12-10-2005

Adjoint-Based Error Estimation and Grid Adaptation for Functional Outputs from CFD Simulations

Ravishankar Balasubramanian

ADJOINT-BASED ERROR ESTIMATION AND GRID ADAPTATION FOR FUNCTIONAL OUTPUTS FROM CFD SIMULATIONS

By

Ravishankar Balasubramanian

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Computational Engineering
in the Department of Engineering

Mississippi State, Mississippi

December 2005

ADJOINT-BASED ERROR ESTIMATION AND GRID ADAPTATION FOR FUNCTIONAL OUTPUTS FROM CFD SIMULATIONS

By

Ravishankar Balasubramanian

Approved:

James C. Newman III Associate Professor of Aerospace Engineering (Director of Dissertation)	Greg W. Burgreen Associate Research Professor of Computational Engineering (Committee Member)
Edward A. Luke Assistant Professor of Computer Science (Committee Member)	David L. Marcum Professor of Mechanical Engineering (Committee Member)
Mark Janus Associate Professor of Aerospace Engineering and Graduate Coordinator	Roger L. King Associate Dean of Research and Graduate Studies of the Bagley College of Engineering

Name: Ravishankar Balasubramanian

Date of Degree: December 9, 2005

Institution: Mississippi State University

Major Field: Computational Engineering

Major Professor: Dr. James C. Newman, III

Title of Study: ADJOINT-BASED ERROR ESTIMATION AND GRID ADAPTATION FOR FUNCTIONAL OUTPUTS FROM CFD SIMULATIONS

Pages in Study: 133

Candidate for Degree of Doctor of Philosophy

This study seeks to reduce the degree of uncertainty that often arises in computational fluid dynamics simulations about the computed accuracy of functional outputs. An error estimation methodology based on discrete adjoint sensitivity analysis is developed to provide a quantitative measure of the error in computed outputs. The developed procedure relates the local residual errors to the global error in output function via adjoint variables as weight functions. The three major steps in the error estimation methodology are: (1) development of adjoint sensitivity analysis capabilities; (2) development of an efficient error estimation procedure; (3) implementation of an output-based grid adaptive scheme. Each of these steps are investigated.

For the first step, parallel discrete adjoint capabilities are developed for the variable Mach version of the U^2NCLE flow solver. To compare and validate the implementation of adjoint solver, this study also develops direct sensitivity capabilities. A modification is proposed to the commonly used unstructured flux-limiters, specifically, those of Barth-Jespersen and Venkatakrishnan, to make them piecewise continuous and suitable for sensitivity analysis. A distributed-memory message-passing model is employed for the parallelization of sensitivity

analysis solver and the consistency of linearization is demonstrated in sequential and parallel environments.

In the second step, to compute the error estimates, the flow and adjoint solutions are prolongated from a coarse-mesh to a fine-mesh using the meshless Moving Least Squares (MLS) approximation. These error estimates are used as a correction to obtain highly-accurate functional outputs and as adaptive indicators in an iterative grid adaptive scheme to enhance the accuracy of the chosen output to a prescribed tolerance. For the third step, an output-based adaptive strategy that takes into account the error in both the primal (flow) and dual (adjoint) solutions is implemented. A second adaptive strategy based on physics-based feature detection is implemented to compare and demonstrate the robustness and effectiveness of the output-based adaptive approach. As part of the study, a general-element unstructured mesh adaptor employing h-refinement is developed using Python and C++. Error estimation and grid adaptation results are presented for inviscid, laminar and turbulent flows.

ACKNOWLEDGMENTS

A sincere thanks to Professor James C. Newman III for his excellent guidance, friendship and patience throughout my five years here. He has been a great advisor, letting me explore lot of interesting topics and always coming up with insightful comments to move the research forward.

A special thanks to Dr. Greg Burgreen for the discussions on object oriented research and for introducing me to "Hawg" cooking. I am grateful to him for expanding my views on CFD research. I would like to thank the other members of my committee, Professors Edward Luke, David Marcum and Mark Janus for agreeing to serve in my committee. Their constructive comments and suggestions are greatly appreciated.

I thank the ERC system administration for providing me with the resources to complete this work. A special thanks to Kevin for helping with all my grid generation and Eric for helping me with U^2NCLE. My thanks to all my friends and colleagues at ERC. A special acknowledgment to Paresh for all the daily coffee discussions.

A special thanks to my sister and niece for keeping me sane with their regular phone calls.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

TABLE Page

LIST OF FIGURES

NOMENCLATURE

Subscripts:

∞	freestream quantity
r	reference quantity
H	associated with coarse-mesh
h	associated with fine-mesh
^H_h	values prolongated from coarse-mesh to fine-mesh

Fluid Dynamic Variables:

$Q,\ q$	flow solution vector
X	grid coordinates
$R(Q)$	flow residual vector
ρ	fluid density
ρ_r	reference density
u, v, w	flow velocity components in x, y and z directions
U_r	reference velocity
p	fluid pressure
$\rho_r U_r^2$	dynamic pressure
T	temperature
T_r	reference temperature
M_∞, M_r	freestream/reference Mach number
$\beta(M_r)$	preconditioning parameter
L_r	reference length
a_r	reference speed of sound
μ	molecular viscosity

μ_t	eddy viscosity
$\bar{\nu}$	turbulent quantity
Re	Reynolds number
AOA	angle of attack
C_L, C_D	lift and drag coefficients
V_i	volume of control volume surrounding vertex i
∇Q	gradient of solution vector
ϕ_f	flux limiter value at face
Sensitivity Analysis Variables:	
α_k	set of design variables
$\frac{\partial Q}{\partial \alpha_k}$	direct solution vector
γ	adjoint solution vector
$R(\gamma)$	adjoint residual vector
$F(Q, X, \alpha_k)$	objective/cost function
Error Estimation Variables:	
P_H^h	prolongation operator
Q_H	flow solution vector at coarse-mesh
Q_h	flow solution vector at fine-mesh
Q_H^h	prolongated flow solution vector at fine-mesh
γ_H	adjoint solution vector at coarse-mesh
γ_h	adjoint solution vector at fine-mesh
γ_H^h	prolongated adjoint solution vector at fine-mesh
Φ^k	Moving Least Squares shape functions of order k
A_p^1	adjoint-based adaptation parameter
A_p^2	feature-based adaptation parameter
Error Estimation Definitions:	
uniformly refined mesh	fine-mesh obtained using h-refinement

non-uniformly refined mesh	fine-mesh obtained from a grid generator by manually setting the point spacing
linear-linear basis	linear basis for both flow and adjoint solutions
quadratic-quadratic basis	quadratic basis for both flow and adjoint solutions
linear-quadratic basis	linear basis for flow solution and quadratic basis for adjoint solution
quadratic-linear basis	quadratic basis for flow solution and linear basis for adjoint solution
isotropic	radial weights defined by circular supports
anisotropic	tensor product weights defined by rectangular supports
mixed	both radial and tensor product weights defined by circular-rectangular supports
TE	truncation error

CHAPTER I

INTRODUCTION

1.1 Motivation

In engineering analysis and design, computational fluid dynamics (CFD) is typically used to compute specific quantities that assess the performance of the apparatus under investigation. For example, in a system such as an aircraft wing, these quantities are usually the integral output functions such as the lift and drag coefficients. An accurate estimate of these functional outputs is essential for the design of wings. However, because of the approximations made to the governing partial differential equations, and the compromise between the choice of discretization and available computational resources, there is often a degree of uncertainty in CFD simulations about the accuracy of these computed estimates. This reality, therefore, forces the design engineer to include a large factor of safety in his design to accommodate for the lack of a reliable error estimator to guide his design process.

Error estimates of the computed outputs are an invaluable commodity to the designer and may be used to make informed decisions about the factor of safety bounds for improving existing design. Also, these error estimates are of immense help to the CFD engineer in providing a quantitative assessment of the functional error (which provide a global measure of the local residual/discretization errors) and may be used to develop an output-based adaptive approach. An output-based adaptive approach may be able to identify regions of the flow that have significant influence on the output functional and will also provide a better understanding and insight into the relevancy of resolving physical features of the flow such as shock waves, stagnation points and separation lines to improve functional accuracy. Hence, the motivation for an output-based error estimation methodology becomes obvious.

1.2 Survey of Recent Advancements

A brief survey of the research areas that are pertinent to the current study are presented here. Though, this survey is not exhaustive, every attempt has been made to present the most recent advancements in the respective topic areas. The interested reader may use the survey presented in the subsequent sections as a starting point for further study. As a visualization tool, one possible interpretation of the current study is presented in Fig. 1.1. This outline will serve as a road map for the discussions to follow.

1.2.1 Fluid Analysis

Under low Mach number conditions, compressible flow solvers face numerical difficulties because of the large disparity between the convective and acoustic parts of the eigenvalues. If the flow is iso-energetic, i.e., total enthalpy is constant, then steady solutions approach the incompressible constant-density limit as the Mach number approaches zero. Incompressible or artificial compressibility formulations can be used to simulate this class of problems. However, low-speed flows with heat addition have variable density and incompressible formulations are not suitable. Several preconditioning techniques have been proposed to improve the stability and convergence of compressible algorithms in low Mach number regimes. A detailed review of these techniques is reported by Turkel [1, 2]. Typically, a preconditioning matrix is introduced to the time derivatives in the governing equations to remove the disparity of wave speeds. The details pertinent to the current work can be found in [3, 4].

1.2.2 Sensitivity Analysis

Gradient-based design methodologies require the gradients of the objective/cost functions (e.g., lift, drag, etc.) and constraints (e.g., flow equations, etc.) with respect to design variables (e.g., Reynolds number, angle of attack, a grid coordinate, etc.). These sensitivity derivatives can be evaluated using finite differences. However, this is not only computationally expensive, accuracy is highly dependent on step size selection, and thus a compromise between reducing the truncation and subtractive cancelation errors must be made. Another approach that has found

success is the complex Taylor's series expansion (CTSE) method [5–7]. The CTSE approach follows from a Taylor's series expansion of a function with respect to a complex perturbation, and is not subject to cancelation errors, and thus step size selection becomes automatic. However, this approach is CPU intensive because of the complex arithmetic involved in the function evaluations. Martins et al. [8] have described an automated way to implement CTSE in existing codes.

The common approach to obtain sensitivity derivatives is the analytic evaluation of these derivatives, referred to as *sensitivity analysis*. Sensitivity analysis can be performed either by *directly* differentiating the governing partial differential equations (PDEs), or by introducing Lagrange multipliers that can be defined by a set of linear equations *adjoint* to the governing PDEs. Both these methods can be categorized further into *continuous* or *discrete* formulations based on the derivation procedure. In the continuous approach, the PDEs are differentiated prior to discretization and the resulting directly differentiated or adjoint equations are discretized and solved. The discrete approach differentiates the PDE after discretization. An advantage of the discrete approach is that the boundary conditions are already incorporated in the formulation, whereas, for the continuous approach, they need to be separately derived and discretized. An excellent overview of these approaches is reported by Newman et al. [9]. Detailed discussions on the continuous approach to sensitivity analysis can be found in the works of Jameson [10], Soemarwoto [11], Anderson et al. [12], Nadarajah et. al [13, 14] and Giles [15].

An excellent introduction to discrete direct and adjoint formulations is given by Hou et al. [16]. More recently, Newman [17] has applied the discrete direct formulation to perform multidisciplinary design optimization (MDO), and Burg and Newman [18] have compared the direct formulation with an efficient CTSE method. Nielsen [19] and Nielsen et al. [20–23] have done extensive work on discrete adjoint methods. Nielsen [19] and Nadarajah [24] have applied the discrete adjoint formulation to perform aerodynamic design optimization. Recently, Giles et al. [25, 26] have proposed an exact dual approach for solving the adjoint system to achieve exact numerical equivalence between the direct and adjoint discretizations. Nielsen et al. [21] have extended the exact dual scheme for implicit solution algorithms and showed identical

asymptotic convergence rates for the primal and dual systems. Multigrid solution for the discrete adjoint problem on unstructured meshes has been reported by Mavriplis [27]. A modification to the current class of unstructured flux limiters has been proposed by Balasubramanian and Newman [28, 29] to make them piecewise continuous and suitable for discrete sensitivity analysis. Nielsen et al. [20] and Balasubramanian et al. [28] have described parallel implementation of the discrete adjoint problem. Recently, Burdyshaw et al. [30] have developed an efficient CTSE-based method to compute adjoint sensitivities in a manner that minimizes maintenance required to reflect subsequent updates of the primal solver.

1.2.3 Adjoint Error Estimates

In engineering applications of CFD, the quantities of interest are often the integral output functions such as force and moment coefficients. An error bound on the output function of interest, or an error correction that delivers a more accurate functional estimate than the overall base solution is often desired. The adjoint (dual) solution describes the sensitivity of the output function to the flow (primal) residuals. By invoking the dual problem, local residual errors resulting from approximation of the solution to the PDEs can be related to the global error in output function via adjoint variables as weight functions. These error estimates can be used as a correction to produce improved functional estimates. The idea of error analysis for output functions using a suitably defined adjoint problem originated in the work of Aubin and Nitsche [31]. Babuška and collaborators [32, 33] were among the first to apply the dual problem in structural analysis for error estimation of point quantities such as displacements and stresses. Becker and Rannacher [34–36] have developed *a posteriori* error estimates for the Navier-Stokes equations based on the dual problem within a finite element framework. More recent discussions on adjoint error analysis for CFD using finite element methods can be found in the research of Süli and collaborators [37, 38], Peraire and collaborators [39, 40] and Giles and Pierce [41–45].

Giles and Pierce [41–44] have developed an adjoint based error correction procedure that exhibits super-convergence properties for functional outputs from finite difference, finite element

or finite volume discretizations. Venditti and Darmofal [46–48] have used an algebraic version of this procedure to estimate the error in the coarse-mesh functional with respect to its value on a globally refined fine-mesh. The auxiliary computations needed by this procedure are: prolongation of coarse-grid flow and adjoint solutions to the fine-mesh; and functional and residual evaluations on the fine-mesh. Park [49, 50] has applied this procedure for estimating functional errors of three dimensional compressible RANS simulations. Balasubramanian [51] has implemented this error correction procedure for two dimensional incompressible flows.

The error estimation procedure require a smooth reconstruction of the primal and dual solutions to compute the error estimates. Giles and Pierce [52–54] have employed a cubic spline interpolation in their research to reconstruct the primal and dual solutions. Venditti and Darmofal [46, 48] have applied a piecewise quadratic prolongation operator, defined by local least squares minimization in the H^1 Sobolev norm. Park [49, 50] has used least squares quadratic interpolation and the prolongation operator of Venditti and Darmofal [46, 48] in his work. Balasubramanian [51] has implemented a finite element bi-quadratic interpolation operator and a reconstruction operator based on least squares to perform the prolongation.

1.2.4 Grid Adaptation

The numerical solution of PDEs governing the flow requires discretization of the continuous flow domain into a finite number of elements or volumes. Two approaches, *structured* and *unstructured*, have evolved over the years to discretize the domain [55, 56]. The structured grids have implied connectivity and are computationally efficient. To handle complex configurations with high curvatures, multi-block structured grids [57] are employed, which are locally structured but, globally unstructured. Unstructured grids provide an alternative to structured grid domain-decomposition methods because of its inherent arbitrariness, and its ability to resolve highly complex geometries efficiently. Another major advantage of unstructured grids is grid adaptation [58–61], as the mesh can be locally enriched where needed, without affecting other regions of the mesh.

