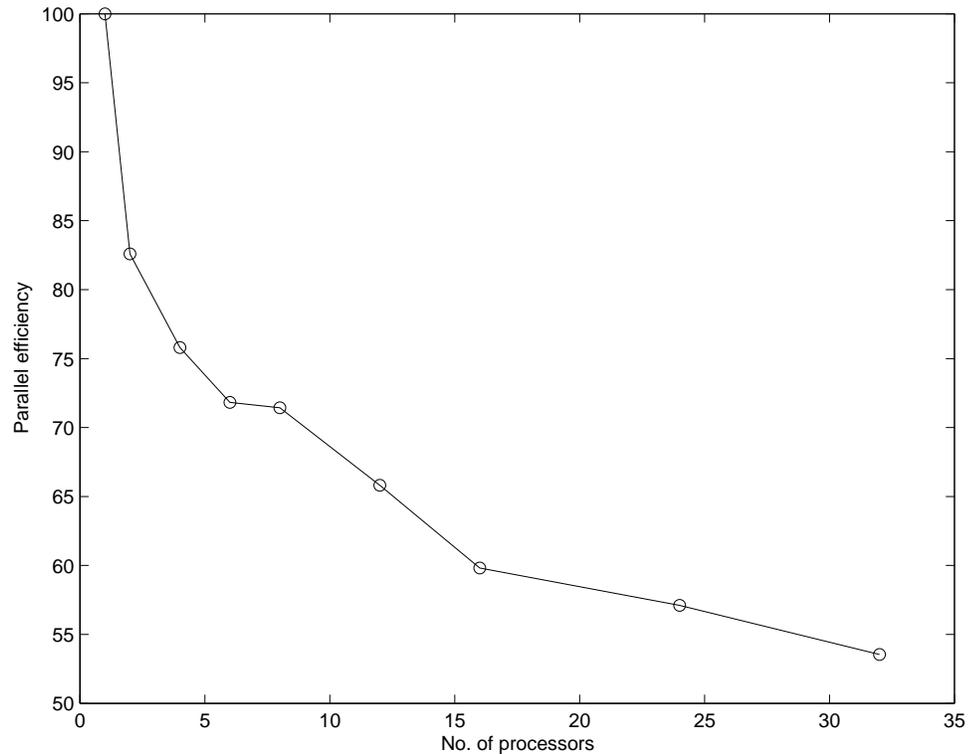
of elements is increased in the same ratio as that of the number of processors. A scaled speed up of upto 88% is observed for the case with 16 processors.

| Number of Processors | $N_{el}$ | Computation time (s) | Scaled speed up |
|----------------------|----------|----------------------|-----------------|
| 1                    | 1296     | 988.7                | 100             |
| 2                    | 2392     | 1047.6               | 94.4            |
| 4                    | 5184     | 1052.1               | 94.0            |
| 8                    | 10368    | 1085.6               | 91.1            |
| 16                   | 20736    | 1122.2               | 88.1            |

**Table 4.3.** Scaled speed up for the parallel solver.

### 4.7.3 Acoustic propagation in the shear layer of a realistic jet

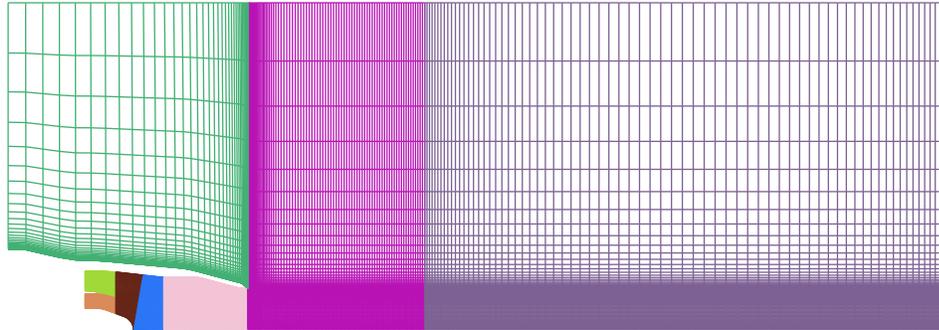
The configuration of the selected jet nozzle is similar to a Rolls Royce Allison test nozzle, as shown in Fig. 1.2, in two dimensions. This may be imagined as a nozzle between



**Figure 4.12.** Parallel efficiency for the pulse propagation problem.

two curved plates, with a splitter plate in the middle. The configuration is not however axysymmetric, since the DG algorithm developed in this thesis is for rectangular cartesian grids only. Hence the solution obtained could be considered as the simulations for a rectangular nozzle. The mean flow solution is obtained from CFD, using a finite difference method. The details of the method used in the obtaining the mean flow solution and the turbulence model used can be found in the Master's thesis of Steve Zygmunt [53].

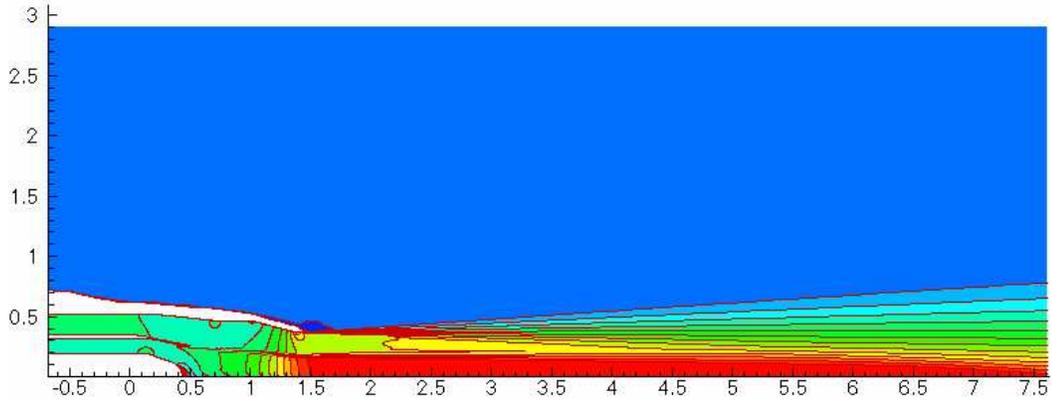
The structured grid used in the mean flow solution is shown in Fig. 4.13. The mean flow solution is obtained using a parallel solver, with 8 partitions, as shown in the figure. A contour plot of the mean flow velocity in the  $x$  direction is shown in Fig. 4.14. The diameter of the nozzle at the exit is  $D = 0.762m$ . The maximum mean flow Mach number based on local speed of sound in the jet is 0.7. The free stream Mach number



**Figure 4.13.** The structured grid used in the mean flow solution. Courtesy Steve Zygmunt [53].

is 0.2. The Reynolds number in the free stream is  $4.375 \times 10^7$ . The length of the computational domain is  $10D$  in the  $x$  direction, and  $3.9D$  in the  $y$  direction. The structured grid has close to 53000 grid points, and 103000 elements when triangulated using the same points. It is clustered in the vicinity of the solid boundaries, as well as the shear layer, to resolve the sharp gradients in the flow variables. As evident from Fig. 4.13, the clustering extends beyond the areas of sharp gradients near the walls of the jet well into the far field. The minimum characteristic length of the grid is  $h_{min} \approx 5 \times 10^{-5}m$ , near the tip of the nozzle. Clearly, this resolution is excessive when extended to the far field, as seen in the figure directly above the tip of the nozzle exit, at the top boundary.

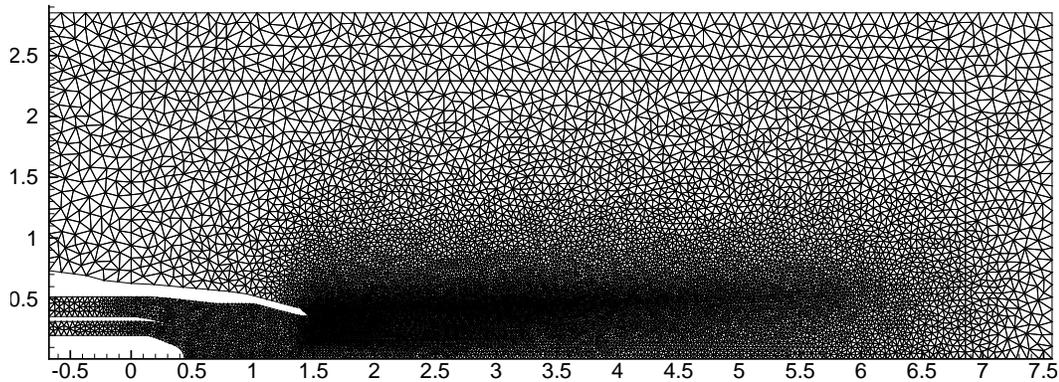
Using a third order time marching scheme for acoustic propagation with the DG



**Figure 4.14.** Contour plot of the mean flow velocity in the  $x$  direction.

method, the time step needs to be at least one order of magnitude less than the minimum spatial dimension. This implies that the time step needs to be less than  $\Delta t_{max} \approx 1 \times 10^{-6}s$ . Assuming a non dimensional frequency of  $St = 0.5$ , this translates to marching through approximately  $0.5 \times 10^6$  time steps for a single cycle of the oscillation. The acoustic propagation has to be simulated to the far field, which will require several ( $\sim O(10)$ ) periods of oscillations, or several million time steps using the structured grid. If the same grid were to be used with the DG method, the calculations for over 100,000 elements would require several days.

The excessive resolution in the boundary layers and shear layer regions of the flow are not usually critical in a computational aeroacoustic calculation. Hence it is imperative to generate and interpolate the flow variables to a separate grid for the CAA calculations. Figure 4.15 shows an unstructured triangulation on the same computational domain, obtained using Gmsh. This grid has approximately 13,000 nodes and 25,000 elements, with the minimum characteristic element dimension  $h_{min} \approx 1 \times 10^{-2}m$ . The coordinates of the boundary points of the solid walls of the nozzle are extracted and plotted in Gmsh, and the resulting plot is triangulated, as shown in Fig. 4.15. The coordinates of the shear layer boundaries in Fig. 4.14 are extracted, and are used to cluster



**Figure 4.15.** The unstructured grid for the splitter plate nozzle problem.

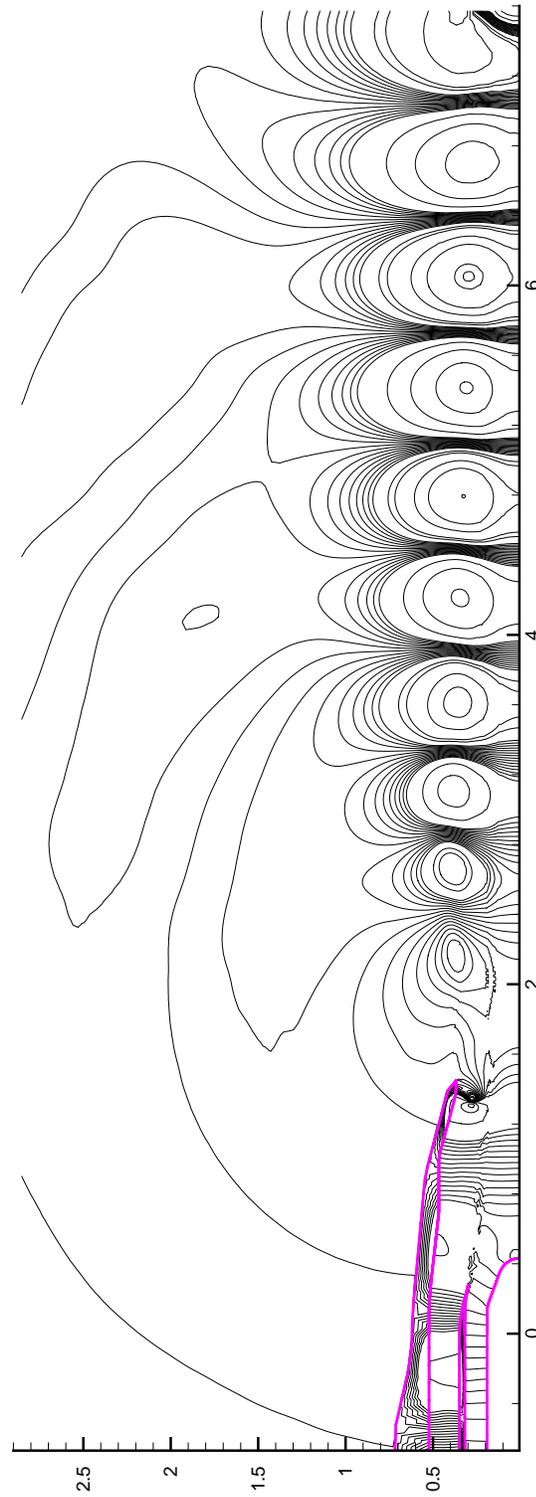
the unstructured grid in the shear layer. The mean flow variables are interpolated onto the unstructured grid using Tecplot [41]. The interpolation introduces errors in the mean flow values that are used for the acoustic calculations. In this study, the interpolation is linear. Linear interpolation finds the values in the destination points based on their location within the elements of the source mesh. The value is linearly interpolated to the destination data points using only the data points at the vertices of the element in the source mesh [41]. Other interpolation methods within Tecplot include the inverse and the kriging algorithm, which use a weighting function to reduce the error in the interpolation. Further details about the interpolation algorithms could be obtained from Tecplot's user manual [41].