Grid adaptive techniques typically employ local error indicators to identify regions that affect the solution accuracy and locally enrich those regions. A common approach is to use error indicators based on the flow gradients or flow discontinuities. Flow regions that have large solution gradients are resolved with more points and regions of minimal significance are coarsened. This will typically lead to refinement of regions that are of physical significance such as boundary layer, shocks, separation lines, stagnation points, etc. [61–68]. This physics-based feature approach sometimes leads to erroneous requests to the grid generator and results in continuous local refinement of certain regions, whereas, globally the adapted grid may not produce the desired results. Moreover, these adaptive indicators may not be rigorous from an engineering context, where the main concern will be the accuracy of the output functions.

Using the adjoint error correction procedure [41–44, 46–48], a grid adaptive strategy may be developed to enhance the accuracy of the chosen output to a prescribed tolerance. The adaptive strategy strives to improve the computable error estimates by forming adaptation parameters/indicators based on the level of error in computable error estimates. Based on this strategy, a grid adaptive scheme can be implemented that takes into account the error in the primal solution, or both the primal and dual solutions. Becker and Rannacher [34–36] have developed this output-based adaptive procedure by exploiting finite element orthogonality properties and duality concepts. Their adaptation parameter included only the error in primal solution. By invoking the dual (adjoint) problem, Süli [37] and Süli and Houston [38] have performed global error control for adaptive finite element approximations of hyperbolic problems. They found computable error bounds (based on error in primal solution) of linear functional to drive the adaptive algorithm. Venditti and Darmofal [48, 69–71] have enhanced this output-based adaptive procedure by including the error in both primal and dual solutions to form the adaptation parameters. They have compared this procedure with a curvature-based adaptive approach and demonstrated its robustness on finite element and finite volume discretizations. Park [49, 50] has employed this output-based adaptive procedure for three dimensional RANS simulations. Peraire and collaborators [39, 40] have incorporated an adaptive procedure based on an implicit *a*

posteriori procedure for computing upper and lower bounds on functional outputs of finite element solutions. Müller and Giles [72] have employed an alternate output-based strategy based on the computable error estimates. In principle, this leads to an adaptive algorithm for minimizing the magnitude of the correction and hence, not a very robust approach. Balasubramanian [51] has compared the adaptive strategies by Venditti and Darmofal [48, 69–71] and Müller and Giles [72].

1.3 Objectives and Approach

The primary goal of this study is to develop an effective and efficient error estimation methodology to provide a quantitative measure of the error in computed outputs and improve the computed accuracy of functional outputs. The error estimation methodology depicted in Fig. 1.1, may be decomposed into three major steps: (1) develop adjoint sensitivity analysis capabilities; (2) develop an error estimation procedure; (3) implement an output-based grid adaptive scheme. In the current work, all three steps have been accomplished as explained below.

The first step is to develop parallel discrete *adjoint* sensitivity analysis capabilities for the arbitrary Mach version of U^2NCLE flow solver [73, 74]. This study also develops *direct* sensitivity capabilities in addition, to compare and validate the implementation of adjoint solver. The accuracy of the derivatives from discrete sensitivity analysis necessitates a consistent and complete linearization of the flow solver. A modification is proposed to make the commonly used unstructured flux limiters (Barth-Jespersen [75] and Venkatakrishnan [76]), piecewise continuous and numerically differentiable, without compromising the monotonicity conditions. The modified limiters are essentially a *weak form* of the original limiters and avoid the numerical instability introduced by the linearization of limiters in their original form. An improved version of Symmetric Gauss-Seidel (SGS) scheme suggested by Whitfield [77] is implemented to solve the linear system of equations from direct formulation. An exact dual algorithm of the improved SGS scheme based on [21, 25, 26] is presented for the adjoint formulation. The parallelization of sensitivity analysis solver is accomplished using a distributed-memory message passing

model. Consistency of linearization is demonstrated in sequential and parallel environments, and validation cases are presented for inviscid, laminar and turbulent flows.

Previous work on three-dimensional adjoint error estimation [49, 50] defined the fine-mesh as embedded within the coarse-mesh, formed by h-refinement of each coarse-mesh element in a fixed ratio, say 1:8. Moreover, all the previous work [49, 50] have been performed on unstructured tetrahedral meshes. The objective of the present study is to extend the adjoint error estimation procedure for a generalized fine-mesh, not necessarily a h-refined embedded mesh. Also, the present study strives to support a mixed-element unstructured mesh, comprising of hexahedrals, prisms, pyramids and tetrahedrals. To accomplish this, the meshless approximation, Moving Least Squares (MLS) [78–83] is chosen to reconstruct the coarse-mesh flow and adjoint solutions to fine-mesh. Linear and quadratic basis functions are considered with cubic spline [82] and inverse-distance weight functions. Weight functions with compact, circular or rectangular supports are defined by isotropic (radial) or anisotropic (tensor product/directional) weights. Mixed supports are also defined; i.e., circular support with isotropic weights for tetrahedrals and pyramids (inviscid regions) and rectangular support with anisotropic weights for prisms and hexahedrals (boundary layer regions).

The final step is to implement an adjoint-based grid adaptive strategy [48, 69–71] to improve the accuracy of the chosen output to a prescribed tolerance. The output-based approach strives to improve the computable error estimates by forming adaptation parameters based on the level of error in both the primal and dual solutions. A feature-based adaptive approach [64–67] is also implemented to compare and demonstrate the robustness and effectiveness of the output-based approach. The feature-based approach identifies significant features in the flow field by using error indicators that represent expansions and compressions in the flow direction and gradients normal to the flow direction. To perform grid adaptation, a general element unstructured mesh adaptor is developed using Objective Oriented (OO) techniques. The adaptive mesh library is based on [84–86] and performs isotropic h-refinement of the elements. A refinement template [87] controls the pattern of subdivision of the mesh elements.

In the current study, the three-dimensional compressible Reynolds-averaged Navier-Stokes equations are solved as described in [4], by introducing preconditioning that is uniformly applicable to Mach numbers ranging from essentially incompressible to supersonic. The one-equation turbulence model of Spalart and Allmaras [88] is used for turbulent flows. The motivation to implement the procedure within an arbitrary Mach framework stems from the desire to handle a wide-range of applications. Applications of this study have been reported in references [28, 29, 89, 90].

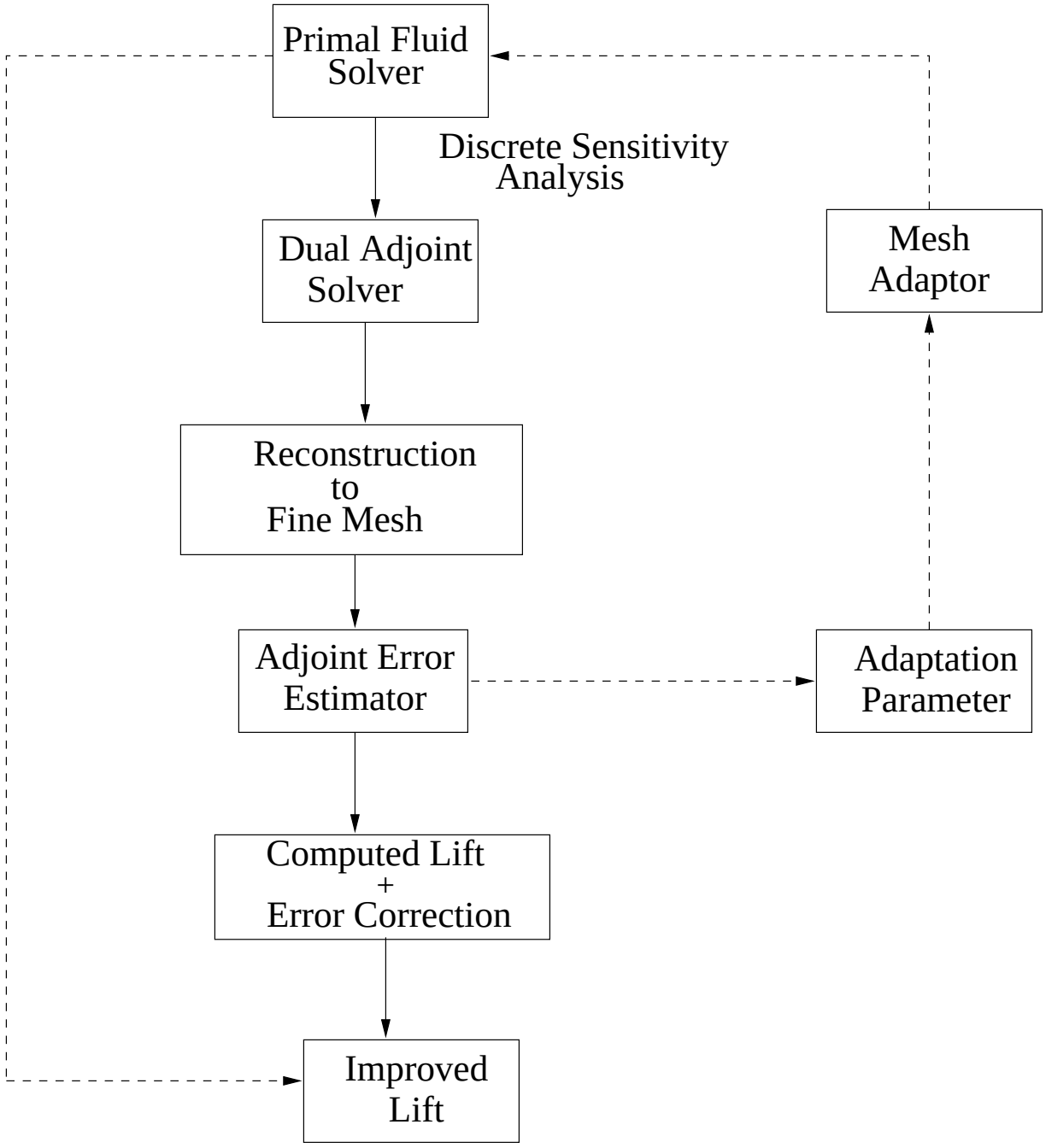


Figure 1.1: Error correction and grid adaptation for functional outputs using discrete adjoint sensitivity analysis.

CHAPTER II

FLUID ANALYSIS

This chapter is intended to give a brief introduction to the baseline flow solver used to develop the present error estimation methodology. The governing equations are introduced, the finite-volume formulation is discussed and the solution algorithm is explained.

2.1 Governing Equations

The Reynolds-averaged Navier-Stokes equations for three-dimensional, variable Mach number flows are used in the present study. The Navier-Stokes equations represent the conservation laws of mass, momentum, and energy. These equations are presented here in a non-rotating Cartesian coordinate system for a finite-volume framework. In the absence of body forces, the non-dimensionalized equations in integral form for a bounded domain Ω, with boundary $\partial\Omega$, can be expressed in primitive variable form [74] as

$$\boldsymbol{M}\boldsymbol{\Gamma}_q^{-1}\frac{\partial}{\partial t}\int_\Omega \boldsymbol{q}\,dV + \oint_{\partial\Omega}\boldsymbol{F}_i\cdot\hat{n}\,dS - \oint_{\partial\Omega}\boldsymbol{F}_v\cdot\hat{n}\,dS = 0 \tag{2.1}$$

where $\hat{n} = \{n_x,\ n_y,\ n_z\}^T$ is the outward pointing unit normal vector to the boundary $\partial\Omega$. The conservative flux formulation is written in terms of primitive variables to facilitate preconditioning. The preconditioning matrix

$$\boldsymbol{\Gamma}_q^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \beta(M_r) \end{bmatrix} = diag\left[1,\ 1,\ 1,\ 1,\ \beta(M_r)\right] \tag{2.2}$$

is a constant diagonal matrix that only depends on the reference Mach number. In Eqn.(2.1), $\boldsymbol{M} = \left[\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{q}}\right]$ is a transformation matrix from conservative variables $\boldsymbol{Q} = \{\rho,\ \rho u,\ \rho v,\ \rho w,\ \rho e_t\}^T$ to primitive variables $\boldsymbol{q} = \{\rho,\ u,\ v,\ w,\ p\}^T$. Here, ρ is the density; u, v, and w are the components of velocity in the x, y, and z directions respectively; p is the pressure; and e_t is the specific total energy. $\boldsymbol{F}_i$ is the inviscid flux vector given by

$$\mathbf{F}_i \cdot \hat{n} \ = \ \begin{bmatrix} \rho\,\Theta \\ \rho\,u\,\Theta \ + \ p\,n_x \\ \rho\,v\,\Theta \ + \ p\,n_y \\ \rho\,w\,\Theta \ + \ p\,n_z \\ \rho\,h_t\,\Theta \end{bmatrix} \tag{2.3}$$

where $h_t = e_t + (\gamma - 1)\,p\,M_r^2$ is the specific total enthalpy. Θ is the normal velocity given by

$$\Theta \ = \ \mathbf{V} \cdot \hat{n} \ = \ u\,n_x + v\,n_y + w\,n_z \tag{2.4}$$

The viscous flux vector $\mathbf{F}_v$ is given by

$$\mathbf{F}_v \cdot \hat{n} \ = \ \begin{bmatrix} 0 \\ \sigma_x \\ \sigma_y \\ \sigma_z \\ u\,\sigma_x + v\,\sigma_y + w\,\sigma_z + Q_n \end{bmatrix} \tag{2.5}$$

where σ_x, σ_y, and σ_z are the viscous shear stresses given as

$$\sigma_x \ = \ \tau_{xx}\,n_x + \tau_{xy}\,n_y + \tau_{xz}\,n_z \tag{2.6}$$

$$\sigma_y \ = \ \tau_{yx}\,n_x + \tau_{yy}\,n_y + \tau_{yz}\,n_z \tag{2.7}$$

$$\sigma_z \;=\; \tau_{zx}\, n_x + \tau_{zy}\, n_y + \tau_{zz}\, n_z \tag{2.8}$$

and

$$\tau_{xx} \;=\; \frac{(\mu+\mu_t)}{Re}\;\frac{2}{3}\;\left(2\,\frac{\partial u}{\partial x} \;-\; \frac{\partial v}{\partial y} \;-\; \frac{\partial w}{\partial z}\right) \tag{2.9}$$

$$\tau_{yy} \;=\; \frac{(\mu+\mu_t)}{Re}\;\frac{2}{3}\;\left(2\,\frac{\partial v}{\partial y} \;-\; \frac{\partial u}{\partial x} \;-\; \frac{\partial w}{\partial z}\right) \tag{2.10}$$

$$\tau_{zz} \;=\; \frac{(\mu+\mu_t)}{Re}\;\frac{2}{3}\;\left(2\,\frac{\partial w}{\partial z} \;-\; \frac{\partial u}{\partial x} \;-\; \frac{\partial v}{\partial y}\right) \tag{2.11}$$

$$\tau_{xy} \;=\; \tau_{yx} \;=\; \frac{(\mu+\mu_t)}{Re}\;\left(\frac{\partial u}{\partial y} \;+\; \frac{\partial v}{\partial x}\right) \tag{2.12}$$

$$\tau_{xz} \;=\; \tau_{zx} \;=\; \frac{(\mu+\mu_t)}{Re}\;\left(\frac{\partial u}{\partial z} \;+\; \frac{\partial w}{\partial x}\right) \tag{2.13}$$

$$\tau_{yz} \;=\; \tau_{zy} \;=\; \frac{(\mu+\mu_t)}{Re}\;\left(\frac{\partial v}{\partial z} \;+\; \frac{\partial w}{\partial y}\right) \tag{2.14}$$

where Re is the Reynolds number based on the reference values. μ_t is the turbulent or eddy viscosity and μ is the molecular viscosity given by Sutherland's law [91] as

$$\mu \;=\; \frac{(1+C^*)\,T^{3/2}}{T+C^*} \tag{2.15}$$

where T is the temperature, $C^* = 198.6/T_r$ is Sutherland's constant and $T_r = 460.0^o R$ is the reference temperature. The heat flux terms are

$$Q_n \;=\; q_x\, n_x \;+\; q_y\, n_y \;+ q_z\, n_z \tag{2.16}$$

and

$$q_x = -\frac{1}{(\gamma-1)\,Re}\left(\frac{\mu}{Pr}+\frac{\mu_t}{Pr_t}\right)\frac{\partial T}{\partial x} \tag{2.17}$$

$$q_y = -\frac{1}{(\gamma-1)\,Re}\left(\frac{\mu}{Pr}+\frac{\mu_t}{Pr_t}\right)\frac{\partial T}{\partial y} \tag{2.18}$$

$$q_z = -\frac{1}{(\gamma-1)\,Re}\left(\frac{\mu}{Pr}+\frac{\mu_t}{Pr_t}\right)\frac{\partial T}{\partial z} \tag{2.19}$$

where $Pr = 0.72$ and $Pr_t = 0.9$ are the Prandtl numbers for the laminar and turbulent parts respectively. The transport equations are closed with the equation of state for a perfect gas (non-dimensional form)

$$p = \frac{\rho\, e_t}{M_r^2} - \frac{(\gamma-1)}{2}\,\rho\,|\mathbf{V}|^2 \tag{2.20}$$

Here, the governing equations have been reduced to the non-dimensional form by the following reference values: density, ρ_r; velocity, U_r; temperature, T_r; length, L_r; pressure, $\rho_r U_r^2$; speed of sound, a_r; time, L_r/U_r; laminar and turbulent viscosity, μ_r; energy and enthalpy, C_pT_r. The reference Mach number is given by $M_r = U_r/a_r$ and $\gamma = C_p/C_v$ is the ratio of specific heats at constant pressure (C_p) and constant volume (C_v).

2.1.1 Preconditioning Parameter

The purpose of preconditioning is to reduce the large disparity between the acoustic and convective parts of the wave speed at low Mach numbers. In Eqn.(2.2), the choice $\beta = 1$ recovers the unpreconditioned formulation. In the present study, β is chosen with values representative of global flow properties as given in [4]

$$\beta\,(M_r) = \begin{cases} {M_r}^2 & ;\ M_r < 1 \\ 1 & ;\ M_r \geq 1 \end{cases} \tag{2.21}$$

where $M_r\,(U_r, T_r) = U_r/\sqrt{\gamma R T_r}$ is the reference Mach number.

2.1.2 Turbulence Model

For the present study, the one-equation turbulence model of Spalart and Allmaras [88] is used. This model formulates a transport equation for the turbulent Reynolds number ($\tilde{\nu}$), which is related to the kinematic turbulent viscosity ($\nu_t = \mu_t/\rho$) by

$$\nu_t = \tilde{\nu}\, f_{v1} \tag{2.22}$$

where $f_{v1} = f_{v1}(\tilde{\nu}, \nu)$ is a function of turbulent Reynolds number and kinematic molecular viscosity $\nu(= \mu/\rho)$. The turbulence equation is non-dimensionalized by the reference values mentioned above and the equation for $\tilde{\nu}$ is given by

$$\frac{\partial \tilde{\nu}}{\partial t} + \mathbf{V}\cdot\nabla\tilde{\nu} = c_{b1}\left[f_{r1} - f_{t2}\right]\tilde{S}\tilde{\nu} - \frac{1}{Re}\left[c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}\right]\left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{1}{\sigma Re}\{\,\nabla\cdot[(\nu + (1+c_{b2})\tilde{\nu})\nabla\tilde{\nu}] - c_{b2}\tilde{\nu}\nabla\cdot[\nabla\tilde{\nu}]\,\} \tag{2.23}$$

where d is the distance to the closest wall and S is the magnitude of vorticity. The function definitions [88, 92] in non-dimensionalized form are

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \qquad \chi = \frac{\tilde{\nu}}{\nu} \tag{2.24}$$

$$\tilde{S} = S + \frac{\tilde{\nu}}{Re\,\kappa^2\,d^2}\,f_{v2}, \qquad f_{v2} = 1 - \frac{\chi}{1 + \chi\, f_{v1}} \tag{2.25}$$

$$f_w = g\left[\frac{1+c_{w3}^6}{g^6 + c_{w3}^6}\right]^{1/6}, \qquad g = r + c_{w2}\left(r^6 - r\right), \qquad r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2 Re} \tag{2.26}$$

$$f_{t2} = c_{t3} exp\left(-c_{t4}\chi^2\right) \tag{2.27}$$

$$f_{r1} = \begin{cases} 1 & : default \text{ [88]} \\ (1+c_{r1})\frac{2r^*}{1+r^*}\left[1 - c_{r3}\tan^{-1}\left(c_{r2}\tilde{r}\right)\right] - c_{r1} & : modified \text{ [92]} \end{cases} \tag{2.28}$$

$$\tilde{r} = \frac{(1-r^*)}{2} \tag{2.29}$$

$$r^* = \sqrt{\frac{1}{2}\frac{S_{ij}}{|S|}} \qquad where\ S_{ij} = \left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right]\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right] \tag{2.30}$$

The constant definitions are as follows:

$$\kappa = 0.41, \qquad \sigma = 2/3, \qquad c_{v1} = 7.1,$$

$$c_{b1} = 0.1355, \qquad c_{b2} = 0.622,$$

$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1+c_{b2}}{\kappa}, \qquad c_{w2} = 0.3, \qquad c_{w3} = 2.0,$$

$$c_{t1} = 1.0, \qquad c_{t2} = 2.0, \qquad c_{t3} = 1.1, \qquad c_{t4} = 2.0$$

$$c_{r1} = 1, \qquad c_{r2} = 12, \qquad c_{r3} = 1$$

2.2 Finite Volume Formulation

Discretization of the governing equations is accomplished using a finite volume technique. The flow domain is divided into a finite number of elements and a dual mesh is constructed by connecting the centroid of elements to the midpoint of edges, to form non-overlapping control volumes around each vertex in the mesh. Equation (2.1) is then integrated over each of these control volumes and solved for the conserved state variables.

The semi-discrete approximation to the spatial terms associated with the control volume surrounding a vertex can be expressed as

$$V\frac{\partial Q}{\partial t} = -R \tag{2.31}$$

where V is the volume of the control volume, Q the cell averaged state variables, and R the residual vector with contributions from the spatial approximation to the inviscid and viscous flux terms.

2.3 Solution Methodology

The flow solver used in the present study is the arbitrary Mach number version of the three-dimensional RANS solver U^2NCLE, described at length in [73, 74]. The solver employs a node-based, finite volume implicit scheme built on an unstructured grid framework capable of handling mixed elements. The solver uses upwind differencing to discretize the convective terms and the modified Roe's flux-difference scheme described in [4, 93] to evaluate the fluxes. The viscous flux terms are evaluated using either the directional-derivative, or the normal-derivative edge-based schemes presented in [73, 74]. For turbulent flows, the one-equation model of Spalart and Allmaras [88] is solved separately in a loosely coupled manner; i.e., the mean flow equations are solved first, followed by the equation for the turbulence model. Temporal discretization is accomplished using a backward-Euler time integration scheme and Newton's method is used to advance the solution in time. The discrete approximation to Eqn.(2.31) can be written as

$$\left[\frac{V}{\triangle t}\bar{I} + \frac{\partial \tilde{R}}{\partial Q}\right] \triangle^n Q = -R \tag{2.32}$$

where $\triangle^n Q = Q^{n+1} - Q^n$. Q^n and Q^{n+1} are the solution vectors at time levels n and $n+1$, $\bar{I}$ is the identity matrix, and $\triangle t$ is the time step. R is the residual vector accounting for the spatial and temporal (in case of unsteady flows) discretizations and $\frac{\partial \tilde{R}}{\partial Q}$ is the residual Jacobian matrix. The flow equations represent a large linear system of equations of the form $Ax = b$. They are solved

using an improved point-iterative Symmetric Gauss-Seidel (SGS) solver as explained below. The original SGS scheme can be written as

forward sweep

$$[D]x_i^k + [U]x_{i+1}^{k-1} - [L]x_{i-1}^k = b_i \tag{2.33}$$

backward sweep

$$[D]x_i^{k+1} + [U]x_{i+1}^{k+1} - [L]x_{i-1}^k = b_i \tag{2.34}$$

where $[D]$, $[L]$, and $[U]$ represent the diagonal, strictly lower-triangular and strictly upper-triangular blocks of $[A]$ and *k* represent the sub-iteration number. In forward sweep, the most recently updated values of x at sub-iteration *k* is used for nodes numbered less than the current node $(i-1 < i)$ and the solution at the previous sub-iteration *k-1* is used for nodes numbered greater than the current node ($i+1 > i$). The procedure is reversed for the backward sweep. By rearranging Eqns.(2.33) and (2.34) as

$$[D]x_i^k + [U]x_{i+1}^{k-1} = b_i + [L]x_{i-1}^k \tag{2.35}$$

$$[D]x_i^{k+1} + [U]x_{i+1}^{k+1} = b_i + [L]x_{i-1}^k \tag{2.36}$$

and recasting them, an improved version of SGS suggested by Whitfield [77] can be obtained

forward sweep

$$[D]x_i^k + [U]x_{i+1}^{k-1} - [L]x_{i-1}^k = b_i \tag{2.37}$$

backward sweep

$$[D]\triangle x_i^k + [U](x_{i+1}^{k+1} - x_{i+1}^{k-1}) = 0 \tag{2.38}$$

where $\triangle x_i^k = x_i^{k+1} - x_i^k$. An extra vector memory is needed for storing the solution vector at iteration level *k-1*. However, one matrix-vector multiplication is avoided compared to the original

SGS backward sweep. A considerable reduction in computational cost is achieved without loss of convergence rate.

CHAPTER III

DISCRETE SENSITIVITY ANALYSIS

This chapter explains the first step in developing the present error estimation methodology: development of adjoint sensitivity analysis capabilities. The discrete sensitivity equations are introduced and the solution methodology is discussed. The linearization issues with the current family of unstructured flux limiters are identified and a modification is proposed to make the limiters suitable for sensitivity analysis. Parallel implementation is elaborated and speedup results are provided. Finally, validation cases are presented to demonstrate the consistency of linearization.

3.1 Discrete Sensitivity Analysis Formulation

Consider the following form of steady state non-linear governing equations, where Q represent the steady state solution vector, X, the grid co-ordinates and α_k, the set of design variables. The discrete residual vector R at steady state is given by

$$R(Q, X, \alpha_k) \equiv R_I(Q, X, \alpha_k) + B(Q, X, \alpha_k) = 0 \tag{3.1}$$

Here, R_I is the discretized residual at the interior and B, the residual at the boundaries (accounting for the boundary conditions). Let $F(Q, X, \alpha_k)$ represent the cost/output function of interest and ∇F, the gradient of the output function with respect to design variables.

3.1.1 Direct Formulation

By directly differentiating Eqn.(3.1) with respect to the vector of design variables, the discrete direct equations are obtained as

$$\frac{dR}{d\alpha_k} = \frac{\partial R}{\partial Q}\,\frac{\partial Q}{\partial \alpha_k} + \frac{\partial R}{\partial X}\,\frac{\partial X}{\partial \alpha_k} + \frac{\partial R}{\partial \alpha_k} = 0 \tag{3.2}$$

Rearranging the above equation in matrix form yields

$$\left[\frac{\partial R}{\partial Q}\right]\left\{\frac{\partial Q}{\partial \alpha_k}\right\} = -\left\{\frac{\partial R}{\partial X}\,\frac{\partial X}{\partial \alpha_k} + \frac{\partial R}{\partial \alpha_k}\right\} \tag{3.3}$$

Here, $\left[\frac{\partial R}{\partial Q}\right]$ is the residual jacobian matrix; $\left\{\frac{\partial Q}{\partial \alpha_k}\right\}$ is the vector of direct sensitivity variables; $\left\{\frac{\partial X}{\partial \alpha_k}\right\}$ and $\left\{\frac{\partial R}{\partial \alpha_k}\right\}$ are the linear sensitivities of the mesh and residue with respect to the design variable α_k. The gradient of the output function using the direct formulation is given by

$$\nabla F = \left\{\frac{\partial F}{\partial Q}\right\}^T \left\{\frac{\partial Q}{\partial \alpha_k}\right\} + \left\{\frac{\partial F}{\partial X}\,\frac{\partial X}{\partial \alpha_k} + \frac{\partial F}{\partial \alpha_k}\right\} \tag{3.4}$$

3.1.2 Adjoint Formulation

For the discrete adjoint formulation, the output function of interest may be augmented with the non-linear discrete flow equations via adjoint variables γ as

$$F(Q, X, \alpha_k) = F(Q, X, \alpha_k) + \{\gamma\}^T R(Q, X, \alpha_k) \tag{3.5}$$

Equation (3.5) results from the fact that for a steady solution the residual vector is zero. Hence, the inner product of any vector with the residual vector must also be zero. Linearizing Eqn.(3.5) yields

$$\begin{aligned}\nabla F = \{\gamma\}^T \left\{\frac{\partial R}{\partial X}\,\frac{\partial X}{\partial \alpha_k} + \frac{\partial R}{\partial \alpha_k}\right\} + \left\{\frac{\partial F}{\partial X}\,\frac{\partial X}{\partial \alpha_k} + \frac{\partial F}{\partial \alpha_k}\right\} \\ + \left(\left\{\frac{\partial F}{\partial Q}\right\}^T + \{\gamma\}^T\left[\frac{\partial R}{\partial Q}\right]\right)\left\{\frac{dQ}{d\alpha_k}\right\}\end{aligned} \tag{3.6}$$

Since $\{\gamma\}$ is yet to be defined, it may be chosen to force the coefficients of $\left\{\frac{\partial Q}{\partial \alpha_k}\right\}$ to be zero. The equation for adjoint variables is then

$$\{\gamma\}^T \left[\frac{\partial R}{\partial Q}\right] + \left\{\frac{\partial F}{\partial Q}\right\}^T = 0 \tag{3.7}$$

By taking the transpose of the above equation, the adjoint variable co-state vectors are given by

$$\left[\frac{\partial R}{\partial Q}\right]^T \{\gamma\} = -\left\{\frac{\partial F}{\partial Q}\right\} \tag{3.8}$$

and the gradient of the output function can be obtained as

$$\nabla F = \{\gamma\}^T \left\{\frac{\partial R}{\partial X}\ \frac{\partial X}{\partial \alpha_k} + \frac{\partial R}{\partial \alpha_k}\right\} + \left\{\frac{\partial F}{\partial X}\ \frac{\partial X}{\partial \alpha_k} + \frac{\partial F}{\partial \alpha_k}\right\} \tag{3.9}$$

3.1.3 Direct Vs Adjoint Formulation

The selection of an appropriate sensitivity analysis formulation (direct or adjoint) depends on the particular design problem being studied. In the direct method, a linear system must be solved for derivatives with respect to each design variable. For the adjoint formulation, the number of linear systems that must be solved scales with the number of augmented objective/cost functions. Thus, if the number of design variables are more compared to the number of objective functions, the adjoint formulation is preferred and vice versa.