The mean flow variables interpolated for the aeroacoustic computation are the flow density, velocity, pressure and their spatial derivatives at the nodes. The thickness of the PML is  $0.9D$  in the  $x$  direction at each end, and  $0.8D$  at the top. The original structured grid in Fig. 4.13 near the entrance to the nozzle is extended upstream to provide for the PML at the inlet. The acoustic propagation from a time harmonic pressure source embedded in the shear layer of the jet is simulated using the parallel DG solver. The

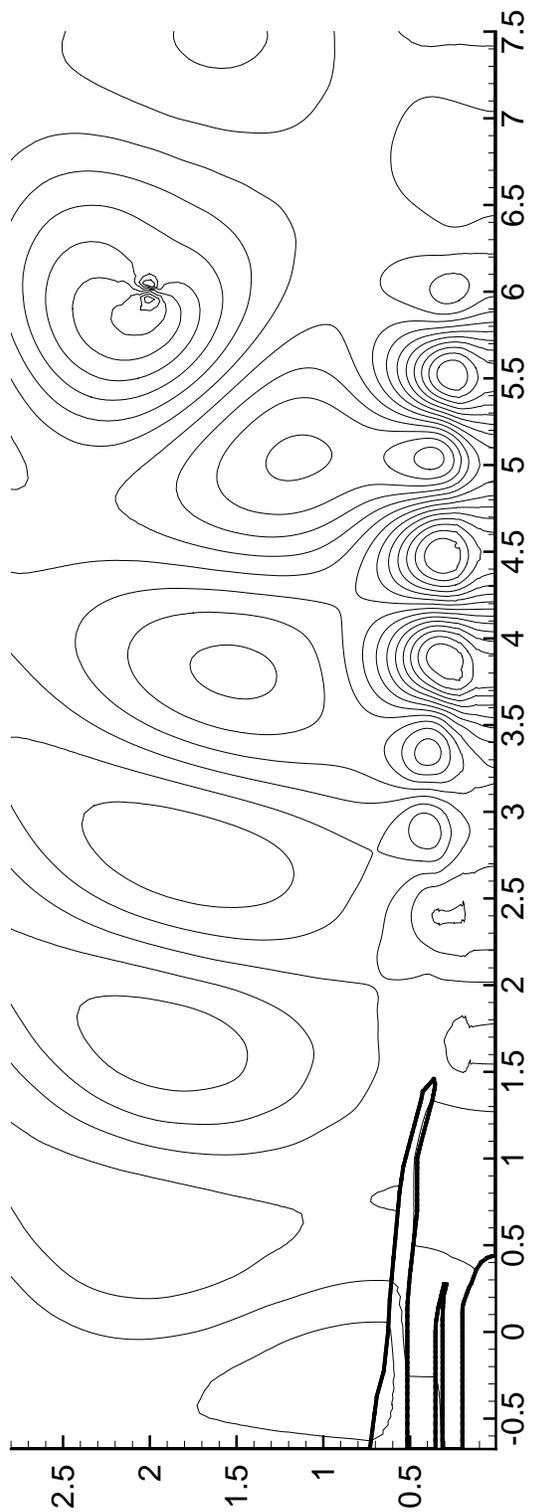
source location is at  $(x,y) = (1.45, 0.262)$ , near the exit of the nozzle. The origin of the coordinates is at the lower left corner of Fig. 4.15. The source frequency is chosen to be  $St = fD/U_c = 0.33$  where  $U_c$  is the jet inlet speed of sound,  $U_c = 437.9m/s$ . The characteristic variables for non-dimensionalization are chosen to be  $U_c$ ,  $D$ , and the free stream density. The source frequency based on  $St = 0.33$  is  $f = 189.6Hz$ . The source is a periodically oscillating Gaussian distribution, with an amplitude of  $0.01N/m^2$ , and a width of  $1\text{ cm}$ . A symmetry boundary condition is applied at the bottom boundary, and a combination of PML with radiating boundary conditions is used on the other three boundaries. The time step for the computation is chosen as  $\Delta t = 1 \times 10^{-3}$  non-dimensional units, with a third order basis set.

Figure 4.16 shows a contour plot of the pressure after a time of  $t = 6T_s$  where  $T_s$  is the non-dimensional time period of the source. The computation is performed on the Beowulf cluster ‘Lionxe’, of the Pennsylvania State University, with 32 processors running at 1GHz for approximately 7 hours. Incidentally, the source produced an instability wave which is evident in the contour plot. The reflections of the acoustic pressure waves are seen to travel upstream of the nozzle in the external flow field. Figure 4.17 shows a contour plot of the pressure after a time of  $t = 6T$  for the same source placed in the far field, at  $(x,y) = (6,2) = (7.874D, 2.624D)$ . The sound waves propagate upstream inside the nozzle, indicating that the sources inside the nozzle contribute to the noise at the source location. The experimental results of the noise from this nozzle configuration involve the data at several observer locations, from all the sources in the flow, at various frequencies. This information can only be gathered by processing the output from the DG solver further, using the acoustic reciprocity principle. These additional calculations are have not been performed in the present thesis.

The next chapter deals with the evaluation and development of both discontinuous and continuous Galerkin methods for solution of the LEE in the frequency domain.



**Figure 4.16.** Contour plot of the acoustic pressure at the end of a cycle of oscillation of a source inside the nozzle.



**Figure 4.17.** Contour plot of the acoustic pressure at the end of a cycle of oscillation of a source in the far field.

# Chapter 5

## Frequency domain calculations

In this chapter, the application of finite element methods in the frequency domain is considered. As mentioned previously in Chapter 1, the linearized Euler equations support instability wave solutions for certain source frequencies. These instability wave solutions are only mathematical in nature, but are absent in real flows. Various techniques have been used by several researchers to suppress the instability wave solution in the time domain solution [6]. These techniques range from assuming zero gradient in the mean flow to numerical filtering of the instability waves. One of the approaches that suppresses the instability wave solution is to solve the LEE in the frequency domain [6]. The solution of LEE in the frequency domain also saves substantial computational time compared to time domain calculations. In the frequency domain, a harmonic response is assumed for the acoustic variables. The frequency of response is chosen as the frequency of oscillation of the source. The resulting set of equations is elliptic in nature. Since the boundary conditions affect the solution at every point inside the computational domain for an elliptic system, the algebraic system of equations obtained after numerical discretization needs to be assembled globally.

The use of iterative techniques to solve the resulting global matrix has been dis-

cussed by Agarwal [54]. It is proved that the use of any iterative technique to solve the global matrix is equivalent to a pseudo-time marching method, and hence, produces an instability wave solution. Hence, the solution of the global matrix needs to be sought by using direct methods such as Gaussian elimination or LU decomposition techniques.

In the following sections, a brief review of the literature on the development and application of the DG method to elliptic problems is provided. This is followed by the development of an algorithm using the DG method for frequency domain problems. The algorithm is tested using examples in one and two dimensions. Further, a comparison of the DG method with the continuous Galerkin method in terms of the computational expense is carried out. Finally, a stabilized continuous Galerkin method is used to seek the suppression of the instability wave solution.

## **5.1 Literature on the DG method for elliptic equations**

The DG methods were originally developed for applications to convection dominated hyperbolic problems. Some early work on the extension of the DG methods to elliptic problems was carried out by Cockburn and Shu [55], in their development of the local DG method for convection-diffusion problems. The mathematical development of the DG method for elliptic problems was addressed by Oden *et al.* [56]. An *a priori* error estimate, convergence proofs and stability estimates were developed. An exponential convergence rate was proved for simple problems in one and two dimensions. Later, a comprehensive discussion of the literature on the DG methods applied to elliptic equations was provided by Arnold *et al.* [57]. A unified DG approach to elliptic systems was also developed. However, there have been relatively few discussions on the application aspects of DG method for elliptic problems. Hesthaven and Warburton [58] applied the elliptic DG method to problems in electromagnetics, by solving Maxwell's equations in

the frequency domain.

Several researchers have attempted to solve the linearized Euler equations in the frequency domain [6]. However, in all such implementations, an assumption has been made regarding the mean flow gradients, or the equations themselves have been altered so as to obtain a Green's function for the resulting set of equations. For example, the assumption of no gradients in the direction of the mean flow allows the reduction of the LEE to Lilley's equation, for which an analytical solution can be found. A frequency domain solution of the LEE for the acoustic propagation and scattering of jet noise around an aircraft was sought by Xu *et al.* [47]. They did not reduce the LEE to an equation with an analytical solution. However, the mean flow around the aircraft was assumed to be irrotational, so that the momentum equation reduced to an algebraic relation between the density to the velocity potential. The development of a frequency domain solver for the case of a general mean flow configuration is still an issue which needs much attention. The primary goal of the present work is the initial development and demonstration of a finite element frequency domain solver of the full LEE, for a general mean flow configuration.

## 5.2 DG method for the LEE in the frequency domain

Consider the LEE in two dimensions given by equation (3.1). In the frequency domain approach, a time harmonic response is assumed for the acoustic variables of the form

$$\mathbf{q} = \tilde{\mathbf{q}}e^{-i\omega t}, \quad (5.1)$$

with the frequency of response  $\omega$  equal to the frequency of oscillation of the source. Under this assumption, the linearized Euler equations reduce to

$$-i\omega\tilde{\mathbf{q}} + \frac{\partial \mathbf{A}_r \tilde{\mathbf{q}}}{\partial x_r} + \mathbf{C}\tilde{\mathbf{q}} = \tilde{\mathbf{s}} \quad \text{for } r = 1, 2, \quad (5.2)$$

with  $\mathbf{A}_r$  and  $\mathbf{C}$  as given in section 3.1. The discretized equation following the application of the DG method to the above equation within an element in the domain is

$$\mathbf{P}^e \tilde{\mathbf{q}}^e + \sum_{j=1}^3 \mathbf{P}^{nej} \tilde{\mathbf{q}}^{nej} = \mathbf{f}^e \quad (5.3)$$

with

$$\mathbf{P}^e = \left[ -i\omega \mathbf{I} \otimes \mathbf{M}^e + \mathbf{A}_r^e \odot \mathbf{K}_r^e + \mathbf{C}^e \odot \mathbf{M}^e \right] + \frac{1}{2} \sum_{j=1}^3 \left[ [\mathbf{A}_r^e \tilde{\mathbf{n}}_r^j] \odot \mathbf{B}^{ej} + \alpha \mathbf{I} \otimes \mathbf{B}^{ej} \right], \quad (5.4)$$

$$\mathbf{P}^{nej} = \frac{1}{2} \sum_{j=1}^3 \left[ [\mathbf{A}_r^{nej} \tilde{\mathbf{n}}_r^j] \odot \mathbf{B}^{ej} - \alpha \mathbf{I} \otimes \mathbf{B}^{ej} \right], \quad (5.5)$$

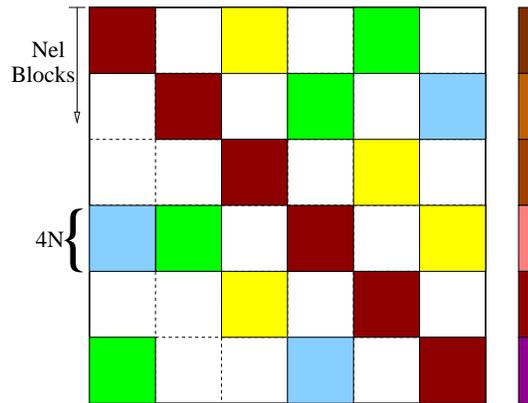
and

$$\mathbf{f}^e = \left[ \mathbf{I} \otimes \mathbf{M}^e \right] \tilde{\mathbf{s}}^e. \quad (5.6)$$

The element matrices and the symbols are as follows.  $\tilde{\mathbf{q}}^e$  and  $\tilde{\mathbf{s}}^e$  are vectors of size  $4N$  made up of the nodal values for all variables in  $\tilde{\mathbf{q}}$ , and the acoustic source values  $\tilde{\mathbf{s}}$  respectively. The superscript  $(\ )^{nej}$  denotes the values from the neighbor element on side  $j$ . The mass matrix  $\mathbf{M}^e$  is given by equation (2.12). The stiffness matrix  $\mathbf{K}_r^e$  is given by the component of the  $\mathbf{K}^e$  in the  $r^{th}$  direction of  $\nabla\psi$  in equation (2.13). The elemental edge matrix  $\mathbf{B}^{ej}$  for a given side  $j$  is given in section B.3.1 in Appendix B. The symbols  $\otimes$  and  $\odot$  are the Kronecker product and the row multiple, which are defined in Appendix C.

The elemental equations need to be assembled to form a global matrix. In the global

assembly, the only connection between the elements is through the elemental boundary matrices. Since the solution in each element is discontinuous, each element has its own set of  $4N$  nodal variables, for the nodal values of the four variables in  $\tilde{\mathbf{q}}^e$ . There are a total of  $4N \cdot N_{el}$  variables in the computational domain. The structure of the global matrix is illustrated in Fig. 5.1. The global matrix is made of smaller block matrices.



**Figure 5.1.** The structure of the global matrix and the vector of variables in the frequency domain DG method.

The numerical equation from each element occupies a set of  $4N$  rows. The diagonal block matrix in a set of rows for an element corresponds to the contribution from the element. There are three more blocks in the same set of rows, from the three neighbors surrounding the element.

### 5.2.1 Boundary conditions

A buffer zone type boundary condition is implemented. The physical domain in a problem is surrounded by a buffer zone of sufficient width. The acoustic amplitudes are damped by adding an imaginary component to the frequency of response only in the buffer zone, similar to the method utilized in Agarwal and Morris [6]. In the buffer

zone,

$$\omega \rightarrow \omega(1 + i\sigma(x, y)). \quad (5.7)$$

The damping function  $\sigma$  is varied smoothly from zero at the beginning to a maximum value at the end of the buffer zone. The values of the acoustic variables are assumed to be zero on the boundary of the computational domain. For a one dimensional problem, the damping function is taken as

$$\sigma(x) = \varepsilon \frac{1 - \exp(\kappa x_b^2)}{1 - \exp(\kappa)}, \quad (5.8)$$

with  $x_b$  being the coordinate in the buffer region normalized by the width of the region.  $\varepsilon$  and  $\kappa$  are taken as positive constants.

### 5.2.2 The direct solver

In the present work, a sparse LU decomposition solver routine from Matlab [59] is utilized to solve algebraic systems of reasonably smaller sizes, of the order of a few tens of thousand rows. For algebraic systems of a larger size, a parallel solver called SuperLU [60] is used. The size of the global matrix is stated from this point in terms the number of rows. SuperLU is an open source sparse matrix LU decomposition solver with MPI based parallel implementation written in C. It has been extensively tested on various parallel platforms. The only constraint with the SuperLU solver is that the sparse matrix must be written in the Harwell-Boeing format. The other alternative is, however, to modify the solver's source code itself, but this is a tedious process, which in turn would consume more time and effort. In the present work, the elements of the global matrix are written out in the coordinate-storage format, and then, a modified Matlab routine from Xiaoye Li [61] is utilized to write out the global matrix in the Harwell-

Boeing format.