3.1.4 Objective Functions

The objective functions or cost functions (from an optimization point of view) are mostly the integral outputs of CFD calculations that are relevant to engineering applications. For example, these outputs can be the force and moment coefficients on an aircraft. The objective functions that are considered for the present study are the lift C_L and drag C_D coefficients. The expressions for C_L and C_D are

$$C_L = \frac{Lift}{q_\infty S} = \frac{f_y\, cos\bar{\alpha} - f_x\, sin\bar{\alpha}}{q_\infty S} \tag{3.10}$$

$$C_D = \frac{Lift}{q_\infty S} = \frac{f_x\, cos\bar{\alpha} + f_y\, sin\bar{\alpha}}{q_\infty S} \tag{3.11}$$

where $\bar{\alpha}$ is the freestream angle of attack; $q_\infty = \frac{1}{2}\rho_r {U_r}^2$ is the dynamic pressure; and S is the reference area and for aircraft wings, S can be the wing area. f_x and f_y are the forces in the x and y directions given by

$$f_x = \oint_{\partial\Omega} (p + sf)\vec{n}_x\, dA \tag{3.12}$$

$$f_y = \oint_{\partial\Omega} (p + sf)\vec{n}_y\, dA \tag{3.13}$$

where p is the pressure; sf is viscous stress due to skin friction; $\vec{n}_x$, $\vec{n}_y$ are the boundary normals in the x and y directions; and dA is the surface area of the boundary.

3.2 Solution Methodology

The discrete sensitivity equations for the fluid equations and turbulence model are solved in a loosely coupled manner much like the flow solver; i.e., the sensitivity equations for the mean flow are solved first, followed by the loosely coupled equation for the turbulence model. The loosely coupled implementation of the turbulence model in the flow solver sometimes result in stalled convergence or limit-cycle oscillations that is detrimental to the convergence of sensitivity analysis solver. This behavior has also been reported by Nielsen et al. [21]. This issue needs further investigation and should be addressed in future research.

Although the sensitivity Eqns.(3.3) and (3.8) for the direct and adjoint formulations can be solved as such, it is more robust to solve them by adding a time-derivative term and recast in an incremental iterative form [94–96]. The recast equations can be written as

$$\left[\frac{V}{\triangle t}\bar{I} + \frac{\partial \tilde{R}}{\partial Q}\right] \triangle^n \left(\frac{\partial Q}{\partial \alpha_k}\right) = -\left\{\frac{\partial R}{\partial X}\,\frac{\partial X}{\partial \alpha_k} + \frac{\partial R}{\partial \alpha_k}\right\} - \left[\frac{\partial R}{\partial Q}\right]_{exact} \left(\frac{\partial Q}{\partial \alpha_k}\right)^n \tag{3.14}$$

$$\left[\frac{V}{\triangle t}\bar{I} + \frac{\partial \tilde{R}}{\partial Q}\right]^T \triangle^n \gamma = -\frac{\partial F}{\partial Q} - \left[\frac{\partial R}{\partial Q}\right]^T_{exact} \gamma^n \tag{3.15}$$

where $\triangle^n \left(\frac{\partial Q}{\partial \alpha_k}\right) = \left(\frac{\partial Q}{\partial \alpha_k}\right)^{n+1} - \left(\frac{\partial Q}{\partial \alpha_k}\right)^n$ and $\triangle^n \gamma = \gamma^{n+1} - \gamma^n$. The time term makes the equations diagonally dominant and allows the solution to be obtained in a time-marching manner much like the flow solver. It has been found that an approximate or first-order Jacobian works well for the left-hand side. However, no approximations may be made to the Jacobian on the right-hand side. Note that, the block-Jacobian matrix on the right-hand side is exact for turbulent flows, even though the mean flow and turbulent sensitivity equations are solved in a loosely coupled manner; i.e. each block is a 5×6 matrix for the mean flow, and a 1×6 matrix for the turbulence model to account for the five flow variables and one turbulent quantity.

The direct and adjoint sensitivity Eqns.(3.14) and (3.15) represent a large linear system of equations that can be cast as $Ax = b$. The direct equations are solved using the improved Symmetric Gauss-Seidel (SGS) solver explained in section 2.3.

Recently, Giles et al. [25, 26] proposed an exact dual approach for solving the adjoint system to achieve exact numerical equivalence between the direct and adjoint discretizations. Nielsen et al. [21] extended the exact dual scheme for implicit solution algorithms and showed asymptotically equivalent convergence rates for the primal and dual systems. An exact dual algorithm of the improved SGS scheme based on [21, 25, 26] is presented here. By writing

$$[A]^T = (\,[D] + [U] - [L]\,)^T = [D]^T + [U]^T - [L]^T \equiv [\bar{D}] + [\bar{L}] - [\bar{U}] \tag{3.16}$$

where $[\bar{L}] = [U]^T$ and $[\bar{U}] = [L]^T$ are the strictly lower-triangular and strictly upper-triangular parts of the matrix A^T, an exact dual form of Eqns.(2.37) and (2.38) for the adjoint system can be obtained

backward sweep

$$[\bar{D}]x_i^k - [\bar{U}]x_{i+1}^k + [\bar{L}]x_{i-1}^{k-1} = b_i \tag{3.17}$$

forward sweep

$$[\bar{D}]\triangle x_i^k + [\bar{L}](x_{i-1}^{k+1} - x_{i-1}^{k-1}) = 0 \tag{3.18}$$

In the above exact dual scheme, backward sweep is done first followed by forward sweep.

3.3 Flux Limiters

For higher-order upwind differencing, a Taylor series expansion of the state variables (Q) about the node (n) in Fig. 3.1, gives the limited data on the face as

$$Q_L \equiv (Q_{f_1})_L = Q_n + \Phi_n \, \nabla Q_n \cdot \vec{r}_1 \tag{3.19}$$

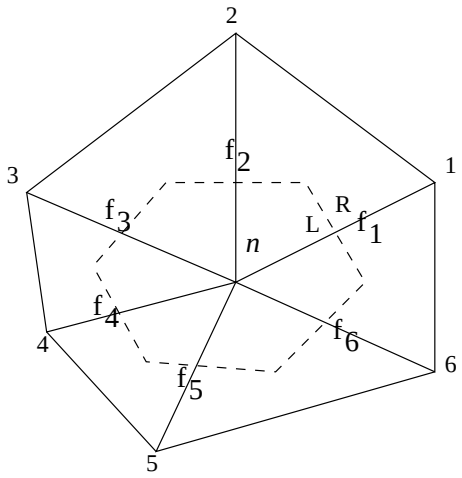


Figure 3.1: Control volume surrounding vertex n formed by centroid-midpoint dual in 2D.

where ∇Q_n is the gradient at n, $\vec{r}_1$ is the distance vector from n to a point on the control volume face f_1, and Q_L is the reconstructed solution at the face. The flux limiter value at n [75, 76] is given by

$$\Phi_n = min\{\Phi_{f_j}, \; f_j = f_1, f_2, \cdots, f_6\} \tag{3.20}$$

where Φ_{f_j} are the limited values at the control volume faces evaluated using Barth-Jespersen [75] or Venkatakrishnan [76] limiter. The limited face value is obtained by

$$\Phi_{f_j} = \begin{cases} min\left(1, \phi\left(\frac{{Q_n}^{max} - Q_n}{Q_{f_j} - Q_n}\right)\right) & , if \quad Q_{f_j} - Q_n > 0 \\ min\left(1, \phi\left(\frac{{Q_n}^{min} - Q_n}{Q_{f_j} - Q_n}\right)\right) & , if \quad Q_{f_j} - Q_n < 0 \\ 1 & , if \quad Q_{f_j} - Q_n = 0 \end{cases} \tag{3.21}$$

Barth-Jespersen:

$$\phi\left(\frac{a}{b}\right) = \frac{a}{b} \tag{3.22}$$

Venkatakrishnan:

$$\phi\left(\frac{a}{b}\right) = \frac{1}{b}\left[\frac{(a^2 + \epsilon^2)b + 2ab^2}{a^2 + 2b^2 + ab + \epsilon^2}\right] \tag{3.23}$$

where Q_n^{max}, Q_n^{min} and ϵ are as given in [75] and [76]. Both these limiters are widely used in unstructured flow solvers to ensure no new extrema are created in the reconstruction process.

Differentiating Eqn.(3.19) with respect to solution vector Q yields

$$dQ_L \equiv (dQ_{f_1})_L = dQ_n + d\Phi_n \, \nabla Q_n \cdot \vec{r}_1 + \Phi_n \, d\left(\nabla Q_n \cdot \vec{r}_1\right) \tag{3.24}$$

Φ_n, given by Eqn.(3.20) can be numerically differentiated and $d\Phi_n \in \{d\Phi_{f_1}, d\Phi_{f_2}, \cdots, d\Phi_{f_6}\}$. However, use of non-differentiable functions such as the *min* function to obtain Φ_n introduces numerical scaling issues when the linearization is performed. To illustrate the scaling issues, consider the term $\{d\Phi_n \, \nabla Q_n \cdot \vec{r}_1\}$ in Eqn.(3.24). Suppose, $\Phi_n = \Phi_{f_2}$, then $d\Phi_n$ can be written as

$$d\Phi_n = d\Phi_{f_2} = \frac{dQ_n^{max/min} - dQ_n}{\nabla Q_n \cdot \vec{r}_2} - \frac{d(\nabla Q_n) \cdot \vec{r}_2}{(\nabla Q_n \cdot \vec{r}_2)^2} \tag{3.25}$$

and

$$d\Phi_n \, \nabla Q_n \cdot \vec{r}_1 = \left(\frac{dQ_n^{max/min} - dQ_n}{\nabla Q_n \cdot \vec{r}_2} - \frac{d(\nabla Q_n) \cdot \vec{r}_2}{(\nabla Q_n \cdot \vec{r}_2)^2}\right)(\nabla Q_n \cdot \vec{r}_1) \tag{3.26}$$

In unstructured meshes, the distance vectors $\vec{r}_1$ and $\vec{r}_2$ need not be of the same order of magnitude, leading to instances where $(\nabla Q_n \cdot \vec{r}_2) << (\nabla Q_n \cdot \vec{r}_1)$. In such cases, the numerator and denominator of the terms in Eqn.(3.26) may be of different orders of magnitude. For example, if $\nabla Q_n \cdot \vec{r}_1 \sim O(10^{-1})$; $\nabla Q_n \cdot \vec{r}_2 \sim O(10^{-3})$; and $d(\nabla Q_n) \cdot \vec{r}_2 \sim O(10^{-3})$; then

$$d\Phi_n \, \nabla Q_n \cdot \vec{r}_1 \; \sim \; \left(\frac{O(1)}{O(10^{-3})} + \frac{O(10^{-3})}{O(10^{-6})} \right) O(10^{-1}) \; \sim \; O(100) \tag{3.27}$$

The linearization for limiters in this form introduces numerical instability if used with sensitivity analysis. One option may be to use limiters in flow solver, and freeze them (assume as constant with derivatives $d\Phi_n = 0$) when performing sensitivity analysis. This approach is not a consistent linearization and may produce inaccurate sensitivity derivatives.

A modification is proposed to make the limiters piecewise continuous and numerically differentiable, without compromising the monotonicity conditions. Instead of using Φ_n to reconstruct the solution to the faces, each face is reconstructed with its own limited value Φ_f as shown below

$$Q_{f_j} = Q_n + \Phi_{f_j} \, \nabla Q_n \cdot \vec{r}_j \tag{3.28}$$

and Φ_{f_j} are computed the same way as given in Eqns.(3.21), (3.22) and (3.23). This modification makes the limiters piecewise continuous and also avoids the numerical instability introduced by the linearization of limiters in their original form. Since, the base schemes to compute Φ_{f_j} have not changed, the modified limiters satisfy the monotonicity criteria ensuring no new extrema creation. The modified limiters are essentially a *weak form* of the original limiters.

To illustrate the need for the modified form of limiters, consider the case of inviscid flow over a Onera-M6 wing at an angle of attack of 3.06^0 and Mach number of 0.84. A lambda shock is typical of these flow conditions and the solution is reconstructed using Eqn.(3.28) with the modified form of Venkatakrishnan limiter. Sensitivity analysis is performed in two ways: (1) with the limiters linearized accounting for a consistent and complete linearization; (2) with the limiters frozen (assumed as constant with derivatives $d\Phi_f = 0$) accounting for an incomplete linearization. Table

Table 3.1: Comparison of sensitivity derivatives with complete and incomplete linearization of limiters

Method	dC_L/dy	dC_D/dy
Central Finite Difference		
$1.e^{-5}$	$-5.24676e^{-5}$	$9.4514e^{-6}$
$1.e^{-6}$	$-5.2390e^{-5}$	$9.4795e^{-6}$
$1.e^{-8}$	$-5.1850e^{-5}$	$9.450e^{-6}$
adjoint (complete linearization)	$-5.16418e^{-5}$	$9.3635e^{-6}$
adjoint (incomplete linearization)	$-4.92201e^{-5}$	$1.20769e^{-5}$

(3.1) compares the adjoint sensitivity derivatives from complete and incomplete linearizations with central finite differences. The table shows the derivatives of lift (C_L) and drag (C_D) coefficients with respect to y coordinate of a node on the wing. From Table (3.1), it can be observed that the adjoint sensitivities for C_L and C_D from complete linearization match with finite differences upto 5 and 6 decimal places respectively. However, there is atleast one significant digit loss in accuracy in the derivatives obtained from the incomplete linearization of limiters. This loss of accuracy illustrates the need for complete linearization of limiters, which can not be accomplished if they are implemented in their original form. The author believes if limiters have to be used in conjunction with direct or adjoint sensitivity analysis, they must be employed in their modified/weak-form for a consistent and complete linearization.

A grid resolution study is also performed for the above inviscid case to study the behavior of the modified limiters. Three levels of grid, ranging from coarse to relatively fine discretizations are chosen, and numerical tests are performed employing the modified and original limiters. Similar behavior is observed on all grids and results are presented here for the intermediate grid level. Fig. 3.2 shows the convergence history of the modified and original limiters. The modified limiters are more dispersive because of their weak form, and show better convergence behavior when compared with the original limiters. The C_p distribution on the upper and lower surfaces of the Onera-M6 wing at three span wise locations are plotted in Figs. 3.3, 3.4 and 3.5. These figures compare the C_p distribution obtained using the original and weak-form of the limiters with

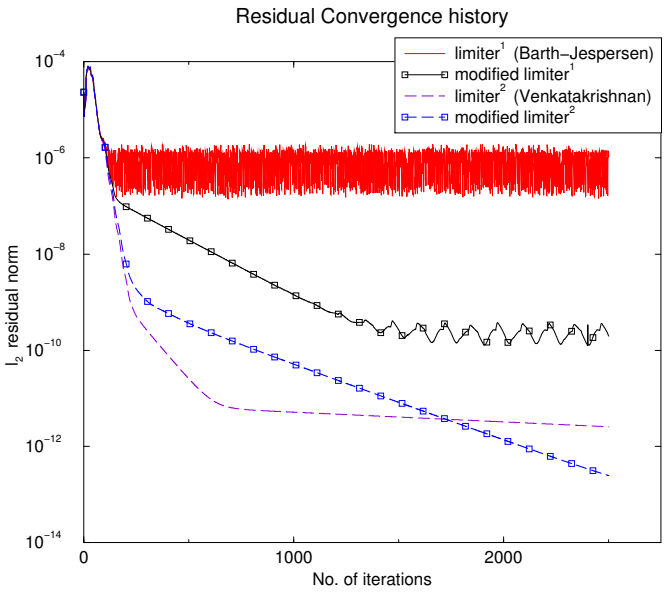


Figure 3.2: L_2 norm convergence of flow residual employing Barth-Jespersen ($limiter^1$) or Venkatakrishnan ($limiter^2$) limiter.

experimental data obtained from [97]. The lambda shock that is typical of these flow conditions is captured well.

3.4 Parallel Implementation

The parallel methodology is based on a distributed memory message passing model and employs coarse-grained domain decomposition with sub-domains assigned to multiple processors. Message passing is achieved using MPI-based libraries and the mesh partitioner MeTiS [98, 99] is used to subdivide the mesh into sub-domains. Since, the baseline flow solver [73] could run in a multiprocessor environment when this work was started, the existing MPI-based libraries are utilized to develop parallel capabilities for the direct and adjoint solvers.

Because of the gradient terms used in the reconstruction process, a complete linearization of the higher-order spatial terms require neighbor's information and neighbor's neighbor information. Fig. 3.6 shows the complete linearization stencil for the higher-order spatial discretization of node i. For nodes in the partition boundary, this requirement involves two levels of ghost nodes as shown in Fig. 3.6. The flow solver typically needs only level-1 ghost nodes and they are included

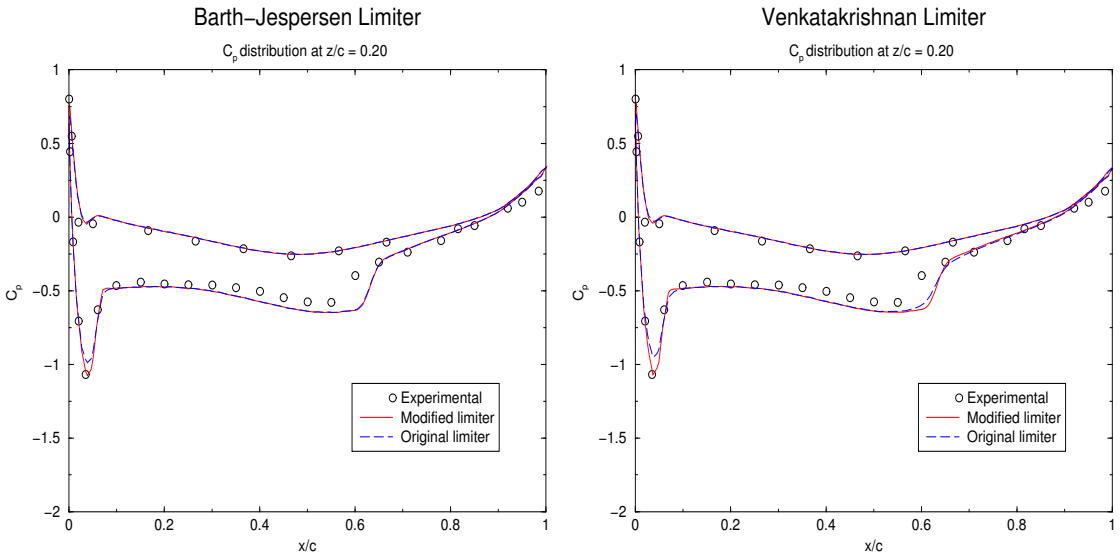


Figure 3.3: C_p distribution on the Onera-M6 wing at span wise locations $z/c = 0.20$.

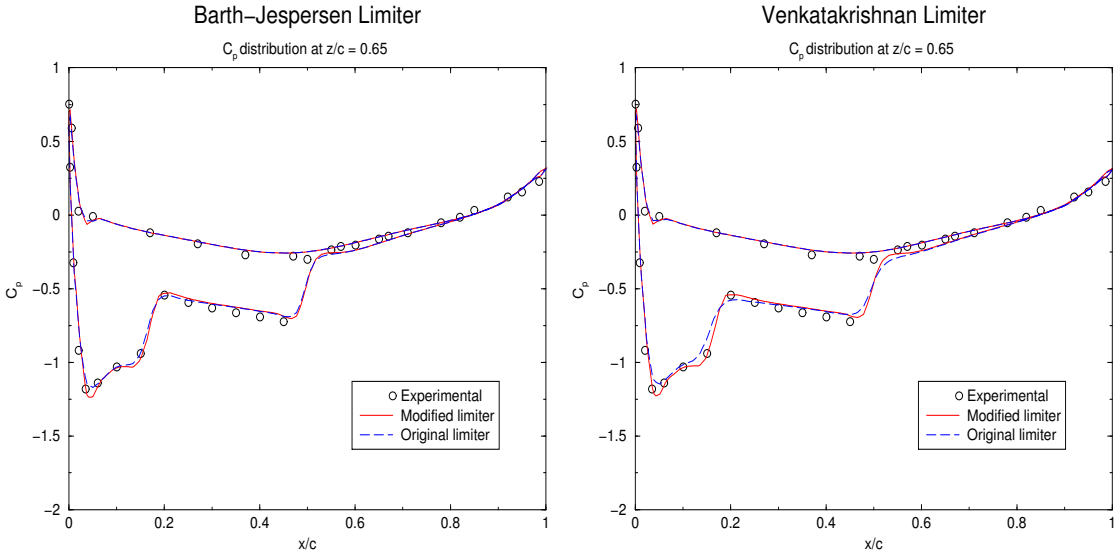


Figure 3.4: C_p distribution on the Onera-M6 wing at span wise locations $z/c = 0.65$.

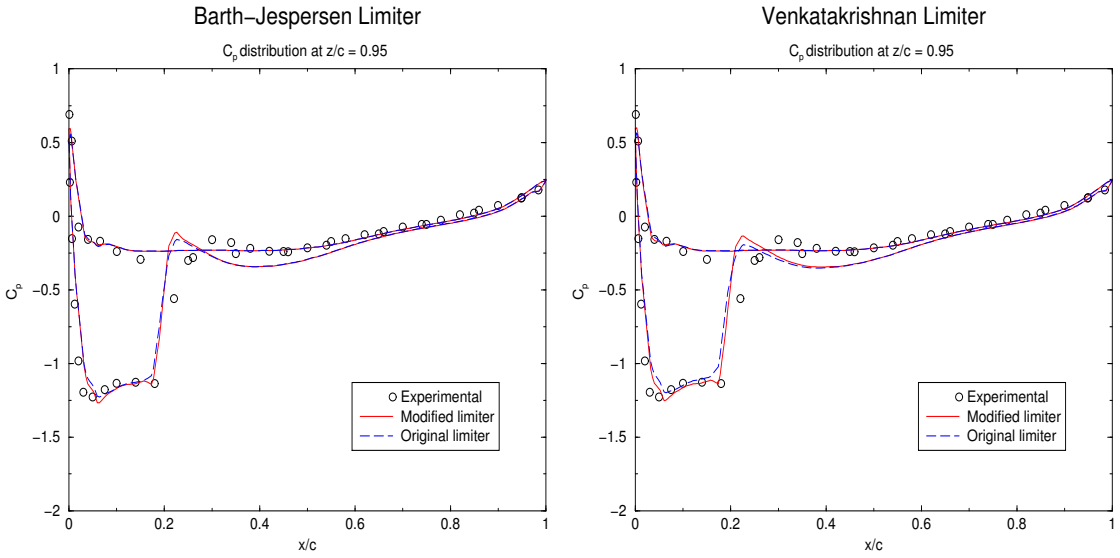


Figure 3.5: C_p distribution on the Onera-M6 wing at span wise locations $z/c = 0.95$.

in the partition. The information from level-2 ghost nodes that are not included in the partition must be accounted for in sensitivity analysis to perform a complete linearization. The parallel implementation of the sensitivity analysis solver is described below for nodes in the partition boundary.

- Only level-1 ghost nodes are included in the partition.
- Data in level-1 ghost nodes are used to compute nearest neighbor residual contributions and gathered by the physical nodes in each block (similar to the flow solver). In Fig. 3.6, node i is a physical node in **block 1**. It uses the information from physical nodes 3, 4 and level-1 ghost nodes 1, 2, 5 to gather nearest neighbor residual contributions.
- Residual contributions from level-2 ghost nodes (nearest neighbor's neighbors that are not included in the partition) are gathered by the respective level-1 ghost nodes in corresponding block. In Fig. 3.6, node i is a level-1 ghost node in **block-2**. Nodes 7, 8, 9, 10 are level-2 ghost nodes for **block-1**, but, are physical nodes in **block 2**. Residual contributions from nodes 7, 8, 9, 10 are gathered by level-1 ghost node i in **block-2**.

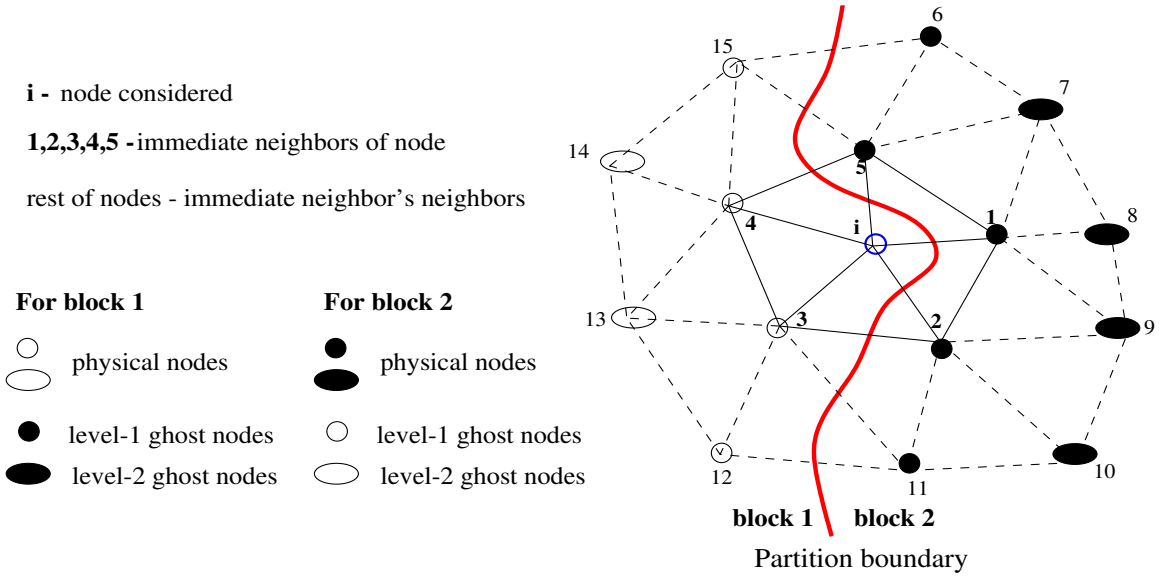


Figure 3.6: Stencil for the complete linearization of higher-order spatial discretization for node i.

- Swap the send and receive lists. Each block has a list of send lists and receive lists. Send lists have the list of physical nodes that must share it data with neighboring blocks and receive lists have the list of ghost nodes that need to update its data. By swapping the send and receive lists, the new send lists will have list of ghost nodes that hold the scattered residual contributions and the new receive lists will have the list of physical nodes that need to gather this data.

- Data in ghost nodes is communicated to neighboring blocks and gathered by the physical nodes to perform a consistent and complete linearization.

The speedup results for the sensitivity analysis solvers are demonstrated in Figs.(3.7a) and (3.7b). It can be observed that a nearly linear speedup has been obtained. The computations for the test case is performed on a 384 processor Intel Xeon cluster. For this test, turbulent flow is simulated over the Onera M6 wing at an angle of attack of 1^0, with freestream Mach number of 0.52, and chord-based Reynolds number of $1,000,000$. For illustrative purposes, the mixed-element grid for the turbulent flow simulation is shown in Fig.(3.8). This grid contains 232,003 prisms, 10882 pyramids, 319,016 tetrahedrals, 178,193 nodes, 31402 surface triangles and 1787

surface quadrilaterals with a wall spacing of $1e^{-05}$ of mean aerodynamic chord. The linearization results are presented in next section.

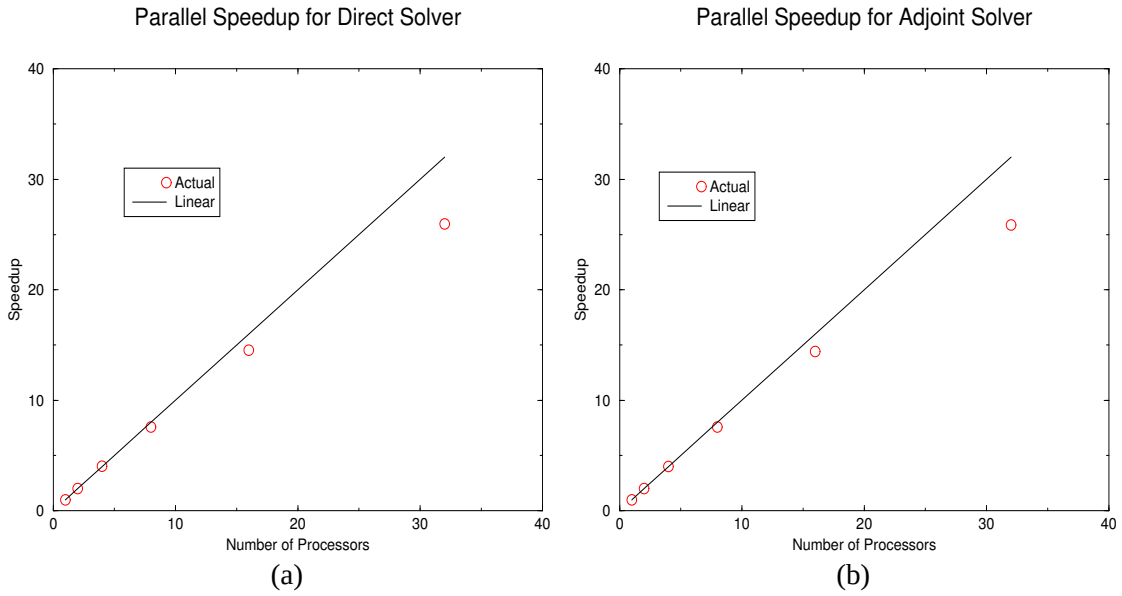


Figure 3.7: Parallel speedup for turbulent flow over a Onera M6 wing. (a): Direct solver. (b): Adjoint solver.

3.5 Validation Cases

Linearizations are performed analytically by hand-differentiation for the viscous flux terms and by the complex Taylor's series expansion approach [5–7] for the inviscid terms. The consistency of linearization is validated by performing a series of test cases with the Onera M6 wing. The design or independent variable for all validation cases is the *y* co-ordinate of a node on the wing. Sensitivity of the lift (C_L) and drag (C_D) coefficients from direct and adjoint analysis are compared with the derivatives evaluated using finite differences. In all finite difference results, the flow solution and force coefficients have been converged to machine precision. Similar convergence behavior is observed on all variables and for sake of clarity, only convergence plots of direct and adjoint densities are presented here.