## 5.3 Applications

In this section the application of the DG method to a one dimensional wave equation is considered. This is followed by an example of the acoustic propagation in a two dimensional parallel jet flow, with the presence of instability waves in the solution of the LEE. The DG method is then compared with the continuous Galerkin methods in terms of computational requirements.

### 5.3.1 One dimensional calculations

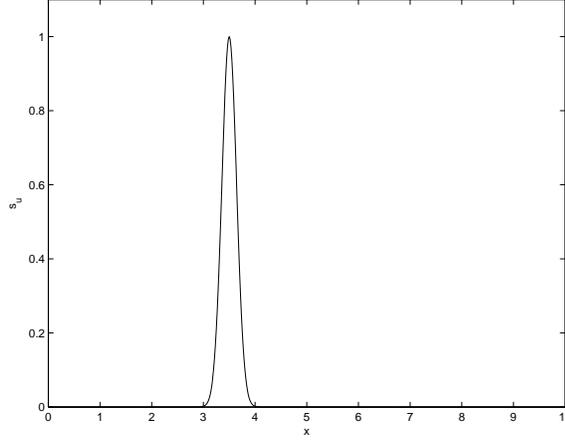
The one dimensional inhomogeneous wave equation in the frequency domain is considered. The wave equation is written as a system of first order differential equations:

$$-i\omega\tilde{\mathbf{q}} + \mathbf{A}\frac{d\tilde{\mathbf{q}}}{dx} = \tilde{\mathbf{s}}, \quad (5.9)$$

where

$$\tilde{\mathbf{q}} = \begin{Bmatrix} \tilde{u} \\ \tilde{p} \end{Bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & a \\ a & 0 \end{bmatrix}, \quad \text{and} \quad \tilde{\mathbf{s}} = \begin{Bmatrix} \tilde{s}_u \\ \tilde{s}_p \end{Bmatrix}. \quad (5.10)$$

The computational domain is  $0 < x < 10$ . The value of the wave propagation velocity,  $a = 4$ , and the source frequency is chosen to be  $\omega = 10$ . The sources are chosen to have a Gaussian variation, with the peak at  $x = 3.5$ , and a variance of 0.2. The variation of the source terms is given in Fig. 5.2. Characteristic boundary conditions are implemented at the boundaries. The exact solution, which is obtained using Green's functions [62] is



**Figure 5.2.** The strength of the source for the 1d wave equation.

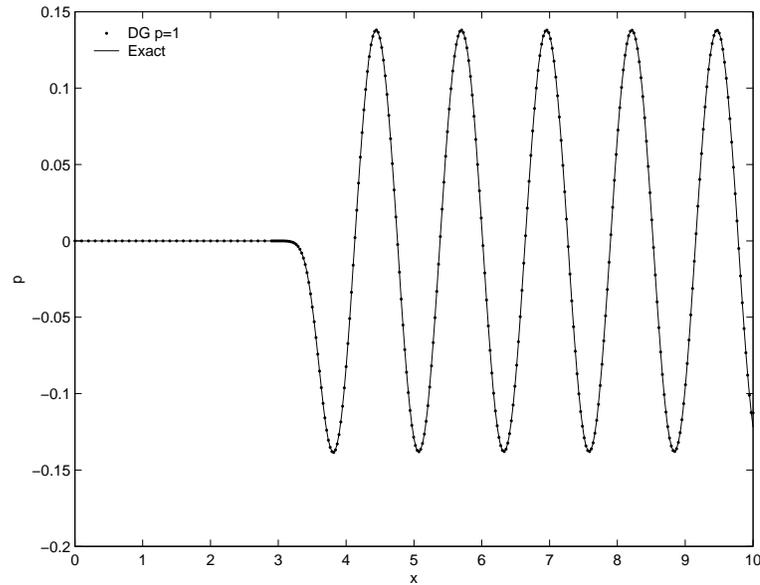
given by:

$$p = -\frac{1}{2ika^2} \int_{-\infty}^{+\infty} e^{ik|x-\zeta|} \left( i\omega + \frac{2a(y-3.5)}{0.2^2} \right) e^{-(\zeta-3.5)^2/0.2^2} d\zeta \quad (5.11)$$

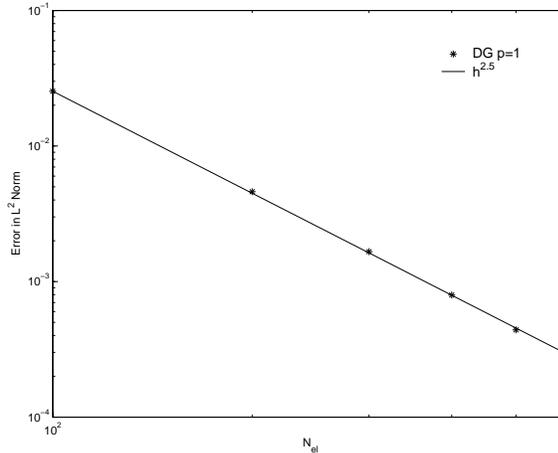
The solution obtained using the DG method with a basis set made up of first order polynomials is compared to the exact solution in Fig. 5.3. There are 300 equisized elements. The elements are clustered in the region of variation of the solution. There are only 10 elements per unit length between  $0 < x < 2.9$ , and there are 38 elements per unit length in the rest of the domain. The  $L^2$  norm of the global error between the DG solution and the exact solution is calculated for varying number of elements. The convergence plot in Fig. 5.4 shows the variation in the error, which decreases as  $h^{2.5}$ .

### 5.3.2 Two dimensional calculations

The benchmark problem from the fourth CAA workshop [63] is chosen as the test problem. The problem concerns the radiation and refraction of acoustic waves generated by a time harmonic source immersed in the shear layer of a jet. The geometry of the prob-



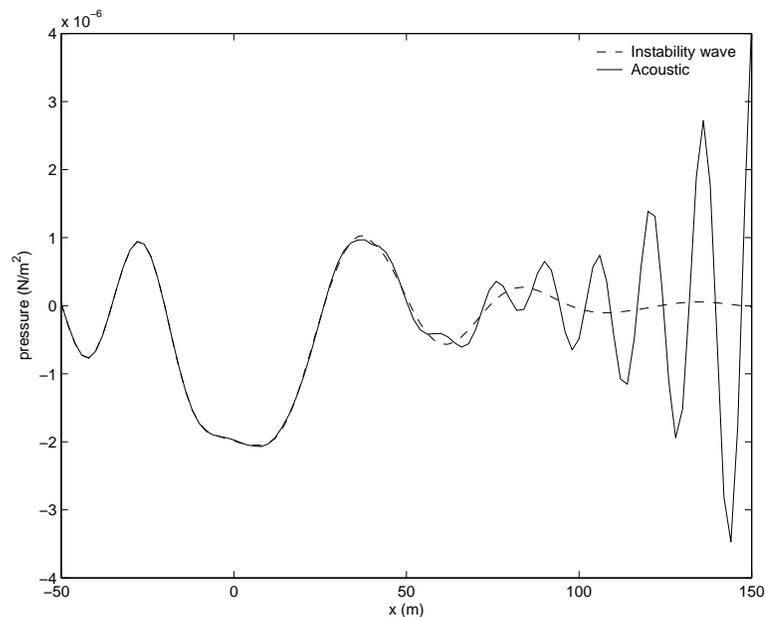
**Figure 5.3.** Comparison of the DG solution with the exact solution for the  $1d$  wave equation in the frequency domain.



**Figure 5.4.** Convergence of the DG solution for the  $1d$  wave equation.

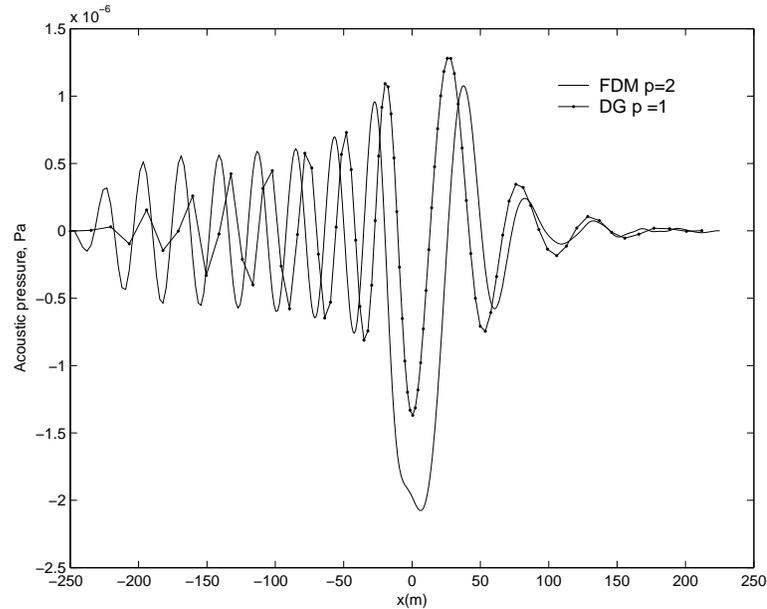
lem domain, and the mean flow profile is similar to that shown in Fig. 3.12. However, the physical domain is  $-50 < x < 150$  and  $0 < y < 50$ . The source is located at the origin. The frequency of excitation of the source is 76 rad/s. The Mach number of the mean flow, which is only in the  $x$  direction, is 0.756, based on the speed of sound in the

jet. The gradient of the mean flow along the  $y$  direction is very high between  $y = 0$  to  $y \approx 4$  compared to the region  $4 < y < 50$ . An instability wave is excited by the source at the specified frequency of the source, but only the acoustic solution is obtained by solving the LEE in the frequency domain. The instability wave and the acoustic solution from Agarwal *et al.* [6] are shown in Fig. 5.5, where the instantaneous pressure values at  $y = 15$  at the end of a cycle of oscillation of the source are plotted.



**Figure 5.5.** The instability wave and the acoustic solution from Agarwal *et al.* [6]

The physical domain is surrounded by a buffer domain, in which the acoustic amplitudes are damped by adding an imaginary component to the frequency. The computational domain is  $-250 < x < 225$  and  $0 < x < 100$ . The mesh consists of similar right angle triangles obtained by halving the rectangles of a  $89 \times 49$  grid. There are 8500 elements in the mesh. The mesh is clustered near the origin in both  $x$  and  $y$  directions to resolve the mean flow gradient and also the source. The result obtained using a first order polynomial basis set is shown in Fig. 5.6. The solution from the DG method is



**Figure 5.6.** Comparison of the DG solution with the exact solution for the two dimensional frequency domain problem.

compared with a solution from a 2nd order finite difference method given by Agarwal *et al.* [6]. The size\* of the global matrix is over  $10^5$ . The solution time using the SuperLU solver on 4 processors is approximately 3.5 minutes. By comparison, a time domain solver used on the same number of elements takes a few hours to obtain a solution at the far field. However, the DG solution needs improvement. There are several areas that could be worked on to improve the solution. The solution depends heavily on the boundary conditions, which could be changed by extending the buffer zone, and changing the damping coefficients, or by using a PML. The order of approximation and the number of grid points could be increased. But, an analysis of the computational needs and a comparison with other finite element methods has shown that such improvements are unnecessary, since the DG method itself proves to be far more expensive for aeroacoustic problems in the frequency domain.

\*The size of a matrix has been previously defined as just the number of rows.

### 5.3.3 Evaluation of the DG method

As mentioned above, the size of the global matrix in the DG method is given by  $4NN_{el}$ . The number of non-zeros ( $N_{nz}$ ) in the matrix can be calculated as follows. In the discretized equation (5.3) for the two dimensional LEE, The global matrix is made of element volume matrices and element edge matrices. The total number of these matrices in  $\mathbf{P}^e$  is equal to the total number of terms in the matrix  $\mathbf{A}_r^e \vec{n}_r^{(j)}$ , that is 10. Each volume matrix is of size  $N$ . For simplicity, the volume matrices will be considered to be full. The number of non zeros in the edge matrices is equal to the number of entries in the mass matrix in one dimension, and is given by  $(p+1)^2$ . Considering the three edges, the total number of non-zeros in an elemental matrix is given by,  $10(3(p+1)^2 + N^2)$ . Hence the total number of non-zeros in the global matrix is given by,

$$N_{nz} = 10(3(p+1)^2 + N^2)N_{el}, \quad (5.12)$$

$$\text{with } N = \frac{(p+1)(p+2)}{2}.$$

Consider a modest 8500 element grid. The size of the global matrix is  $10^5$  and the number of non-zeros is over  $1.75 \times 10^6$ . The size of the Harwell-Boeing binary format file containing the sparse global matrix is 92 MB. The sizes of this matrix are 220 MB and 660 MB, using a uniformly second and third degree basis sets. Thus, the files size grows rapidly, and it becomes increasingly difficult to deal with the computer memory. Evidently, the number of non-zeros increases as the square of the degree of the basis set, and is directly proportional to the number of elements.

## 5.4 Evaluation conclusions

In an unstructured grid, the number of elements is generally twice the number of nodes. Using a conventional continuous Galerkin method, the size of the global matrix is equal to the number of nodes. Hence the global matrix from the DG method is twice the size of that from a continuous FEM. Moreover, the edge flux terms are absent in a continuous Galerkin formulation. Therefore the number of non-zeros is also considerably less compared to the DG method. Considering the same 8500 element grid, using a continuous Galerkin method, which is described in the following section, the number of non-zeros is 4.5 times less than the DG method. While the solution of the linear system using the DG method requires a parallel solver, the solution using the SUPG method is obtained in Matlab on a single processor, in relatively less time.

Some useful conclusions can be drawn from this comparison: The DG method is suited for seeking a high order accurate solution on unstructured grids. However, for problems requiring a global assembly, the DG method is expensive in memory requirements compared to conventional continuous Galerkin methods. Even then, the use of the DG method is recommended in those cases where:

- The physics of the problem can be expressed using a single governing equation, and
- An adaptive refinement needs to be performed on unstructured meshes - The DG method can incorporate the  $h - p$  refinement very easily.