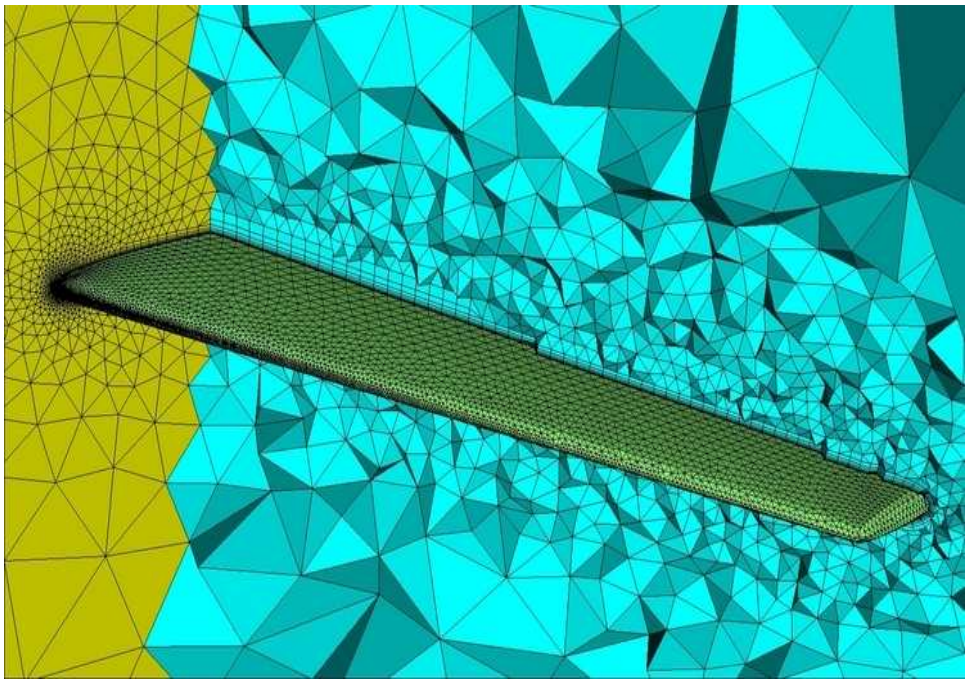


Figure 3.8: Volume grid for Onera M6 wing.

3.5.1 Inviscid Flow

The first test case is inviscid flow at an angle of attack of 2^0 with freestream Mach number of 0.84. The unstructured grid contains 65538 nodes and 360,035 tetrahedrals. The convergence history of direct and adjoint densities are plotted in Fig. 3.9a. Asymptotically equivalent convergence rates are observed. The sensitivity derivative of the drag coefficient (C_D) is compared with central finite difference derivatives in Table (3.2). The finite difference derivatives are obtained by using different perturbation sizes. There is perfect agreement between the direct and adjoint sensitivities and they match with finite differences upto six decimal places. From Table (3.2), it can be inferred that the direct and adjoint sensitivities are consistent over sequential and parallel runs.

3.5.2 Viscous Laminar Flow

Two viscous laminar cases are studied with chord-based Reynolds number of 5000 and angle of attack of 2^0. The first case is transonic flow with freestream Mach number of 0.84 and the

Table 3.2: Comparison of sensitivity derivatives for inviscid flow

Method	dC_D/dy
Central Finite Difference	
$1.e^{-4}$	$-7.7161e^{-6}$
$1.e^{-6}$	$-7.6925e^{-6}$
$1.e^{-8}$	$-7.6500e^{-6}$
direct (sequential)	$-7.5936e^{-6}$
direct (parallel)	$-7.5936e^{-6}$
adjoint (sequential)	$-7.5936e^{-6}$
adjoint (parallel)	$-7.5936e^{-6}$

second case is a low Mach number flow of 0.1. The flow is simulated on a mixed-element grid with 76376 prisms, 6800 pyramids, 277,849 tetrahedrals, 93473 nodes, 31392 surface triangles and 500 surface quadrilaterals. Similar asymptotic rates are observed in the convergence of direct and adjoint solutions in Fig. 3.9b. Tables (3.3) and (3.4) compare the sensitivity derivatives of lift and drag coefficients with finite differences. The direct and adjoint sensitivities are in excellent agreement with each other and compare favorably with finite differences, given the errors inherent in finite difference approximation. It can be verified from Tables (3.3) and (3.4) that once again the derivatives are consistent across sequential and parallel environments.

Table 3.3: Comparison of sensitivity derivatives for viscous laminar flow

Method	dC_L/dy	dC_D/dy
Central Finite Difference		
$1.e^{-4}$	$-5.69920e^{-5}$	$1.73546e^{-5}$
$1.e^{-6}$	$-5.69365e^{-5}$	$1.71380e^{-5}$
$1.e^{-8}$	$-5.70e^{-5}$	$1.7150e^{-5}$
$1.e^{-10}$	$-5.50e^{-5}$	$2.0e^{-5}$
direct (sequential)	$-5.69358e^{-5}$	$1.71382e^{-5}$
direct (parallel)	$-5.69358e^{-5}$	$1.71382e^{-5}$
adjoint (sequential)	$-5.69358e^{-5}$	$1.71382e^{-5}$
adjoint (parallel)	$-5.69358e^{-5}$	$1.71382e^{-5}$

Table 3.4: Comparison of sensitivity derivatives for low Mach number laminar flow

Method	dC_L/dy
Central Finite Difference	
$1.e^{-6}$	$-1.686295e^{-4}$
$1.e^{-8}$	$-1.680500e^{-4}$
direct (sequential)	$-1.686252e^{-4}$
direct (parallel)	$-1.686252e^{-4}$
adjoint (sequential)	$-1.686252e^{-4}$
adjoint (parallel)	$-1.686252e^{-4}$

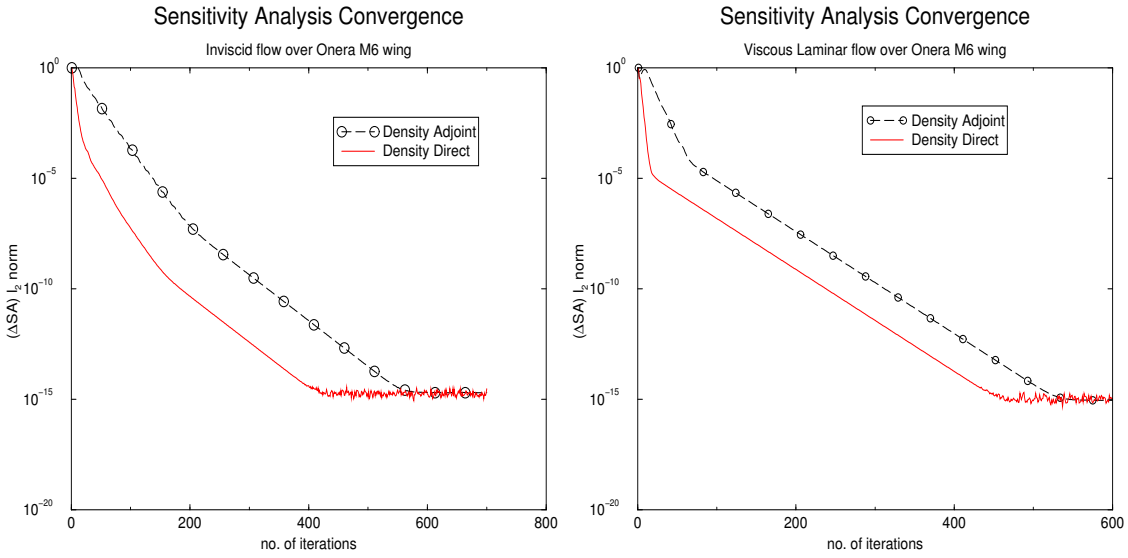
3.5.3 Turbulent Flow

The final validation case is turbulent flow over the Onera M6 wing at an angle of attack of 1^0, with freestream Mach number of 0.52, and chord-based Reynolds number of $1,000,000$. The mixed-element grid for this simulation is shown in Fig.(3.8). The one-equation turbulence model of Spalart and Allmaras [88] is solved in a loosely coupled manner in both flow and sensitivity analysis.

The sensitivity derivatives of lift and drag coefficients are compared with finite differences in Table (3.5). The direct and adjoint sensitivities are in excellent agreement with each other and compare favorably with finite differences. From Table (3.5), it can be verified that the linearizations are consistent over sequential and parallel versions. Fig.(3.10a) shows the convergence histories of density and turbulent quantity for the direct and adjoint solutions. The asymptotic rates are similar. To further demonstrate the numerical equivalence between the direct and adjoint solvers, the error in drag derivatives is plotted in Fig.(3.10b). The error is defined as the difference between the current value and the final converged value. The reduction in error rates are identical.

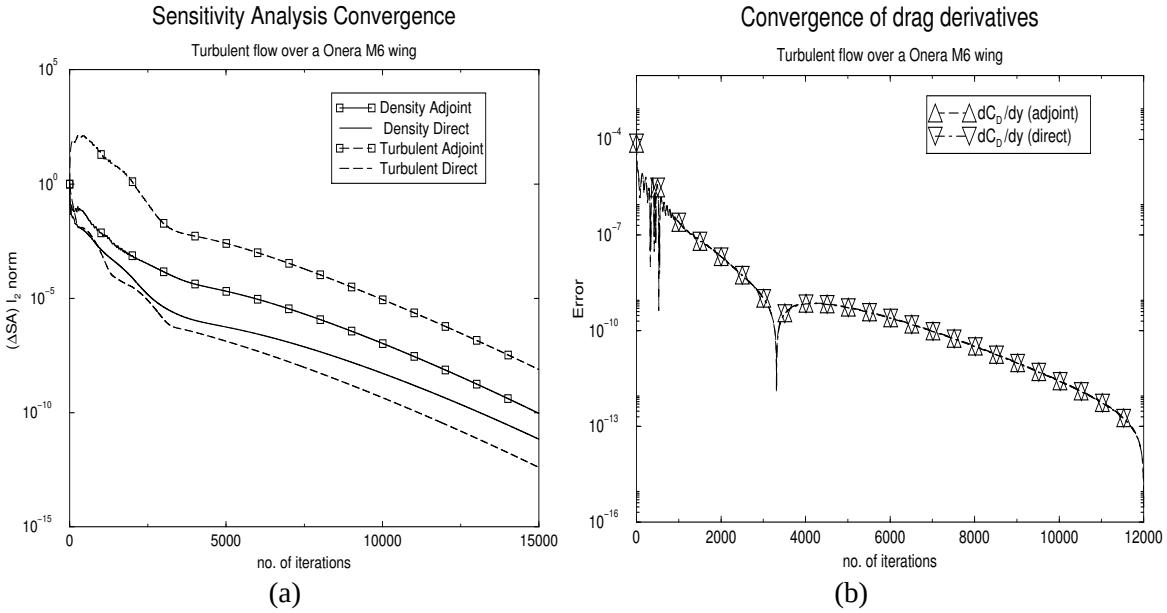
Table 3.5: Comparison of sensitivity derivatives for turbulent flow

Method	dC_L/dy	dC_D/dy
Central Finite Difference		
$1.e^{-6}$	$1.09190e^{-5}$	$-2.77890e^{-5}$
$1.e^{-8}$	$1.20500e^{-5}$	$-2.7800e^{-5}$
direct (sequential)	$1.11464e^{-5}$	$-2.78552e^{-5}$
direct (parallel)	$1.11464e^{-5}$	$-2.78552e^{-5}$
adjoint (sequential)	$1.11464e^{-5}$	$-2.78552e^{-5}$
adjoint (parallel)	$1.11464e^{-5}$	$-2.78552e^{-5}$



(a) (b)

Figure 3.9: L_2 norm convergence of $\triangle \frac{\partial Q}{\partial \alpha_k}$ and $\triangle \gamma$. (a): Inviscid flow. (b): Viscous laminar flow.



(a) (b)

Figure 3.10: Turbulent flow over a Onera M6 wing at $AOA = 1^0$ with $M_\infty = 0.52$ and $Re = 1,000,000$. (a): L_2 norm convergence of $\triangle \frac{\partial Q}{\partial \alpha_k}$ and $\triangle \gamma$. (b): Convergence of drag derivatives.

CHAPTER IV

ERROR ESTIMATION

This chapter explains the second step in the present error estimation methodology: development of an efficient and robust error correction procedure. The adjoint-based error correction procedure is derived and the prolongation operators are introduced. Error correction results are presented for inviscid, laminar and turbulent cases.

4.1 Formulation

For the following discussions, let $F(Q)$ represent the output function of interest; Q, the steady-state solution vector; $R(Q)$, the discrete residual vector; and γ, the adjoint solution vector. Typically, in engineering applications of CFD, an accurate estimate of $F(Q)$ is desired. But often, a compromise must be made between the fidelity of solution obtained and the available resources. To elaborate on this, consider discretization of the computational domain (Ω) using a coarse-mesh Ω_H and a fine-mesh Ω_h. H and h ($H > h$) may represent suitably defined length scales based on the approximation such as finite difference, finite element or finite volume. Let $F_H(Q_H)$ and $F_h(Q_h)$ be estimates of $F(Q)$ from Ω_H and Ω_h. $F_H(Q_H)$ and $F_h(Q_h)$ are evaluated using Q_H and Q_h, the discrete solutions on Ω_H and Ω_h respectively. The coarse-mesh Ω_H is affordable in terms of memory and computation time. However, the estimate $F_H(Q_H)$ may not be accurate enough for engineering applications. The fine-mesh estimate $F_h(Q_h)$ may satisfy the desired accuracy criteria, but is prohibitively expensive to compute. A computationally efficient error correction procedure [41–44, 46–48] is introduced that produce improved estimates of output functions without ever solving on the fine-mesh Ω_h.

By introducing a perturbation δQ_h to the fine-mesh solution Q_h, and performing a Taylor's series expansion of $F_h(Q_h + \delta Q_h)$ yields

$$F_h(Q_h + \delta Q_h) \;=\; F_h(Q_h) \;+\; \left\{\frac{\partial F_h}{\partial Q_h}\right\} \delta Q_h \;+\; \cdots \tag{4.1}$$

Now, defining $\delta Q_h \;=\; Q_H^h - Q_h$, where Q_H^h is the solution at fine-mesh obtained through prolongation of coarse-mesh solution Q_H, Eqn.(4.1) can be written as

$$F_h(Q_h) \;=\; F_h(Q_H^h) \;+\; \left\{\frac{\partial F_h}{\partial Q_h}\right\}_{Q_H^h} (Q_h - Q_H^h) \;+\; \cdots \tag{4.2}$$

Here, $F_h(Q_H^h)$ is the fine-mesh estimate of the function evaluated using Q_H^h and $\left\{\frac{\partial F_h}{\partial Q_h}\right\}_{Q_H^h}$ is the linear sensitivities of the fine-mesh function with respect to Q_H^h. Q_H^h is given by

$$Q_H^h \;=\; P_H^h \, Q_H \tag{4.3}$$

and P_H^h is a suitably defined prolongation operator.