However, for an elliptic system of equations, as in the case of LEE, the use of the DG method poses prohibitive computational memory constraints compared to the continuous FEM. The memory requirements will be much greater for three dimensional calculations also. This is due the increased number of nodal variables per element in three

dimensions, especially for higher orders. For example, the number of variables in a second order triangular element in two dimensions is  $N = 6$  and for a three dimensional tetrahedron,  $N = 10$ . In addition, since the number of surfaces is higher in three dimensions compared to the number of sides in two dimensions, the global matrix is expected to be denser due to the additional surface flux terms.

Although finite difference methods are as good as continuous FEM in terms of the memory requirements, they are not applicable on unstructured grids. In the following sections, the application of continuous Galerkin methods to frequency domain aeroacoustic problems is considered. Particularly, the emphasis will be on the use of the streamline upwind Petrov Galerkin (SUPG) method.

## 5.5 The Streamline Upwind Petrov Galerkin method

Conventional continuous Galerkin methods were originally developed for problems in structural mechanics, where the governing equations are elliptic in nature. It has been shown in [7] that the standard continuous Galerkin method is not well behaved for convection dominated problems, that are hyperbolic in nature. Let us consider the application of the standard Galerkin method to a scalar transport equation in the absence of diffusion:

$$\vec{\mathbf{a}} \cdot \nabla q + cq = s. \quad (5.13)$$

Let the convection velocity,  $\vec{\mathbf{a}}$  and the coefficient  $c$  be constants. The discretized weak form using the element approximation (2.2) is:

$$\int_{\Omega^e} \psi_k (\vec{\mathbf{a}} \cdot \nabla \psi_j) q_j^e d\Omega^e + \int_{\Omega^e} c (\psi_k \psi_j) q_j^e d\Omega^e = \int_{\Omega^e} \psi_k s d\Omega^e, \quad k, j = 1 \dots N. \quad (5.14)$$

The weak form has an unsymmetrical term associated with the convection operator. Consequently, spurious node to node oscillations appear in the solution [7]. This is better understood by recalling that the standard Galerkin method is similar to a central finite difference scheme, which suffers from dissipation errors for convection dominated problems. Initial attempts to stabilize the Galerkin method included adding an artificial dissipation term to the weak form, much similar to the upwind finite difference schemes [64]. Later attempts were the introduction of the Petrov Galerkin method, in which the test functions in equation (5.14) are modified to add a symmetric term to the weak form. Several variations of the Petrov Galerkin method were then proposed [64, 7] to stabilize the weak form, such as the streamline upwind Petrov Galerkin (SUPG) method, and the Galerkin least squares (GLS) method.

In the SUPG method, an upwind biased diffusion term is added [64] to the weak form by modifying the test function:

$$\psi \Rightarrow \psi + \tau^e \vec{\mathbf{a}} \cdot \nabla \psi. \quad (5.15)$$

Here,  $\tau^e$  is a stabilizing parameter controlling the artificial dissipation, and is determined based on the element size, the reaction coefficient  $c$  and the convection velocity. A possible candidate for  $\tau^e$  when  $c = 0$  is:

$$\tau^e = \frac{\alpha h^e}{2|\mathbf{a}|}, \quad (5.16)$$

with  $\alpha$  as a positive constant. The discretized weak form in equation (5.14) now becomes:

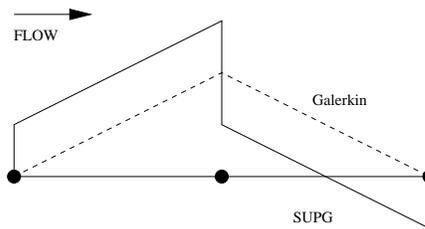
$$\int_{\Omega^e} \psi_k (\vec{\mathbf{a}} \cdot \nabla \psi_j q_j^e d\Omega^e + c \psi_k \psi_j q_j^e - s) d\Omega^e$$

$$+ \int_{\Omega^e} \tau^e \vec{a} \cdot \nabla \psi_k (\vec{a} \cdot \nabla \psi_j q_j^e + \psi_j q_j^e - s) d\Omega^e = 0. \quad (5.17)$$

The second term corresponds to the SUPG formulation. The product of the gradients of the test and trial functions in the first term of the SUPG expression is essentially the artificial diffusion. This is equivalent to adding a diffusion term to the left hand side in the original equation (5.13) of the form:

$$\mu_{kj} \frac{\partial}{\partial x_k} \left( \frac{\partial q}{\partial x_j} \right) \quad \text{with} \\ \mu_{kj} = \tau^e a_k a_j \quad \text{for } k, j = 1 \dots d \quad (5.18)$$

The diffusion is introduced only in the direction of the resultant velocity, due to the inner product of the velocity vector with the gradient of the test function in equation (5.15). This accounts for the upwind bias [64]. Considering the one dimensional case, the basis functions in equation (5.15) at a node are schematically sketched in Fig. 5.7 for  $p = 1$ . It can be seen that the element upstream of the node has more weight compared to the downstream.



**Figure 5.7.** The one dimensional basis functions for  $p = 1$ .

## 5.6 Literature on the SUPG method

The SUPG method was introduced simultaneously by Hughes and Brooks [65] and Kelly *et al.* [66], in 1980. Hughes and Mallet [67] established error estimates for general linear advective diffusive multidimensional systems using the SUPG method. The dependence of the convergence properties of the SUPG method on the stability parameter was analyzed for scalar advection diffusion systems by Franca *et al.* [68]. A comparison of the SUPG method with other stabilized Galerkin methods such as the GLS method was conducted by Codina [69]. In 1987, Hughes [70] gave an overview of the applications of the SUPG method to compressible and incompressible Navier-Stokes equations. The relation between the DG method and the SUPG method was also explained in the same work. Application of the SUPG method to an advective-diffusive problem was analyzed in detail by Franca *et al.* [68]. Beau *et al.* [71] tested the use of entropy variables in the SUPG method for the solution of compressible Navier-Stokes equations in the presence of shocks. They also described the design and addition of a shock capturing term to the SUPG formulation. Recently, Behr [72] applied stabilized Galerkin methods in the finite element formulation for free surface flows.

The selection of the stability parameter for systems of hyperbolic equations is still an active research area. The SUPG formulation for a system of equations involves the flux coefficient matrices in the place of the convection velocity components in equation (5.17). Also, the stability parameter can be a matrix in such a formulation. The two components in the expression for the stabilizing parameter for a scalar variable are the convection speed and the element size. The selection of values equivalent to these two parameters in multiple variable problems, in multiple dimensions has attracted significant attention in research. Some of the works in this area are noted below.

Hughes and Mallet [67] first discussed the selection of the stability parameter for

general multidimensional systems. However, some assumptions were made so that the system could be decoupled. Codina [73] discussed the selection of the stability parameter for nonlinear systems of equations. Codina also developed a methodology for choosing the stabilization parameter on the basis of inverse estimates of the coefficient matrices for some particular examples, such as multidimensional Stokes equation and Reissner-Mindlin plate bending equations. Beau *et al.* [71] used the spectral radius of the flux coefficient matrices to define the stability parameter for compressible Euler equations. Recently, Tezduyar and Osawa [74] developed a methodology for selecting the stability parameter based on the element matrices and vectors of the Galerkin formulation. They also looked at various possibilities of defining the element length for different types of elements in multiple dimensions. Tezduyar and Sathe [75] improved on this formulation and applied the method to advection-diffusion and Navier-Stokes equations. It is clear that a consistent methodology for the design of the stabilizing parameter has not yet been established. The methods used by many of the researchers are either ad-hoc, or involve very complex calculations for specific problems.

## 5.7 SUPG method for the LEE in the frequency domain

Consider the linearized Euler equation defined in  $\Omega = \mathbb{R}^2$  in the frequency domain:

$$-i\omega\tilde{\mathbf{q}} + \frac{\partial \mathbf{A}_r \tilde{\mathbf{q}}}{\partial x_r} + \mathbf{C}\tilde{\mathbf{q}} = \tilde{\mathbf{s}} \quad \text{for } r = 1, 2 \quad (5.19)$$

with  $\mathbf{A}_r$  and  $\mathbf{C}$  as given in section 3.1. The weak form obtained from the SUPG formulation for such a system of equations over an element can be written as:

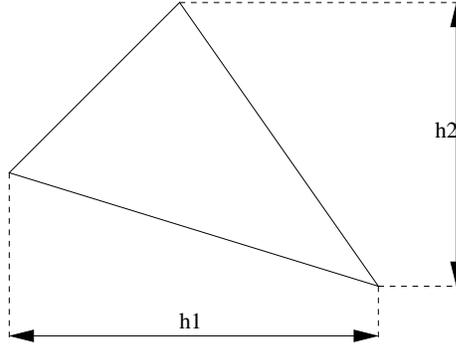
$$\begin{aligned}
\int_{\Omega^e} \psi \left( -i\omega \tilde{\mathbf{q}} + \frac{\partial \mathbf{A}_r \tilde{\mathbf{q}}}{\partial x_r} + \mathbf{C} \tilde{\mathbf{q}} \right) d\Omega^e + \int_{\Omega^e} \tau^e \mathbf{A}_k^T \frac{\partial \psi}{\partial x_k} \left( -i\omega \tilde{\mathbf{q}} + \frac{\partial \mathbf{A}_r \tilde{\mathbf{q}}}{\partial x_r} + \mathbf{C} \tilde{\mathbf{q}} \right) d\Omega^e \\
= \int_{\Omega^e} \psi \tilde{\mathbf{s}} d\Omega^e + \int_{\Omega^e} \tau^e \mathbf{A}_k^T \frac{\partial \psi}{\partial x_k} \tilde{\mathbf{s}} d\Omega^e \quad (5.20)
\end{aligned}$$

In the conventional SUPG method, the test function is given by  $\psi + \mathbf{A}_r \frac{\partial \psi}{\partial x_r}$ . However, the use of  $\mathbf{A}_r^T$  in the place of  $\mathbf{A}_r$  in the stabilizing term has been shown to produce superior results for non-linear problems [73]. Also, Beau *et al.* [71] have used  $\mathbf{A}_r^T$  in the stabilizing term for Euler equations applied to compressible flows with shocks. In this work, since the linearized version of the compressible Euler equations is used,  $\mathbf{A}_r^T$  is used in the formulation.

The choice of the stability parameter matrix has a significant influence on the result obtained using the SUPG method. The simplest method of selecting the stability parameter involves replacing the convection speed by functions of the eigenvalues of the flux coefficient matrices. In this work, an expression similar to that provided in Beau *et al.* [71] is used. The stability parameter matrix is expressed as

$$\tau^e = \max \left( \frac{\alpha h_r^e}{\lambda_r} \right) \mathbf{I}, \quad (5.21)$$

with  $\mathbf{I}$  being a  $4 \times 4$  identity matrix. The element size  $h_r^e$  is the width of the element in the  $r^{\text{th}}$  direction, as shown in Fig. 5.8.  $\lambda_r$  is the spectral radius of the coefficient matrix  $\mathbf{A}_r$ .



**Figure 5.8.** The element lengths in a triangle.

### 5.7.1 The discretized equation

The discretized form of the weak formulation in an element can be written as:

$$\mathbf{P}^e \tilde{\mathbf{q}}^e = \mathbf{f}^e \quad (5.22)$$

with

$$\begin{aligned} \mathbf{P}^e = & \left[ -i\omega \mathbf{I} \otimes \mathbf{M}^e + \mathbf{A}_r \odot \mathbf{K}_r^e + \mathbf{C}^e \odot \mathbf{M}^e \right] \\ & + \left[ -i\omega [\boldsymbol{\tau}^e \mathbf{A}_k^T] \odot \mathbf{K}_k^{eT} + [\boldsymbol{\tau}^e \mathbf{A}_k^T \mathbf{A}_r] \odot \mathbf{V}_{kr} + [\boldsymbol{\tau}^e \mathbf{A}_k^T \mathbf{C}] \odot \mathbf{M}^e \right] \end{aligned} \quad (5.23)$$

and

$$\mathbf{f}^e = \left[ \mathbf{I} \otimes \mathbf{M}^e + [\boldsymbol{\tau}^e \mathbf{A}_k^T] \odot \mathbf{K}_k^{eT} \right] \tilde{\mathbf{s}}^e \quad \text{for } k, r = 1, 2 \quad (5.24)$$

The element matrices and the symbols are as defined before for the DG algorithm. The diffusion matrix in an element is defined as

$$\mathbf{V}_{kr} = \int_{\Omega^e} \frac{\partial \psi}{\partial x_k} \frac{\partial \psi}{\partial x_r} d\Omega^e. \quad (5.25)$$

A global assembly is carried out with the element matrices and vectors of equation (5.22). After imposing the boundary conditions, the resulting linear system is solved using direct solvers.

### 5.7.2 The boundary conditions

A buffer zone type boundary condition similar to the type described in section 5.2.1 is implemented in two dimensional calculations. The width of the buffer region is taken as 8 – 10 wavelengths of oscillations of the solution. In the two dimensional calculations, the values of the parameters  $\varepsilon$  and  $\kappa$  in equation (5.8) are chosen to be 5.0 and 1.0 respectively.

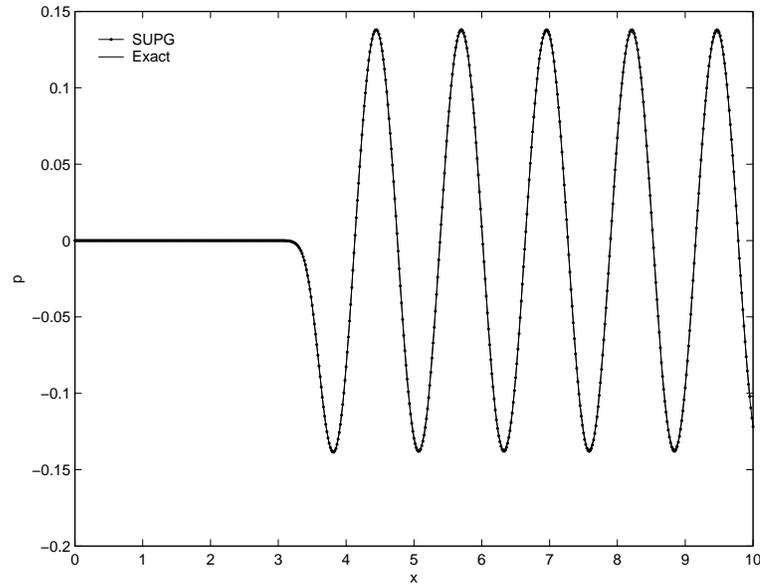
## 5.8 Applications

In this section we consider the application of the SUPG method to the wave equation in one dimension and the linearized Euler equations in two dimensions for acoustic propagation in a plane shear layer.

### 5.8.1 One dimensional wave equation

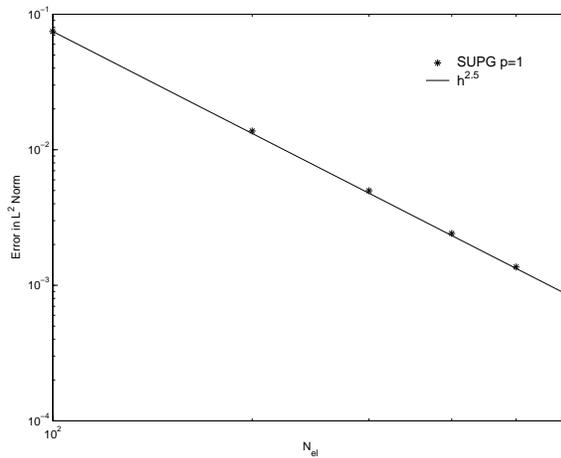
The SUPG method is applied to the one dimensional inhomogeneous wave equation in the frequency domain, to understand and validate the algorithm for a system of equations in a simple case. The governing equations, the source variation and the computational domain are the same as in section 5.3.1. Characteristic boundary conditions are implemented at the boundaries. A comparison of the solution obtained using SUPG with the exact solution is shown in Fig. 5.9. The basis set is made of first degree Lagrange polynomials. The domain is divided into 300 equal parts. That is,  $h = 0.033$ . The stabilizing

parameter is chosen to be  $\tau^e = 0.5h/a$ . The  $L^2$  norm of the global error between the ex-



**Figure 5.9.** Comparison of the exact and SUPG solution for the 1d wave equation.

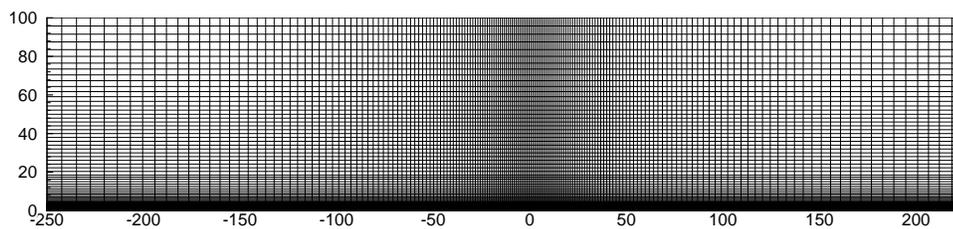
act and the SUPG solution is calculated for varying number of elements. The variation is shown in Fig. 5.10.



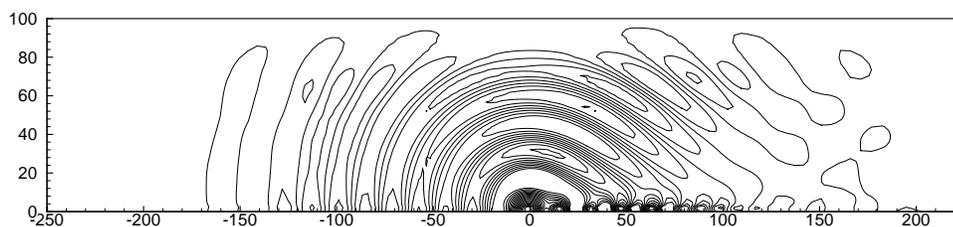
**Figure 5.10.** Convergence of the SUPG solution for the 1d wave equation.

## 5.8.2 Two dimensional LEE

The problem considered is the same as the benchmark problem involving the instability wave solution, as described in section 5.3.2. The computational domain is taken to be  $-250 < x < 225$  and  $0 < y < 100$ . In the buffer region, a damping function of the form in equation (5.8) is used, with  $\varepsilon = 5$  and  $\kappa = 1.0$ . The computation is performed on two grids made of similar right angle triangles. The first grid has  $89 \times 49$  nodes and the second has  $177 \times 97$  nodes in  $x$  and  $y$  directions respectively. The second grid is shown in Fig. 5.11a, without the triangulation. The triangles are created by diagonally halving the rectangular elements. The nodes are clustered near the origin in both



(a)

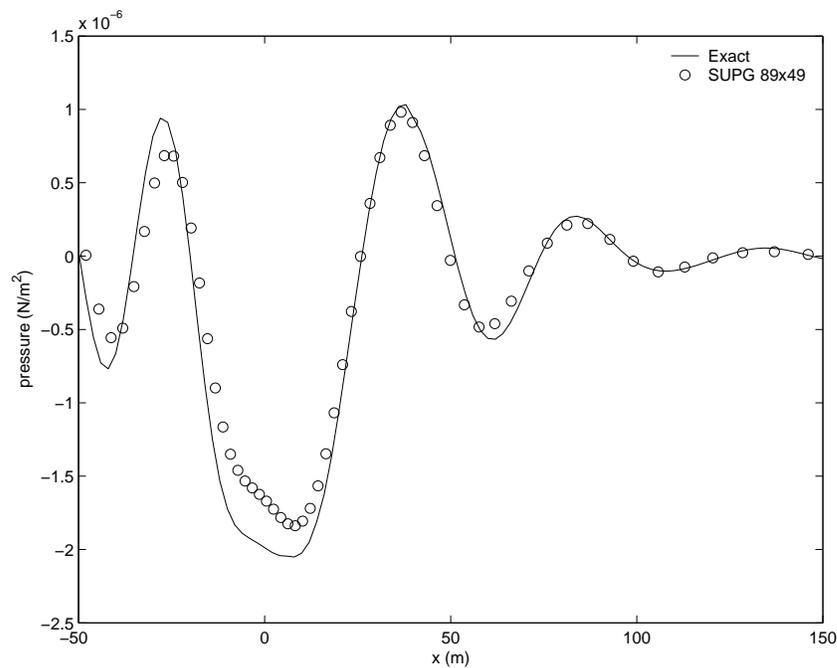


(b)

**Figure 5.11.** The SUPG method in 2d LEE. (a) The  $177 \times 97$  grid used. (b) A contour plot of the acoustic pressure.

the directions to resolve the source and the high mean flow gradient. The resulting numerical linear algebraic system is solved using a direct solver in Matlab [59], using LU decomposition. The size of the global matrix to be inverted is  $(4 \times N_{nodes})^2$ . The

actual LU decomposition and the solution took only 4 minutes for the finer grid case, on a 1GHz Pentium IV processor, with a 512 MB RAM. A contour plot of the acoustic pressure obtained using a first order SUPG method is shown in Fig. 5.11b. The values of the instantaneous pressure along  $y = 15$  at the end of a cycle of oscillation of the source for the  $89 \times 49$  grid is shown in Fig. 5.12. The improved solution, which is much closer to the exact solution, is shown in Fig. 5.13.

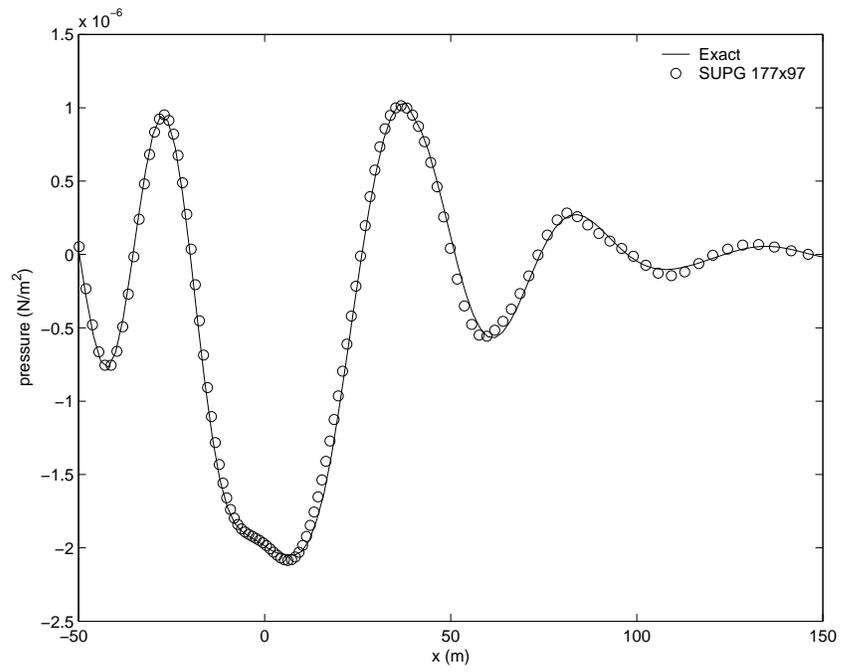


**Figure 5.12.** The pressure values at  $y = 15$ : The  $89 \times 49$  grid.

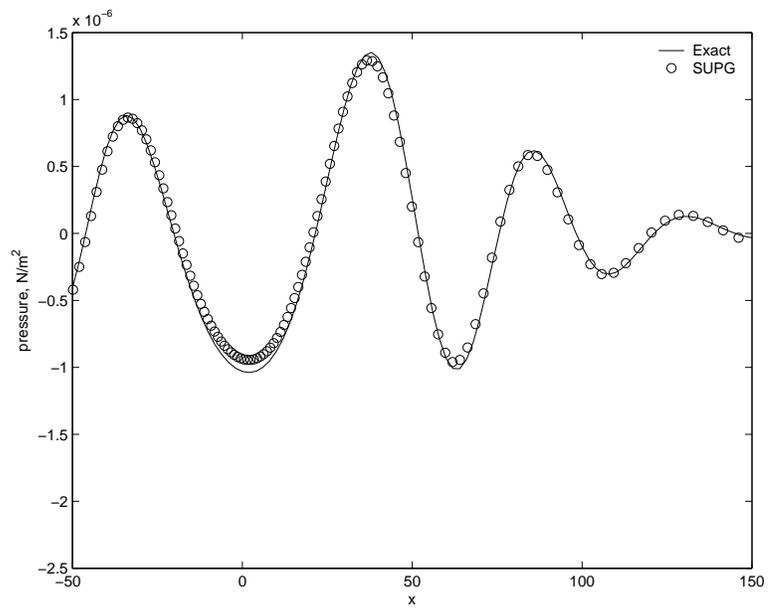
The values of the pressure along  $y = 50$  using the finer grid is shown in Fig. 5.14. The exact solution is obtained from Agarwal *et al.* [6].

## 5.9 Observations and future directions

The SUPG method has an inherent damping, which increases with the size of the element. This can be seen from the expression (5.16). This basically means that the



**Figure 5.13.** the 177 × 97 grid.



**Figure 5.14.** The pressure values at y = 50.

artificial dissipation is decreased as the grid resolution is increased. The grid in the buffer region is more coarse compared to the physical domain, and hence there is an inherent damping of the acoustic waves in the SUPG method. This needs to be taken into account while choosing the parameters in the damping function in equation (5.8). There is also a need to reduce the length of the buffer region surrounding the problem domain, to cut down on the computation time. This requires a study on the damping parameters, as well as an implementation of PML in the frequency domain.

The results presented here were obtained using a value of  $\alpha = 0.125$  in equation (5.21). It was observed that the solution is very sensitive to the value of  $\alpha$ , and subsequently the stabilizing parameter matrix. The design of the stability parameter matrix requires significant research and analysis. In this implementation of the SUPG method, a fairly simple expression (5.21) was chosen for the stability parameter matrix. Methods using the inverse estimates of the coefficient matrices and the source matrices, as mentioned in Codina [73] are expected to be more accurate. Also, for higher order solutions, estimates based on the element matrices  $\mathbf{M}^e$  and  $\mathbf{K}^e$  have to be evaluated, since they have been documented [74] to produce better solution in such cases.

# Chapter 6

## Conclusions

Several contributions have been made in this research to the application of numerical methods to computational aeroacoustics on unstructured grids. The work can be broadly classified to be in three main areas, as follows.

A solver using the DG method has been developed for the linearized Euler equations in the time domain. An in-depth analysis of various factors affecting the overall computational cost of a solver using the DG method is conducted. A novel method to reduce the computational cost using a Lagrange polynomial basis set to approximate the flux expressions has been developed. The element integrals have been shown to be easily evaluated for a basis set made of any polynomial, without the use of quadrature. As a result of this analysis, the final DG solver developed in this study is considerably more efficient compared to the recently developed, and presently the most popular quadrature-free DG method using moment polynomials. It is concluded that the choice of the basis set polynomials plays a critical role in determining the efficiency of the DG solver. The use of polynomials that reduce the edge flux as well as flux expansion computations significantly reduces the computation time.

The time domain DG solver has been transformed to run on parallel platforms, so

that it is practically useful for realistic flow applications involving fully non-uniform mean flows. The parallel version of the solver is capable of using a domain decomposition with unstructured partitions. The solver has been validated using several benchmark problems involving acoustic propagation with uniform and non-uniform mean flows. The sound propagation in a realistic aircraft jet engine exhaust configuration from a source upstream of the exit of the nozzle has been calculated to the far field. It is concluded that in many practical time domain applications, the mean flow solution obtained using CFD might have to be interpolated to a new grid that is generated particularly for the simulation of noise propagation to the far field observer locations.

The time domain solution of the LEE includes the Kelvin-Helmholtz instability waves that are solutions to the LEE, but not the original governing equations, namely, the full Navier-Stokes equations. To suppress the instability waves in the solution, the development of a frequency domain solver has been undertaken. The test problem in two dimensions is a recent benchmark problem of acoustic refraction and propagation in the shear layer of a parallel jet. The DG method is considered initially. The frequency domain calculations involve the solution of a global matrix, and the DG method produces a prohibitively large algebraic system. It is concluded that the DG method is not suited for aeroacoustic applications in the frequency domain, if adaptive meshing is not to be used. The use of continuous Galerkin methods has also been explored, and the suppression of instability waves has been demonstrated with the use of the SUPG method. Unlike other frequency domain solutions in the present literature, no assumptions are made regarding the nature of the mean flow. The solver using the SUPG method has been shown to be applicable for a non-uniform mean flow, which makes it more suitable for practical problems in aeroacoustics.