Let $R_h(Q_h)$ be the non-linear residual vector obtained by discretization of the flow equations at the fine-mesh. For a steady state problem

$$R_h(Q_h) \;=\; 0 \tag{4.4}$$

By performing a Taylor's series expansion of $R_h(Q_h + \delta Q_h)$

$$R_h(Q_h + \delta Q_h) \;=\; R_h(Q_h) \;+\; \left[\frac{\partial R_h}{\partial Q_h}\right] \delta Q_h \;+\; \cdots \tag{4.5}$$

and substituting $\delta Q_h = Q_H^h - Q_h$

$$R_h(Q_h) \;=\; R_h(Q_H^h) \;+\; \left[\frac{\partial R_h}{\partial Q_h}\right]_{Q_H^h} (Q_h - Q_H^h) \;+\; \cdots \tag{4.6}$$

Here, $R_h(Q_H^h)$ is the residual vector evaluated at the fine-mesh using Q_H^h and $\left[\frac{\partial R_h}{\partial Q_h}\right]_{Q_H^h}$ is the fine-mesh Jacobian evaluated using Q_H^h. Combining equations (4.4 and 4.6) and assuming the well-posedness of the equations

$$(Q_h - Q_H^h) \;\approx\; -\left[\frac{\partial R_h}{\partial Q_h}\right]^{-1}_{Q_H^h} R_h(Q_H^h) \tag{4.7}$$

Substituting Eqn.(4.7) in Eqn.(4.2)

$$F_h(Q_h) \;\approx\; F_h(Q_H^h) \;-\; \left\{\frac{\partial F_h}{\partial Q_h}\right\}^T_{Q_H^h} \left[\frac{\partial R_h}{\partial Q_h}\right]^{-1}_{Q_H^h} R_h(Q_H^h) \tag{4.8}$$

$$\approx\; F_h(Q_H^h) \;+\; \{\gamma_h\}^T_{Q_H^h} R_h(Q_H^h) \tag{4.9}$$

where $\{\gamma_h\}^T_{Q_H^h}$ is the adjoint solution vector at the fine-mesh evaluated using Q_H^h. The adjoint equation for $\{\gamma_h\}_{Q_H^h}$ can be written as

$$\left[\frac{\partial R_h}{\partial Q_h}\right]^T_{Q_H^h} \{\gamma_h\}_{Q_H^h} \;=\; -\left\{\frac{\partial F_h}{\partial Q_h}\right\}_{Q_H^h} \tag{4.10}$$

To avoid the need for computing $\{\gamma_h\}^T_{Q_H^h}$ on the fine-mesh, it may also be approximated through interpolation as

$$\{\gamma_h\}_{Q_H^h} \;\approx\; \gamma_H^h \;=\; P_H^h\, \gamma_H \tag{4.11}$$

where γ_H is the adjoint solution at the coarse-mesh given by

$$\left[\frac{\partial R_H}{\partial Q_H}\right]^T \{\gamma_H\} \;=\; -\left\{\frac{\partial F_H}{\partial Q_H}\right\} \tag{4.12}$$

The computable estimate of the output function is given by

$$F(Q) \;=\; F_h(Q_H^h) \;+\; \{\gamma_H^h\}^T\, R_h(Q_H^h) \tag{4.13}$$

In the above expression, $\{\gamma_H^h\}^T R_h(Q_H^h)$ is the *error correction* term that approximates the error in output function as the inner product of the adjoint solution and the primal residual error. The auxiliary computations needed by this procedure are: prolongation of coarse-mesh flow and adjoint solutions to the fine-mesh; and functional and residual evaluations on the fine-mesh.

4.2 Enhancements to the Error Correction Procedure

Previous work on three-dimensional adjoint error estimation [49, 50] defined the fine-mesh Ω_h as embedded within the coarse-mesh, formed by h-refinement of each coarse-mesh element in a fixed ratio, say 1:8. Moreover, all the previous work [49, 50] have been performed on unstructured tetrahedral meshes. The present study extends the adjoint error estimation procedure for a generalized fine-mesh, not necessarily a h-refined embedded mesh. Also, the present study strives to support a mixed-element unstructured mesh, comprising of hexahedrals, prisms, pyramids and tetrahedrals. For subsequent discussions, the fine-mesh can be categorized into

Uniformly Refined: fine-mesh obtained using h-refinement; each fine-mesh element is embedded within a coarse-mesh element.

Non-uniformly Refined: fine-mesh obtained from a grid generator by manually setting the point spacing; fine-mesh element need not be embedded within a coarse-mesh element.

The isotropic h-refinement of different element types is shown in Appendix B. In three dimensions, uniform refinement of a coarse-mesh often results in a fine-mesh that has approximately 8 times the number of coarse-mesh elements. In the present study, the error correction procedure is sequential (done in a single processor) and hence, uniformly refined meshes become prohibitively expensive in terms of memory and computational cost. Also, successive refinements result in meshes with very poor quality, especially in the boundary layer, and projection of boundary nodes on the surface geometry raises additional complexity. Non-uniformly refined meshes generated from a grid generator have better quality than uniformly refined meshes. Also, they maintain boundary integrity between the coarse and fine-meshes, since, the same CAD definition will be used to generate the meshes. Moreover, the user will

know *a priori* the available memory and computational resources and hence, the level of fine-mesh refinement can be modified accordingly to make use of the available resources.

In the present study, the error estimation procedure is applied to both uniformly and non-uniformly refined meshes. The non-uniform fine-meshes are generated using AFLR3 [100] and the uniformly refined meshes are generated using the mesh adaptor discussed in section 5.3. To handle a general mesh (both uniform and non-uniform), the meshless Moving Least Squares (MLS) approximation [78–83] is chosen to reconstruct the coarse-mesh flow and adjoint solutions to fine-mesh. The MLS procedure is explained in Appendix A.

4.3 Prolongation Operators

The prolongation operation given by Eqn.(4.3) can be expanded as

$$Q_H^h \;=\; P_H^h\, Q_H \equiv \sum_{I=1}^{n} \phi_I^k \left(Q_H\right)_I \tag{4.14}$$

where $\Phi^k = \{\phi_1^k, \phi_2^k, \cdots, \phi_n^k\}$ are the MLS shape functions, k is the order of the basis function, and n is size of the MLS support stencil. In the present study, linear and quadratic basis functions given in Eqns.(A.2) and (A.3) are considered with cubic spline and inverse-distance [82] weights. Weight functions are defined with compact circular or rectangular supports as explained in Appendix A. Once the supports are built, weights are applied, either isotropically as radial weights, or anisotropically as tensor product weights. Mixed supports are also defined; i.e., circular support with isotropic weights for tetrahedrals and pyramids (inviscid regions) and rectangular support with anisotropic weights for prisms and hexahedrals (boundary layer regions). In the the present study, the basis functions are classified as:

linear-linear: prolongation performed with linear basis for both flow and adjoint solutions.

quadratic-quadratic: prolongation performed with quadratic basis for both flow and adjoint solutions.

linear-quadratic: prolongation performed with linear basis for flow solution and quadratic basis for adjoint solution.

quadratic-linear: prolongation performed with quadratic basis for flow solution and linear basis for adjoint solution.

The weight functions are classified as:

cubic spline: cubic spline weights given by Eqn.(A.21) in Appendix A.

inverse-distance: inverse-distance weights given by Eqn.(A.22) in Appendix A.

and the weights are applied as:

isotropic: radial weights defined by circular supports.

anisotropic: tensor product weights defined by rectangular supports.

mixed: both radial and tensor product weights defined by circular-rectangular supports.

For tetrahedral meshes the stencil is built using circular supports. For mixed-element meshes, circular supports are used to build the stencil in regions of tetrahedrals and pyramids, and rectangular supports are used in regions of prisms and hexahedrals. Once the stencil is built, both isotropic and anisotropic weights are applied. Note that, even if the stencil is built with circular support, anisotropic weights are applied with the directional lengths defined by the absolute values of the position vector from the data point to the MLS seed points. In other words, a rectangular support is constructed using the maximum directional lengths in the circular support. This is equivalent to reconstructing the support to a rectangular, with the circular support as background. The points that make the stencil remain the same, except the weights are applied differently. The same procedure applies to isotropic weights in a rectangular support, with the directional lengths substituted by the radial distances.

4.3.1 Strong Boundary Conditions

In finite volume solvers, the noslip conditions are enforced at viscous boundaries by setting $u_b = 0$, $v_b = 0$ and $w_b = 0$, where u_b, v_b and w_b are the x, y and z components of velocity at the boundary. This is often referred as a weak-enforcement of the no-slip condition. However, the common practice in most node-based finite volume solvers is to additionally set the momentum and energy fluxes in Eqns.(2.3) and (2.5) to zero. Instead of solving for the conservation of momentum and energy at the viscous boundary nodes, they are explicitly set to zero in a strong-enforcement of the no-slip condition. Similarly, in turbulent flows, the turbulent residual and turbulent quantity ($\bar{\nu}$) are explicitly set to zero at the viscous boundary nodes.

To account for this strong-enforcement at the viscous boundary, the adjoint system [25, 26] for an interior node needs to be explicitly modified with all viscous boundary nodes removed from the momentum and energy linearization. So, when prolongation is performed for the interior nodes near viscous boundaries, the boundary nodes should not be included in the stencil because interpolation will be between nodes with different dual properties. In references [48, 50, 71], the boundary adjoint solution is replaced with extrapolated interior values and the prolongation is performed using the extrapolated boundary values. The boundary adjoint is then post processed as explained in [48]. In the present study, the MLS fit for the velocities and turbulent quantity are performed without any viscous boundary nodes in the stencil. However, for adjoint density and pressure, the MLS fit included viscous boundary nodes. Note that, the present study employs the adiabatic boundary condition $\frac{\partial T(\rho,p)}{\partial n} = 0$, based on both interior and boundary temperatures ($T = T(\rho, p)$).

4.3.2 Parameter Definitions for Comparison of Prolongation Operators

The following parameters are defined to make meaningful comparisons of the different prolongation operators.

Quality of MLS fit: The MLS procedure produces the *best possible* approximation for a given data point based on a least squares fit of the seed points in the support stencil. The quality of the fit is largely dependent on the support stencil. This is extremely important in boundary layer

regions, where the presence of high-aspect ratio elements may produce an ill-conditioned MLS system, degrading the quality of the fit. To measure the approximation error in the MLS fit and thereby, get an idea about the quality of the support stencil, a simple test is performed by finding the approximation for the seed points in the stencil. The error in the MLS approximation can be defined as

$$\text{MLS fit error} = u_h(x_I) - u(x_I) \tag{4.15}$$

where $u_h(x_I)$ is the approximation from MLS fit and $u(x_I)$ is the actual value.

% True Error: The actual functional error is defined as

$$\%\,\text{True Error} = 1.0 - \frac{\text{computed output at coarse-mesh}}{\text{computed output at fine-mesh}} \times 100 \tag{4.16}$$

% Error after correction: The remaining error in the corrected functional is defined as

$$\%\,\text{Error after correction} = 1.0 - \frac{\text{error corrected output at coarse-mesh}}{\text{computed output at fine-mesh}} \times 100 \tag{4.17}$$

Parallel Cost: In the present study, all the flow and adjoint computations are done in parallel. To make uniform comparisons of CPU time between different runs, the parallel CPU/run time is linearly scaled based on the number of processors to represent the overall parallel cost. The parallel cost measured in CPU hours is computed as

$$\text{Parallel cost} = \text{(Actual Parallel CPU time)*(Number of processors)} \tag{4.18}$$

4.4 Results and Discussions

Error correction results are presented here for inviscid, laminar and turbulent test cases. The prolongation operators are compared based on the choice of (a) basis functions: linear-linear, quadratic-quadratic, linear-quadratic or quadratic-linear; (b) weight functions: cubic spline or inverse-distance; and (c) type of support (application of weights): isotropic, anisotropic or mixed. For the cases, when the prolongation operator is compared over a series of coarse, intermediate

and fine meshes, true error^1 is defined between the coarsest and intermediate mesh and true error^2 is defined between the intermediate and finest mesh.

4.4.1 Inviscid flow

The first test case is inviscid flow over an Onera M6 wing at an angle of attack of 3.06^0 and Mach number of 0.8395. Error correction is performed for both lift (C_L) and drag (C_D) coefficients on the wing. Three non-uniformly refined meshes are generated with the coarsest mesh containing $42,114$ nodes, $23,422$ surface triangles and $213,889$ tetrahedrals. The intermediate and finest meshes are generated by reducing the point spacings in the boundary by a factor of two and four with respect to the initial coarse-mesh spacing. The intermediate and finest grids contain respectively, $183,796$ nodes, $76,152$ surface triangles and $976,344$ tetrahedrals; and $976,503$ nodes, $286,728$ surface triangles and $5,372,918$ tetrahedrals. The coarsest mesh is uniformly refined to construct an embedded mesh containing $310,119$ nodes, $93,688$ surface triangles and $1,712,372$ tetrahedrals. The MLS support stencil is built with circular supports and when tensor product (anisotropic) weights are applied, the supports are reconstructed to rectangular from the circular support.

4.4.1.1 C_D in a Onera M6 wing at $M_\infty = 0.8395$ and $AOA = 3.06^0$

The error correction results for C_D are presented in Tables (4.1) to (4.10). Tables (4.1) and (4.2) compare the different MLS fits for prolongation to a uniformly refined mesh using cubic spline and inverse-distance weights respectively. From Tables (4.1) and (4.2), it can be observed that the corrections from all the MLS fits reduce the true error by 50%. The isotropic and anisotropic application of weights produce nearly identical corrections. For the linear-linear and quadratic-quadratic MLS fits, the corrections from cubic spline and inverse-distance weight functions are of the same order of magnitude. However, the cubic spline linear-quadratic and quadratic-linear MLS fits compute better corrections compared to their inverse-distance counterparts. The best estimates of correction are observed in cubic spline linear-quadratic and quadratic-linear MLS fits. The error corrected C_D from the cubic spline linear-quadratic

and quadratic-linear MLS fits are within 1% and 5% of the fine-mesh estimate. One possible explanation for the better correction from a combined MLS fit (linear-quadratic or quadratic-linear) may be a bad flow or adjoint approximation in the individual MLS fits (linear-linear or quadratic-quadratic) and the bad approximation is not employed in the combined fit. Another possible explanation may be the leading truncation error (TE) terms in the linear and quadratic approximations are of different signs and result in a smaller leading TE term and hence, a more accurate correction for the combined approximation.

The behavior of the error estimates is studied by performing the error correction procedure over a sequence of non-uniformly refined meshes. As the mesh resolution increases, the functional estimates get more accurate and this can be observed in the reduction in true error from -40.57% between the coarse and intermediate meshes to -9.93% between the intermediate and fine meshes in Table (4.3). Similar behavior should be observed in the error estimates, with the correction getting smaller as the functional output converges to its asymptotic value. This behavior can be observed in Tables (4.3) to (4.10) for the non-uniformly refined meshes. The tables demonstrate the decrease in error correction estimates as the prolongation is performed between better resolved grids with smaller true error. The weights are applied isotropically and it can be noticed from the tables that the inverse-distance corrections are slightly better than the cubic spline corrections. For all the MLS fits, the true error^1 and true error^2 are reduced by more than 50% after applying correction.

The quadratic-linear MLS fits given in Tables (4.9) and (4.10) consistently produce the best correction estimates. The corrected C_D from the coarse and intermediate-meshes are within 1% of their respective fine-mesh estimates; i.e. the true error^1 and true error^2 are reduced by 99% after correction. The role, the leading TE terms play in a combined approximation (linear-quadratic or quadratic-linear) can be better appreciated by looking at Tables (4.7) to (4.10) for the linear-quadratic and quadratic-linear MLS fits. For the linear-quadratic MLS fits, the corrections at the intermediate-mesh are worse compared to the individual corrections (linear-linear or quadratic-quadratic), whereas, the quadratic-linear corrections are better than the individual corrections.