In terms of the computational time, the time domain methods have been shown to be more expensive compared to the frequency domain methods. However, the time domain

methods have the ability to provide a solution over multiple frequencies in a single computation. For practical problems of aeroacoustics in aeronautical applications, a combined use of both the time domain and frequency domain methods is helpful. The work in this thesis is a fundamental step in that direction.

## 6.1 Suggestions for future work

In this section, some suggestions are provided for the future work in the direction of the present research. This can be concentrated in two main categories: first, the improvement and extension of the existing finite element algorithms, and second, the addition of new features to the solvers and their application to the problems in the aeroacoustics of aircrafts and other vehicles.

An extension of the existing algorithms in both the time and frequency domains to three dimensions is necessary in order to obtain results for practical problems in aeroacoustics. In the time domain solver using the DG method, a substantial pre-processing of the standard FEM grid data is necessary to obtain the neighbor cell data. Various other aspects pertaining to the extension of the DG algorithm to three dimensions have also been discussed in section 3.4. The efficiency of the parallel solver could be increased by altering the overlap period between the computation and communication between the processors. A rearrangement of the existing algorithm, along with some changes is necessary for this purpose. A separate algorithm may have to be written for interpolating the mean flow values to the acoustic grid.

In the frequency domain solver using the SUPG method, an extension to three dimensions would involve additional post-processing to extract and visualize the data. At present, the final global matrix is stored in the Harwell-Boeing format, and is read by the direct solver, which is run independently of the SUPG code. A method to incorpo-

rate a parallel direct solver into the SUPG code needs to be developed, to minimize the implementation time. This is crucial for applications in three dimensions, since the size of the global matrix is much larger. Some suggestions regarding possible future work on the stabilization parameter, the element size, and the treatment of boundary conditions in the SUPG method have already been provided in section 5.9.

The solvers developed in this research are applicable to several problems of practical interest, such as jet noise, fan noise and airframe noise. As explained earlier, the time domain solver has the capability to provide a solution for multiple frequencies with a broadband excitation. On the other hand, the frequency domain solver solves for a single frequency, but requires only a fraction of the computation time required for the time domain solver. It would be interesting indeed to compare the two approaches for problems of practical value.

One of the new features that could enhance the capacity of the solvers is the addition of an adaptive mesh, or a mesh with non-conforming elements. It is particularly straightforward to implement this feature using the DG method. A rearrangement of the algorithm, along with a new routine to calculate the element edge integrals in a more general fashion is necessary to incorporate such non-conforming elements.

# Bibliography

- [1] Brentner, K. S. and Farassat, F., “Modeling aerodynamically generated sound of helicopter rotors,” *Progress in Aerospace Sciences*, Vol. 39, 2003, pp. 83–120.
- [2] Mengle, V., “Anomolous effect of nozzle length reduction on jet noise of forced mixers.” *Fifth AIAA/CEAS Aeroacoustics Conference*, No. AIAA 99-1968, Bellevue, Greater Seattle, WA, May 1999.
- [3] Cockburn, B., Karniadakis G. E., and Shu, C. W., “The development of Discontinuous Galerkin methods,” *Discontinuous Galerkin Methods: Theory, Computation and Applications*, edited by B. Cockburn, G. Karniadakis, and C. W. Shu, Lecture Notes in Computational Science and Engineering 11, Springer Publications, New York, 1999, pp. 5–50.
- [4] Atkins, H. L., “Steps toward a robust high-order simulation tool for aerospace applications,” *Discontinuous Galerkin Methods: Theory, Computation and Applications*, Lecture notes on computational science and engineering, 11, Springer Publications, New York, 1999, pp. 53–61.
- [5] Atkins, H. L. and Shu, C. W., “Quadrature free implementation of Discontinuous Galerkin method for hyperbolic equations,” *AIAA Journal*, Vol. 36, No. 5, May 1998.
- [6] Agarwal, A., Morris, P. J., and Mani, R., “The calculation of sound propagation in non-uniform flows: suppression of instability waves,” *AIAA Journal*, Vol. 42, No. 1, Jan. 2004.
- [7] Donea, J. and Huerta, A., *Finite Element Methods for Flow Problems*, Wiley publishers, 2003.
- [8] Cockburn, B., “Discontinuous Galerkin methods,” Tech. Rep. Preprint Series No. 1921, Institute of Mathematics and its Applications (IMA), University of Minnesota, MN, May 2003, <http://www.ima.umn.edu/preprints/new.preprintlist.html>.
- [9] Cockburn, B., “Lecture notes on Discontinuous Galerkin method for convection dominated problems,” <http://www.math.umn.edu/cockburn/LectureNotes.html>.

- [10] Reed, W. H. and Hill, T. R., "Triangular mesh methods for the neutron transport equation," Tech. Rep. Tech. Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
- [11] Lesaint, P. and Raviart, P. A., "On a finite element method for solving neutron transport equation," *Mathematical Aspect of Finite Elements in Partial Differential Equations*, Academic Press, NY, 1974, pp. 89–123.
- [12] Johnson, C. and Pitkaranta, J., "An analysis of the Discontinuous Galerkin method for scalar hyperbolic equations," *Mathematics of computation*, Vol. 46, No. 173, Jan. 1986, pp. 1–26.
- [13] Hu, F. Q., Hussaini, M. Y., and Rasetarinera, P., "An analysis of the Discontinuous Galerkin method for wave propagation problems," *Journal of Computational Physics*, Vol. 151, 1999, pp. 929–946.
- [14] Hu, F. Q. and Atkins, H. L., "Eigensolution analysis of the Discontinuous Galerkin method with non-uniform grids, part I: One space dimension," Tech. Rep. NASA/CR-2001-211269, ICASE Report no. 2001-40, Dec. 2001.
- [15] Hu, F. Q. and Atkins, H. L., "Two dimensional wave analysis of the Discontinuous Galerkin method with non uniform grids and boundary conditions," *8th AIAA/CEAS Aeroacoustics Conference*, No. AIAA 2002-2514, Breckenridge, CO, Jun. 2002.
- [16] Stanescu, D., Kopriva, D. A., and Hussaini, M. Y., "A dispersion analysis for discontinuous spectral element methods," *Journal of Scientific Computing*, Vol. 15, No. 2, 200, pp. 149–171.
- [17] Cockburn, B. and Shu, C. W., "TVB Runge-Kutta local projection Discontinuous Galerkin finite element method for conservation laws II: General framework," *Mathematics of Computation*, Vol. 52, No. 186, Apr. 1989, pp. 411–435.
- [18] Karniadakis, G. E. and Sherwin, S. J., *Spectral/hp Element Methods for CFD*, Numerical Mathematics and Scientific Computation series, Oxford University press, New York, 1999.
- [19] Lockard, D. P. and Atkins, H. L., "Efficient implementations of the quadrature free Discontinuous Galerkin method," *14th AIAA CFD Conference*, No. AIAA Paper 99-3309, Norfolk, VA, Jun. 1999.
- [20] Stanescu, D., Hussaini, M., and Farassat, F., "Aircraft engine noise scattering- A discontinuous spectral element approach," *40th AIAA Aerospace Sciences Meeting and Exhibit*, No. AIAA Paper 2002-0800, Reno, NV, Jan. 2002.

- [21] Remaki, M., Habashi, W. G., Ait-Ali-Yahia, D., and Jay, A., "A 3D Discontinuous Galerkin method for multiple pure tone noise problems," *40th AIAA Aerospace Sciences Meeting and Exhibit*, No. AIAA Paper 2002-0229, Reno, NV, Jan. 2002.
- [22] Atkins, H. L., "Continued development of the Discontinuous Galerkin method for computational aeroacoustic applications," *Third AIAA/CEAS Aeroacoustics Conference*, No. AIAA Paper 97-1581, Atlanta, GA, May 1997.
- [23] Atkins, H. L., "Application of the Discontinuous Galerkin method to acoustic scatter problems," *Second CAA Workshop on Benchmark Problems*, edited by C. K. W. Tam and J. C. Hardin, No. NASA Conference Publication 3352, Tallahassee, FL, Jun. 1997, pp. 45–56.
- [24] Lockard, D. P. and Atkins, H. L., "A high-order method using unstructured grids for the aeroacoustic analysis of realistic aircraft configurations," *5th AIAA/CEAS Aeroacoustics Conference*, No. AIAA paper 99-1945, Bellevue, WA, May 1999.
- [25] Rao, P. P. and Morris, P. J., "Application of a generalized quadrature free Discontinuous Galerkin method in aeroacoustics," *8th CEAS/AIAA Aeroacoustics Conference*, No. AIAA paper 2003-3120, Hilton Head, SC, 2003.
- [26] Rao, P. P. and Morris, P. J., "A generalized quadrature free Discontinuous Galerkin method," *Second MIT Conference on Computational Solid and Fluid Mechanics*, Boston, MA, 2003.
- [27] Fletcher, C. A. J., *Computational Galerkin Methods*, Springer Verlag, 1984.
- [28] Wasistho, B., Geurts, B. J., and Kuerten, J. G. M., "Simulation techniques for spatially evolving instabilities in compressible flow over a flat plate," *Computers and Fluids*, Vol. 26, No. 7, 1997, pp. 713–739.
- [29] Berenger, J. P., "A Perfectly Matched Layer for the absorption of electromagnetic waves," *Journal of Computational Physics*, Vol. 114, 1994, pp. 185–200.
- [30] Hu, F. Q., "On absorbing boundary conditions for linearized Euler equations by a Perfectly Matched Layer," *Journal of Computational Physics*, Vol. 129, 1996, pp. 201–219.
- [31] Hu, F. Q., "On constructing stable Perfectly Matched Layers as an absorbing boundary condition for Euler equations." No. AIAA 2002-0227, Jan. 2002.
- [32] Tam, C. K. W. and Auriault, L., "Perfectly Matched Layer as an absorbing boundary condition for the linearized Euler equations in open and ducted domains." *Journal of Computational Physics*, , No. 144, 1998, pp. 213–244.

- [33] Hu, F. Q., “A stable Perfectly Matched Layer for linearized Euler equations in unsplit physical variables,” *Journal of Computational Physics*, Vol. 173, 2001, pp. 455–480.
- [34] Benchmark Problems - Category 4, *ICASE/LaRC Workshop on Benchmark Problems in Computational Aeroacoustics.*, Hampton, VA, Oct. 1994.
- [35] Benchmark Problems - Category 5, “Generation and radiation of acoustic waves from a 2D shear layer,” *Third Computational Aeroacoustics (CAA) Workshop on Benchmark Problems*, NASA CP 2000-209790, Cleveland, OH, Nov. 1999.
- [36] Agarwal, A. and Morris, P. J., “Generation and radiation of acoustic waves from a 2-D shear layer,” *Third Computational Aeroacoustics (CAA) Workshop on Benchmark Problems*, NASA CP 2000-209790, NASA Glenn Research Center, Aug. 2000, pp. 309–314.
- [37] Fornberg, B., *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, Cambridge, UK, 1996.
- [38] Chen, Q. and Babuska, I., “Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 128, 1995, pp. 405–417.
- [39] Chen, Q. and Babuska, I., “The optimal symmetrical points for polynomial interpolation of real functions in the tetrahedron,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 137, 1996, pp. 89–94.
- [40] Karypis, G., *METIS, Version 4.0*, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, <http://www-users.cs.umn.edu/karypis/metis/>, Nov. 1998.
- [41] *Tecplot*, <http://www.amtec.com>, 8th ed.
- [42] Baggag, A., Atkins, H. L., Ozturan, C., and Keyes, D., “Parallelization of an object oriented unstructured aeroacoustics solver,” NASA/CR-1999-209098 ICASE Report no 99-11, Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA, 1999.
- [43] Biswas, R., Devine, K. D., and Flaherty, J., “Parallel, adaptive finite element methods for conservation laws,” *Applied Numerical Mathematics*, Vol. 14, No. 1-3, 1994, pp. 225–283.
- [44] Bey, K. S., Oden, J. T., and Patra, A., “A parallel hp-adaptive Discontinuous Galerkin method for hyperbolic conservation laws,” *Applied Numerical Mathematics*, Vol. 20, 1996, pp. 321–336.

- [45] Baggag, A., Atkins, H. L., and Keyes, D., "Parallel implementation of the Discontinuous Galerkin method," NASA/CR-1999-209546 ICASE Report no 99-35, Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA, 1999.
- [46] Atkins, H. L. and Lockard, D. P., "A high order method using unstructured grids for the aeroacoustic analysis of realistic aircraft configurations," *5th AIAA/CEAS Aeroacoustics Conference*, No. AIAA 99-1945, Bellevue, Washington, May 1999.
- [47] Xu, J., Stanescu, D., Hussaini, M. Y., and Farassat, F., "Computation of engine noise propagation and scattering off an aircraft," *International Journal of Aeroacoustics*, Vol. 1, No. 4, 2002, pp. 403–420.
- [48] Öztekin, B. U., *PMVIS - Partitioned Mesh Visualizer*, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, <http://www-users.cs.umn.edu/oztekin/pmvis/>.
- [49] Gropp, W., Lusk, E., and Skjellum, A., *Using MPI - Portable Parallel Programming With the Message Passing Interface.*, The MIT Press, Cambridge, MA, 1996.
- [50] Benchmark Problems Category 1 - Problem 2, "Acoustic scattering," *Second CAA Workshop on Benchmark Problems*, edited by C. K. W. Tam and J. C. Hardin, No. NASA Conference Publication 3352, Tallahassee, FL, Jun. 1997, pp. 1–2.
- [51] Geuzaine, C. and Remacle, J.-F., *Gmsh*, <http://www.geuz.org/gmsh/>, 1st ed., 25 Jun. 2003.
- [52] Kurbatskii, K. A., "Analytical solutions of the category 1 benchmark problems 1 and 2." *Second CAA Workshop on Benchmark Problems*, edited by C. K. W. Tam and J. C. Hardin, No. NASA Conference Publication 3352, Tallahassee, FL, Jun. 1997, pp. 9–17.
- [53] Zygmunt, S., *Flow and noise predictions from nozzles with internal mixers*, Master's thesis, Department of Aerospace Engineering, The Pennsylvania State University, University Park, PA, 2004.
- [54] Agarwal, A., *The prediction of tonal and broadband slat noise*, Ph.D. thesis, The Pennsylvania State University, University Park, PA, May 2004.
- [55] Cockburn, B. and Shu, C. W., "The local Discontinuous Galerkin method for time dependent convection - diffusion systems," *SIAM Journal of Numerical Analysis*, Vol. 35, No. 6, Dec. 1998, pp. 2440–2463.
- [56] Oden, J. T., Babuska, I., and Baumann, C. E., "A discontinuous hp finite element method for diffusion problems," *Journal of Computational Physics*, Vol. 146, 1998, pp. 491–519.

- [57] Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, D., “Discontinuous Galerkin methods for elliptic problems,” *Discontinuous Galerkin Methods: Theory, Computation and Applications*, Lecture Notes in Computational Science and Engineering 11, New York, 1999, pp. 89–101.
- [58] Hesthaven, J. S. and Warburton, T., “High order nodal Discontinuous Galerkin methods for Maxwell’s eigenvalue problem,” Scientific Computing Report Series 2003, Scientific Computing Group, Division of Applied Mathematics, Brown University, <http://www.dam.brown.edu/scicomp/publications/Reports/Y2003/BrownSC-2003-11.ps>, 2003.
- [59] *Matlab -The language of scientific computing*, <http://www.mathworks.com>, 13th ed., 2003.
- [60] Demmel, J. W., Gilbert, J. R., and Li, X. S., *SuperLU*, Lawrence Berkeley National Laboratory, University of California, Berkeley, [http://crd.lbl.gov/xiao/SuperLU/#superlu\\_dist](http://crd.lbl.gov/xiao/SuperLU/#superlu_dist), 2nd ed., Oct. 1999.
- [61] Li, X., “*zm2hb.m*” - A program to convert sparse matrices from coordinate format to Harwell-Boeing format, University of California, Berkeley, <http://math.nist.gov/MatrixMarket/src/zm2hb.m>.
- [62] Roach, G. F., *Green’s Functions*, Cambridge Press, 2nd ed., 1985.
- [63] Benchmark Problems - Category 4 problem 1, “Radiation and refraction of sound waves through a 2-D shear layer,” *Fourth Computational Aeroacoustics (CAA) Workshop on Benchmark Problems*, Cleveland, OH, 20 Oct. 2003.
- [64] Zienkiewicz, O. C. and Taylor, R. L., *The Finite Element Method*, Vol. 3, Butterworth-Heinemann, Oxford, 5th ed., 2000.
- [65] Hughes, T. J. R. and Brooks, A., “A multidimensional upwind scheme with no crosswind diffusion,” *Finite Elements for Convection Dominated Flows*, edited by T. Hughes, Vol. AMD 34 of ASME, 1979.
- [66] Kelly, D. W., Nakazawa, S., and Zienkiewicz, O. C., “A note on anisotropic balancing dissipation in the finite element method approximation to convection diffusion problems,” *International Journal on Numerical Methods in Engineering*, Vol. 15, 1980, pp. 1705–1711.
- [67] Hughes, T. J. R. and Mallet, M., “A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective diffusive systems,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 58, 1986, pp. 305–328.

- [68] Franca, L. P., Frey, S. L., and Hughes, T. J. R., “Stabilized finite element methods: I . Application to the advective-diffusive model,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 95, 1992, pp. 253–276.
- [69] Codina, R., “Comparison of some finite element methods for solving the diffusion-convection-reaction equation,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 156, 1998, pp. 185–210.
- [70] Hughes, T. J. R., “Recent progress in the development and understanding of the SUPG methods with special reference to the compressible Euler and Navier-Stokes equations,” *International Journal for Numerical Methods in Fluids*, Vol. 7, 1987, pp. 1261–1275.
- [71] Beau, G. J. L., Ray, S. E., Aliabadi, S. K., and Tezduyar, T. E., “SUPG finite element computation of compressible flows with the entropy and conservation variables formulations,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 104, No. 397-422, 1993.
- [72] Behr, M., “Stabilized space-time finite element formulations for free surface flows,” *Communications for Numerical Methods in Engineering*, Vol. 11, 2001, pp. 813–819.
- [73] Codina, R., “On stabilized finite element methods for linear systems of convection-diffusion-reaction equations,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 188, 2000, pp. 61–82.
- [74] Tezduyar, T. E. and Osawa, Y., “Finite element stabilization parameters computed from element matrices and vectors,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, 2000, pp. 411–430.
- [75] Tezduyar, T. and Sathe, S., “Stabilization parameters in SUPG and PSPG formulations,” *Journal of Computational and Applied Mechanics*, Vol. 4, No. 1, 2003, pp. 71–88.
- [76] Zhigang, W., “Discontinuous Galerkin method for computational aeroacoustic applications,” ME paper, May 2001, Department of Aerospace ENgineering, The Pennsylvania State University, University Park, PA.

# Appendix A

## The DG formulation using a moment polynomial basis set

A complete formulation of the DG method in two dimensions using moment polynomial basis set is presented in this appendix. Consider a general triangular element as shown on the left in Fig. A.1. The triangle has straight edges, and  $(x_0, y_0)$  are the coordinates of the centroid. A linear transformation is necessary to compute the mass and stiffness matrices for the element. The general element is mapped to an equilateral element in the standard region, as shown on the right in Fig. A.1. The centroid of this element is at the origin in the standard region. The coordinate transformation is achieved using the

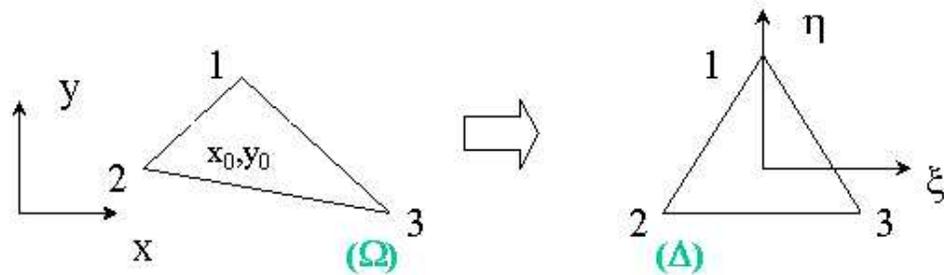


Figure A.1. The transformation.

transformation matrix denoted by  $J_T$ :

$$\begin{Bmatrix} x - x_0 \\ y - y_0 \end{Bmatrix} = J_T \begin{Bmatrix} \xi \\ \eta \end{Bmatrix}, \quad J_T = \begin{bmatrix} a1 & a2 \\ b1 & b2 \end{bmatrix} \quad (\text{A.1})$$

where

$$a1 = x_3 - x_2, \quad a2 = \frac{1}{\sqrt{3}}(2x_1 - x_2 - x_3), \quad \text{and} \\ b1 = y_3 - y_2, \quad b2 = \frac{1}{\sqrt{3}}(2y_1 - y_2 - y_3).$$

In this appendix, vectors are represented by  $\{ \}$  and tensors by  $[ ]$ , and the bold typeface convention for vectors and matrices is dropped. Also, all the symbols are defined as they appear.

The basis set for the element consists of moment polynomials, which are essentially the polynomials in a Taylor series expansion of the solution about the centroid of the element,  $(x_0, y_0)$ . In other words, the basis set in the real region can be denoted by

$$\boldsymbol{\psi} = \{1, (x - x_0), (y - y_0), (x - x_0)^2, (x - x_0)(y - y_0), (y - y_0)^2 \dots\}^T \quad (\text{A.2})$$

and the basis set in the standard equilateral triangle region is formed by the polynomials of a Taylor series expansion about the centroid,  $(0, 0)$ :

$$\boldsymbol{\phi} = \{1, \xi, \eta, \xi^2, \xi\eta, \eta^2 \dots\}^T \quad (\text{A.3})$$

Using equation (A.1) the two basis sets can be related as:

$$\{\boldsymbol{\psi}\} = [A]\{\boldsymbol{\phi}\} \quad (\text{A.4})$$

where  $[A]$  is unique for each element.

## A.1 The mass matrix

The mass matrix is given by,

$$[M]_{element} = \int \{\psi\}\{\psi\}^T dx dy = |J|[A] \left[ \int \{\phi\}\{\phi\}^T d\xi d\eta \right] [A]^T \quad (A.5)$$

where  $|J|$  is the determinant of the Jacobian of transformation given below in equation (A.8). The matrix involving the integral is common for all the elements, and will be denoted by  $[M]$ . Hence, we have

$$[M]_{element} = |J|[A][M][A]^T \quad (A.6)$$

## A.2 The stiffness matrix

The stiffness matrix is formed by the combinations of two matrices (in two dimensions):

$$[K_x] = \int \left\{ \frac{d\psi}{dx} \right\} \{\psi\}^T dx dy, \quad \text{and} \quad [K_y] = \int \left\{ \frac{d\psi}{dy} \right\} \{\psi\}^T dx dy$$

For example, in the case of the advection equation, with advection speeds of ‘ $M_x$ ’ and ‘ $M_y$ ’ in  $x$  and  $y$  directions respectively, the stiffness matrix is given by  $[K]_{element} = -(M_x[K_x] + M_y[K_y])$ . Now, the Jacobian of transformation is defined by,

$$\begin{Bmatrix} d\psi/d\xi \\ d\psi/d\eta \end{Bmatrix} = [J] \begin{Bmatrix} d\psi/dx \\ d\psi/dy \end{Bmatrix} \quad (A.7)$$

or

$$[J] = \begin{bmatrix} \frac{dx}{d\xi} & \frac{dy}{d\xi} \\ \frac{dx}{d\eta} & \frac{dy}{d\eta} \end{bmatrix} = \begin{bmatrix} a1 & b1 \\ a2 & b2 \end{bmatrix} \quad (A.8)$$

Using equation (A.7) and (A.8) the components of the stiffness matrix  $[K_x]$  and  $[K_y]$  are written as:

$$\begin{aligned} [K_x] &= [A] \left[ \int \left( b_2 \left\{ \frac{d\phi}{d\xi} \right\} - b_1 \left\{ \frac{d\phi}{d\eta} \right\} \right) \{\psi\}^T d\xi d\eta \right] [A]^T, \quad \text{and} \\ [K_y] &= [A] \left[ \int \left( -a_2 \left\{ \frac{d\phi}{d\xi} \right\} + a_1 \left\{ \frac{d\phi}{d\eta} \right\} \right) \{\psi\}^T d\xi d\eta \right] [A]^T. \end{aligned} \quad (\text{A.9})$$

The terms in the integral except the 'a's and 'b's are common for all the elements. Hence these can be precomputed and stored.

### A.3 The boundary integrals

The boundary integral is given by

$$\{b\} = \oint \{\psi\} (\vec{F} \cdot \hat{n}) ds \quad (\text{A.10})$$

Where  $\vec{F}$  is the flux vector,  $\hat{n}$  is the unit normal to the edge,  $s$  is the edge coordinate in the real space, and the line integral is performed on the three edges of the element. In linear acoustics, the flux can be considered as a linear function of the flow variable, say  $U$ . The flow variable in turn is represented by the coefficients of the expansion basis set, which is denoted as  $\{u\}$ . We have,  $U = \{u\}^T \{\psi\} = \{\psi\}^T \{u\}$ .

Consider one edge of the triangle. The contribution to the boundary integral is,

$$\{b\} = k \int_e \{\psi\} \left( \{u\}^T \{\psi\} \right) ds,$$

where  $k$  is a scalar term arising from the dot product of the flux vector and the unit normal, involving the sines and cosines of the angle made by the normal to the edge. The subscript  $( )_e$  refers to the fact that the integral is on an edge and the same integral

has to be carried out on the three edges. Using the transformation in (A.1), this can be written as

$$\{b\}_e = k[A] \int_e \{\phi\} \left( \{u\}^T [A] \{\phi\} \right) ds \quad (\text{A.11})$$

Using the same transformation, a set of new variables in the standard space can be derived as:

$$U = \{\psi\}^T \{u\} = \left( [A] \{\phi\} \right)^T \{u\} = \{\phi\}^T [A]^T \{u\} = \{\phi\}^T \{v\}$$

or,

$$\{v\} = [A]^T \{u\} \quad (\text{A.12})$$

Now, there are two transformations involved in carrying out the edge integral:

- The element coordinates have to be transformed to edge coordinates in order to carry out the boundary integral. The transformation is represented as:

$$\{\phi\} = [T] \{S\} \quad (\text{A.13})$$

where  $[T]$  is the transformation matrix, and  $\{S\}$  is the edge basis, given by  $\{S\} = [1, \zeta, \zeta^2, \zeta^3 \dots]^T$  with  $\zeta$  being the edge coordinate in the standard space. The details of the transformation and values of the transformation matrix elements can be found in [5] and [76].

- The edge coordinate in real space,  $s$ , in (A.11) has to be transformed to that in the standard space,  $\zeta$ . This involves a Jacobian, denoted by  $J_e$ , which is a function of the length of the edge, the difference in the coordinates of the vertices, and the elements of the matrix  $[T]$  and  $[J_T]$ .

Using the above two transformations and the notation in (A.12) the boundary integral

can be written as,

$$\{b\}_e = \frac{k}{J_e} [A][T] \left[ \int_e \{S\} \{v\}^T [T] \{S\} d\zeta \right] \quad (\text{A.14})$$

This can be rearranged as:

$$\{b\}_e = \frac{k}{J_e} [A][T] \left[ \int_e \{S\} \{S\}^T d\zeta \right] [T]^T \{v\} \quad (\text{A.15})$$

Here again, the matrix involving the integral, and the edge matrices  $[T]$  are common for all the elements. Finally, this equation is evaluated on each edge to give  $\{b\}$ .

## A.4 The final equation

Assembling the terms, using (A.12), and canceling out the  $[A]$ 's in each term results in:

$$|J|[M] \frac{dv}{dt} = -[K]\{v\} + \{b\}$$

This equation is for the new variable  $\{v\}$  in each element, which needs to be transformed back to the original variable  $\{u\}$  through (A.12). Since  $[M]$  is common for all the elements, it can be readily inverted, and premultiplied with the leading matrices in the expressions for  $[K]$  and  $\{b\}$ . The final equation can be written as:

$$\frac{dv}{dt} = -\frac{1}{|J|} [M]^{-1} [K] \{v\} + \frac{1}{|J|} [M]^{-1} \{b\} \quad (\text{A.16})$$

This equation is solved using the TVBRK schemes developed in Shu [17]. Thus the final solution involves only multiplications and additions of matrices and vectors, instead of matrix inversions.

# Appendix B

## Element vectors and matrices

Here, a full description is given of the evaluation of the element vectors of the final discretized equation (3.3) from the DG formulation in the time domain for the LEE. The basis set is assumed to be made up of Lagrange polynomials of degree  $p$ .

### B.1 The stiffness vector

Following the basic steps in the DG formulation for the advection equation in section 2.2, applied to the LEE leads to

$$\mathbf{k}^e = - \int_{\Omega^e} \frac{\partial \Psi}{\partial x_r} \{ \mathbf{A}_r \mathbf{q} \} d\Omega^e, \quad r = 1, 2. \quad (\text{B.1})$$

Using group interpolation, as defined in equation (2.36), the product of the variables  $\mathbf{q}$  with the elements of  $\mathbf{A}_r$  are approximated within the element as:

$$\mathbf{A}_r^{(kl)} \mathbf{q}^{(l)} \approx \sum_{j=1}^N \mathbf{F}_{r(j)}^{e(kl)} \Psi_j,$$

with  $\mathbf{F}_{r(j)}^{e(kl)} = \left( \mathbf{A}_{r(j)}^{e(kl)} \mathbf{q}_j^{e(l)} \right)$  (B.2)

$$\forall k, l = 1 \dots 4.$$

In the above equation, the Einstein summation convention is not followed for the repeated index  $l$ . The indices  $k, l = 1 \dots 4$ , corresponding to the size of the matrices  $\mathbf{A}_r$  for  $d = 2$ . The stiffness vector can then be expressed as:

$$\mathbf{k}^{e(k)} = - \sum_{l=1}^4 \left[ \int_{\Omega^e} \frac{\partial \psi}{\partial x_r} \psi^T d\Omega^e \right] \mathbf{F}_r^{e(kl)} \quad \forall k = 1 \dots 4, \quad r = 1, 2. \quad (\text{B.3})$$

or,

$$\mathbf{k}^{e(k)} = - \sum_{l=1}^4 [\mathbf{K}_r^e] \mathbf{F}_r^{e(kl)} \quad \forall k = 1 \dots 4, \quad r = 1, 2. \quad (\text{B.4})$$

where  $\mathbf{K}_r^e$  is the part of  $\mathbf{K}^e$  in equation (2.13) corresponding to the  $r^{th}$  component of  $\nabla \psi$ . The size of the stiffness vector is  $4N$ .

## B.2 The reaction vector

The reaction vector is formed in a way similar to that of the stiffness vector. The reaction vector is given by

$$\mathbf{c}^e = - \int_{\Omega^e} \psi \{ \mathbf{C} \mathbf{q} \} d\Omega^e. \quad (\text{B.5})$$

Using group interpolation, the reaction vector is then,

$$\mathbf{c}^{e(k)} = - \sum_{l=1}^4 \mathbf{R}^{e(kl)} \quad \forall k = 1 \dots 4. \quad (\text{B.6})$$

with  $\mathbf{R}^{e(kl)}$  being a vector of size  $N$  given by

$$\mathbf{R}^{e(kl)} = \left\{ \mathbf{C}_j^{e(kl)} \mathbf{q}_j^{e(l)} \right\}, \quad \forall k, l = 1 \dots 4; \quad \text{and } j = 1 \dots N. \quad (\text{B.7})$$

Again, no summation is carried out over the repeated index  $l$ .

### B.3 The edge flux vector

The edge flux term is formed by the sum of the contributions from the three sides of the triangular element:

$$\mathbf{b}^e = \sum_{j=1}^3 \mathbf{b}^{ej} \quad (\text{B.8})$$

where  $\mathbf{b}^{ej}$  is the contribution from the side  $j$ . Using the Lax Friedrich's expression in equation (2.7), the edge flux on the side  $j$  is

$$\mathbf{b}^{ej} = \frac{1}{2} \int_{\Omega^e} \psi \left\{ \overline{(\mathbf{A}_r \mathbf{q})}^{ej} + \overline{(\mathbf{A}_r \mathbf{q})}^{nej} \right\} \vec{n}_r^j d\Gamma^{ej} - \frac{\alpha}{2} \int_{\Omega^e} \psi \left\{ \overline{\mathbf{q}}^{nej} - \overline{\mathbf{q}}^{ej} \right\} d\Gamma^{ej}, \quad (\text{B.9})$$

where the superscript  $( )^{ne}$  denotes the values from the neighbor element. Now, in the discretized set of equations (3.3), each of the equations corresponding to the acoustic variables,  $\rho, u, v$ , and  $p$  in  $\mathbf{q}$  has an edge flux term. This term is denoted as  $\mathbf{b}_{(k)}^{ej}$ , of size  $N$ , for  $k = 1 \dots 4$ . The edge flux term for the equation corresponding to a variable  $\mathbf{q}^{(k)}$  is obtained using (B.9) and (B.2):

$$\begin{aligned} \mathbf{b}_{(k)}^{ej} = & \frac{1}{2} \sum_{l=1}^4 \left[ \int_{\Omega^e} \psi \psi^T d\Gamma^{ej} \right] \left\{ \overline{\mathbf{F}_r^{ej}}^{(kl)} + \overline{\mathbf{F}_r^{nej}}^{(kl)} \right\} \vec{n}_r^j \\ & - \frac{\alpha}{2} \left[ \int_{\Omega^e} \psi \psi^T d\Gamma^{ej} \right] \left\{ \overline{\mathbf{q}^{nej}}^{(k)} - \overline{\mathbf{q}^{ej}}^{(k)} \right\}. \end{aligned} \quad (\text{B.10})$$

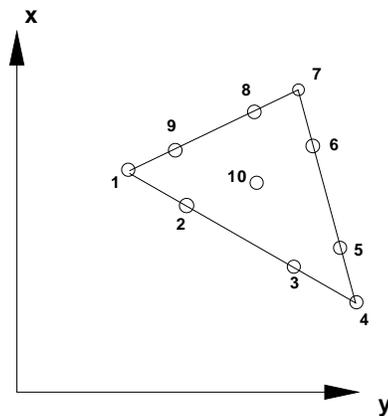
Now, as explained in section 2.4.3, Lagrange polynomials reduce to their one dimensional equivalents along the edges. So, the matrix  $\left[ \int_{\Omega^e} \psi \psi^T d\Gamma^{ej} \right]$  is a rearrangement of the one dimensional mass matrix. The exact structure of this matrix is discussed in the next section. This matrix of size  $N \times N$  is denoted as  $\mathbf{B}^{ej}$ . Then, the edge flux term can

be written as,

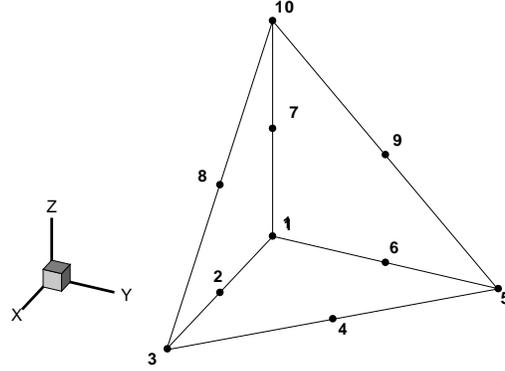
$$\mathbf{b}_{(k)}^{ej} = \frac{1}{2} \sum_{l=1}^4 [\mathbf{B}^{ej}] \left\{ \overline{\mathbf{F}}_r^{ej(kl)} + \overline{\mathbf{F}}_r^{nej(kl)} \right\} \vec{n}_r^j - \frac{\alpha}{2} [\mathbf{B}^{ej}] \left\{ \overline{\mathbf{q}}^{nej(k)} - \overline{\mathbf{q}}^{ej(k)} \right\}. \quad (\text{B.11})$$

### B.3.1 The edge flux matrix

The structure of the edge flux matrix is related to the numbering of the nodes in an element. In the present work, the nodes in a triangle are numbered progressively in a counter-clockwise fashion, until all the nodes on the edges of the element are numbered. Then the nodes inside the triangle are numbered. Fig. B.1 shows a typical triangular element node numbering for a third degree Lagrange polynomial basis set. The numbering in a tetrahedral element can be in a spiral fashion. Fig. B.2 shows a typical tetrahedral element node numbering for a second degree Lagrange polynomial basis set. Let  $\mathbf{nod}_j$



**Figure B.1.** Node numbering in a triangular element for  $p = 3$ .



**Figure B.2.** Node numbering in a tetrahedral element for  $p = 2$ .

denote an ordered set of nodes on a side (or surface)  $j$ . The size of  $\mathbf{nod}_j$  is  $p + 1$  for  $d = 2$ , and  $(p + 1)(p + 2)/2$  for  $d = 3$ . For example, for the triangle shown in Fig. B.1,

$$\begin{aligned}
 \mathbf{nod}_1 &= \{1, 2, 3, 4\} \\
 \mathbf{nod}_2 &= \{4, 5, 6, 7\} \\
 \mathbf{nod}_3 &= \{7, 8, 9, 1\}
 \end{aligned} \tag{B.12}$$

Then the elements of the edge flux matrix  $\mathbf{B}^{ej}$  in  $d$  dimensions are given by

$$\mathbf{B}_{(i,k)}^e = \begin{cases} \hat{\mathbf{M}}_{(l,m)}^e, & \text{if } i, k \in \mathbf{nod}_j, \text{ with } i = \mathbf{nod}_j(l) \text{ and } k = \mathbf{nod}_j(m) \\ 0, & i, k \notin \mathbf{nod}_j, \end{cases} \tag{B.13}$$

$$\forall i, k = 1 \dots N$$

with  $\hat{\mathbf{M}}^e$  being the mass matrix of a standard element in  $d - 1$  dimensions. The edge flux matrix is sparse and the sparsity is higher for higher  $d$ . For time domain calculations, the product of the edge flux matrix with the flux variables in equation (B.10) is not computed by a matrix vector product in the code. Instead, only the values at the nodes of the edge (or surface) are used, saving computational time, as explained in section 2.4.3.

# Appendix C

## The Kronecker product and the row multiple

### C.1 The Kronecker product

The result of the Kronecker product, ' $\otimes$ ', of a matrix  $\mathbf{X}$  of size  $M \times M$  with a matrix of  $\mathbf{Y}$  size  $N \times N$  is defined to be a matrix of size  $MN \times MN$ , made up of  $M \times M$  blocks of  $\mathbf{Y}$  multiplied with the elements of  $\mathbf{X}$ . The  $(k, j)^{th}$  block, which is of size  $N \times N$  is given by,

$$[\mathbf{X} \otimes \mathbf{Y}]^{(kj)} = [\mathbf{X}^{(kj)} \mathbf{Y}], \forall k, j = 1 \dots M. \quad (\text{C.1})$$

An example illustrating the Kronecker product is given below.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \otimes [\mathbf{D}] = \begin{bmatrix} a[\mathbf{D}] & b[\mathbf{D}] \\ c[\mathbf{D}] & d[\mathbf{D}] \end{bmatrix} \quad (\text{C.2})$$

## C.2 The Row multiple

The symbol ‘ $\odot$ ’ is defined to be a “row multiple” of two matrices,  $\mathbf{X}$ , and  $\mathbf{Y}$  of sizes  $M \times M$  and  $N \times N$  respectively. This is a product defined in the case when the elements of  $\mathbf{X}$  are nodal variables, expanded in a basis set of size  $N$ . For example, the flux coefficient matrices are made up of mean flow values that are nodal variables. The “row multiple” of  $\mathbf{X}$  with  $\mathbf{Y}$  is then defined to be made of  $M \times M$  blocks of  $\mathbf{Y}$ , with the columns in every row of  $\mathbf{Y}$  multiplied by the corresponding expansion coefficients of the elements of  $\mathbf{X}$ . The  $(k,j)^{th}$  block of the row multiple, which is of size  $N \times N$  is defined to be:

$$[\mathbf{X} \odot \mathbf{Y}]^{(kj)} = \left[ \mathbf{X}^{e(kj)} {}_m \mathbf{Y}_{nm} \right] \quad \forall k, j = 1 \dots M, \forall m, n = 1 \dots N \quad (\text{C.3})$$

$$\text{with } \mathbf{X}^{(kj)} \approx \sum_{n=1}^N \mathbf{X}^{e(kj)}{}_n \Psi_n.$$

In equation (C.3), the Einstein summation is not carried out on the repeated index  $m$ .

## **Vita**

### **Preetham P. Rao**

Preetham P. Rao was born in a village called Tirthahalli, in the state of Karnataka in India, in 1977. He spent his childhood and got his primary education in the same place. He moved along with his parents and his younger brother to another place called Udupi, where he went to his high school. He finished BE in Mechanical Engineering in 1998, at the Karnataka Regional Engineering College, (KREC) Surathkal (Mangalore University), which is one of the 12 regional engineering institutions established by the government of India. The institution is now autonomous, and is called the National Institute of Technology, Surathkal. Then he moved to Texas, USA, where he obtained a Masters in Aerospace Engineering in 2000 from the Texas A&M University, at College Station. His focus of study was synthetic jet actuators and experimental methods in aerodynamics. He then decided to learn more about computational methods, and in 2000 he came to the Pennsylvania State University at University Park, PA, to enroll in the PhD program in the Department of Aerospace Engineering. He is a member of the AIAA, a recipient of the Regents fellowship from the Texas A&M University and the Incoming Graduate fellowship at the Penn State University. In 2003, he got married to Sowmya Neelavar. Preetham can be reached by email at [raopreetham@yahoo.com](mailto:raopreetham@yahoo.com).