This behavior may be best explained by the signs of the leading TE (truncation error) terms in the linear and quadratic approximations. If they are of the same sign, they add up resulting in a larger TE term and a less accurate correction as observed in the linear-quadratic corrections for the intermediate-mesh. If they are of opposite signs, they cancel out resulting in a smaller leading TE term and a more accurate correction as seen in the quadratic-linear corrections for the intermediate-mesh.

Fig.(4.1) shows the parallel CPU costs for the error corrected C_D. From Fig.(4.1), it can be inferred that significant savings in parallel computational cost can be obtained by performing error correction. The L_2 norm error in the MLS fit for the flow and adjoint solutions on the coarsest mesh is plotted in Figs.(4.2a - 4.2d). As expected, the error in the inverse-distance MLS fit is smaller compared to the cubic spline MLS fit. The inverse-distance weights have a large penalty factor that forces the fit to start interpolating the data as the distance between the data and seed points approaches zero. However, the cubic MLS fit produces a smoother approximation and the corrections in Tables (4.1) to (4.10) are almost identical from both these approximations. The L_2 norm error for the adjoint fit is larger compared to the flow fit. Also, the L_2 norm error in the linear and quadratic approximations are of the same order.

Table 4.1: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Cubic spline weights correction from a coarse-mesh with $C_D = 0.02250563$ to a uniformly refined mesh with $C_D = 0.01510187$; True Error = -49.02%.

MLS fit	Error Correction	Corrected C_D	$\%$ Error after correction
$Linear - Linear(isotropic)$	-0.00468865	0.01781698	-17.97
$Linear - Linear(anisotropic)$	-0.00390798	0.01859765	-23.14
$Quadratic - Quadratic(isotropic)$	-0.00407856	0.01842707	-22.01
$Quadratic - Quadratic(anisotropic)$	-0.00366597	0.01883966	-24.75
$Linear - Quadratic(isotropic)$	-0.00749459	0.01501104	0.601
$Linear - Quadratic(anisotropic)$	-0.00747593	0.01502970	0.478
$Quadratic - Linear(isotropic)$	-0.00666770	0.01583793	-4.873
$Quadratic - Linear(anisotropic)$	-0.00689221	0.01561342	-3.387

Table 4.2: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Inverse-distance weights correction from a coarse-mesh with $C_D = 0.02250563$ to a uniformly refined mesh with $C_D = 0.01510187$; True Error = -49.02%.

MLS fit	Error Correction	Corrected C_D	$\%$ Error after correction
$Linear - Linear(isotropic)$	-0.00451507	0.01799056	-19.12
$Linear - Linear(anisotropic)$	-0.00396752	0.01853811	-22.75
$Quadratic - Quadratic(isotropic)$	-0.00439652	0.01810911	-19.91
$Quadratic - Quadratic(anisotropic)$	-0.00392247	0.01858316	-23.05
$Linear - Quadratic(isotropic)$	-0.00524878	0.01725685	-14.26
$Linear - Quadratic(anisotropic)$	-0.00425534	0.01825029	-20.84
$Quadratic - Linear(isotropic)$	-0.00557292	0.01693273	-12.12
$Quadratic - Linear(anisotropic)$	-0.00530304	0.01720259	-13.91

Table 4.3: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Isotropic cubic spline weights correction for C_D with linear-linear basis on non-uniformly refined meshes; True Error^1 = -40.57%, True Error^2 = -9.93%.

Nodes	C_D	Error Correction	Corrected C_D	$\%$ Error after correction
42114	0.02250563	-0.00413585	0.01836978	-14.74
183796	0.01601001	-0.00153230	0.01447771	0.588
976503	0.01456340	$-$	$-$	$-$

Table 4.4: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Isotropic inverse-distance weights correction for C_D with linear-linear basis on non-uniformly refined meshes; True Error^1 = -40.57%, True Error^2 = -9.93%.

Nodes	C_D	Error Correction	Corrected C_D	$\%$ Error after correction
42114	0.02250563	-0.00471712	0.01778851	-11.10
183796	0.01601001	-0.00166997	0.01434004	1.53
976503	0.01456340	$-$	$-$	$-$

Table 4.5: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Isotropic cubic spline weights correction for C_D with quadratic-quadratic basis on non-uniformly refined meshes; True Error^1 = -40.57%, True Error^2 = -9.93%.

Nodes	C_D	Error Correction	Corrected C_D	$\%$ Error after correction
42114	0.02250563	-0.00384564	0.01865999	-17.08
183796	0.01601001	-0.00127405	0.01473596	-1.184
976503	0.01456340	–	–	–

Table 4.6: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Isotropic inverse-distance weights correction for C_D with quadratic-quadratic basis on non-uniformly refined meshes; True Error^1 = -40.57%, True Error^2 = -9.93%.

Nodes	C_D	Error Correction	Corrected C_D	$\%$ Error after correction
42114	0.02250563	-0.00413646	0.01836917	-14.73
183796	0.01601001	-0.00135075	0.01465926	-0.658
976503	0.01456340	–	–	–

Table 4.7: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Isotropic cubic spline weights correction for C_D with linear-quadratic basis on non-uniformly refined meshes; True Error^1 = -40.57%, True Error^2 = -9.93%.

Nodes	C_D	Error Correction	Corrected C_D	$\%$ Error after correction
42114	0.02250563	-0.00672852	0.01577711	1.454
183796	0.01601001	-0.00175355	0.01425646	2.107
976503	0.01456340	–	–	–

Table 4.8: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Isotropic inverse-distance weights correction for C_D with linear-quadratic basis on non-uniformly refined meshes; True Error^1 = -40.57%, True Error^2 = -9.93%.

Nodes	C_D	Error Correction	Corrected C_D	$\%$ Error after correction
42114	0.02250563	-0.00556712	0.01693851	-0.580
183796	0.01601001	-0.00189285	0.01411716	3.064
976503	0.01456340	–	–	–

Table 4.9: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Isotropic cubic spline weights correction for C_D with quadratic-linear basis on non-uniformly refined meshes; True Error^1 = -40.57%, True Error^2 = -9.93%.

Nodes	C_D	Error Correction	Corrected C_D	% Error after correction
42114	0.02250563	-0.00637202	0.01613361	-0.772
183796	0.01601001	-0.00139431	0.01461570	-0.359
976503	0.01456340	–	–	–

Table 4.10: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Isotropic inverse-distance weights correction for C_D with quadratic-linear basis on non-uniformly refined meshes; True Error^1 = -40.57%, True Error^2 = -9.93%.

Nodes	C_D	Error Correction	Corrected C_D	% Error after correction
42114	0.02250563	-0.00566662	0.01683901	-0.517
183796	0.01601001	-0.00143605	0.01457396	-0.0725
976503	0.01456340	–	–	–

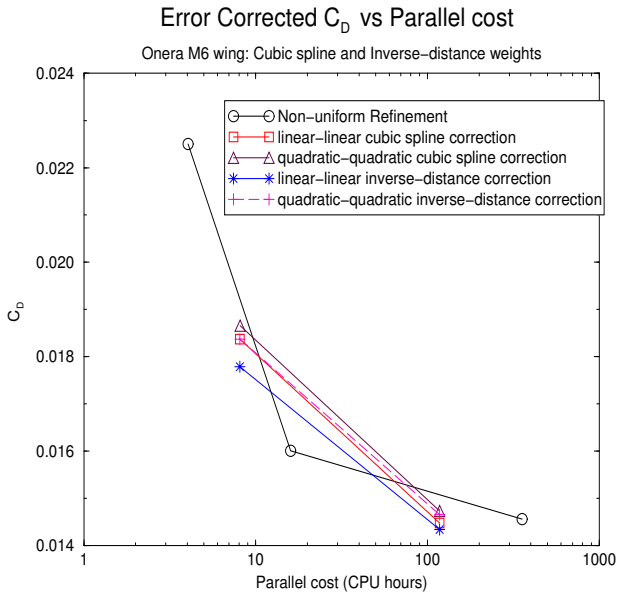


Figure 4.1: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Parallel cost for the error corrected C_D.

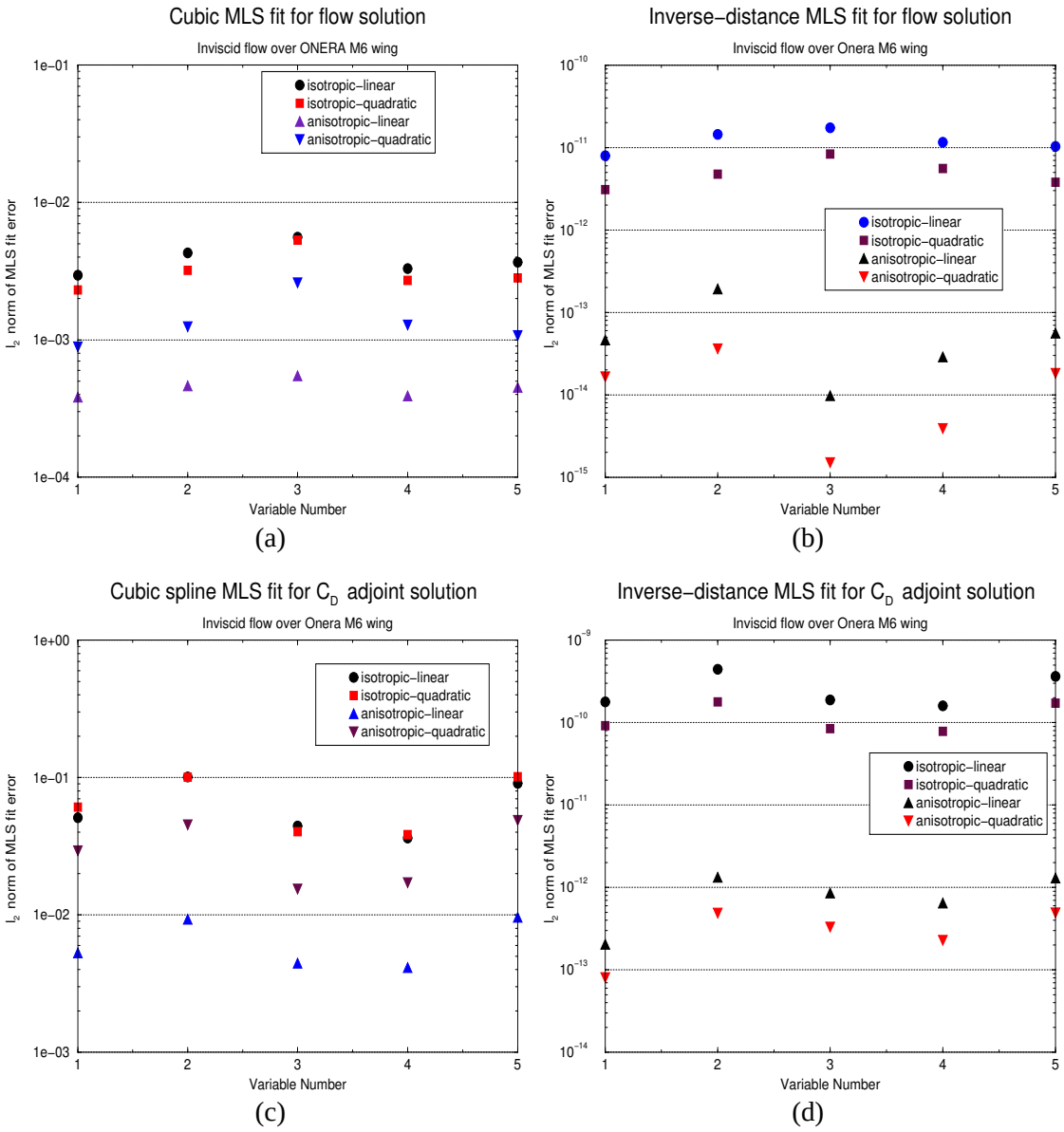


Figure 4.2: Inviscid flow over Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. L_2 norm of error in MLS fit of individual variables $1 = \rho$, $2 = u$, $3 = v$,$4 = w$,$5 = p$. (a) Cubic spline fit for flow solution; (b) Inverse-distance fit for flow solution; (c) Cubic spline fit for adjoint solution; (d) Inverse-distance fit for adjoint solution.

4.4.1.2 C_L in a Onera M6 wing at $M_\infty = 0.8395$ and $AOA = 3.06^0$

The error correction summary for prolongation to a uniformly refined fine-mesh is presented in Tables (4.11) and (4.12). All the MLS fits perform as expected, reducing the true error by 50% to 60%. The best corrections are observed in the cubic spline and inverse-distance quadratic-quadratic approximations with the corrected C_L within 0.07% and 0.5% of the fine-mesh estimate.

The error correction results for C_L on a series of non-uniformly refined meshes are presented in Tables (4.13) to (4.16). The tables compare isotropic, cubic spline prolongation operators for different basis functions. The best corrections are observed in linear-linear and quadratic-linear MLS fits. The quadratic-quadratic MLS fit over-predicts the correction at both the mesh levels and the linear-quadratic fit over-predicts at the intermediate mesh. The inferior corrections from the quadratic-quadratic and linear-quadratic MLS fits can be attributed to a bad quadratic fit for the adjoint. This is further confirmed by the excellent corrections produced by the quadratic-linear fit. The parallel CPU costs for the error corrected C_L are shown in Fig.(4.3). It can be inferred from Fig.(4.3) that the fine-mesh C_L estimate is obtained with a factor 3 reduction in parallel cost by performing error correction.

Table 4.11: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Cubic spline weights correction from a coarse-mesh with $C_L = 0.32904858$ to a uniformly refined mesh with $C_L = 0.34140372$; True Error = 3.62%.

MLS fit	Error Correction	Corrected C_L	$\%$ Error after correction
$Linear - Linear(isotropic)$	0.00543562	0.33448420	2.026
$Linear - Linear(anisotropic)$	0.00557426	0.33462284	1.986
$Quadratic - Quadratic(isotropic)$	0.01208721	0.34113579	0.078
$Quadratic - Quadratic(anisotropic)$	0.01276145	0.34181003	-0.119
$Linear - Quadratic(isotropic)$	0.00460063	0.33364921	2.271
$Linear - Quadratic(anisotropic)$	0.00706106	0.33610964	1.550
$Quadratic - Linear(isotropic)$	0.00872527	0.33777385	1.063
$Quadratic - Linear(anisotropic)$	0.01006647	0.33911505	0.670

Table 4.12: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Inverse-distance weights correction from a coarse-mesh with $C_L = 0.32904858$ to a uniformly refined mesh with $C_L = 0.34140372$; True Error = 3.62%.

MLS fit	Error Correction	Corrected C_L	$\%$ Error after correction
$Linear - Linear(isotropic)$	0.00778603	0.33683461	1.334
$Linear - Linear(anisotropic)$	0.00788570	0.33693428	1.309
$Quadratic - Quadratic(isotropic)$	0.01068615	0.33973473	0.488
$Quadratic - Quadratic(anisotropic)$	0.00998060	0.33902918	0.695
$Linear - Quadratic(isotropic)$	0.00600940	0.33505798	1.858
$Linear - Quadratic(anisotropic)$	0.00628382	0.33533240	0.335
$Quadratic - Linear(isotropic)$	0.01325666	0.34230524	-0.264
$Quadratic - Linear(anisotropic)$	0.00987792	0.33892650	0.725

Table 4.13: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Isotropic cubic spline weights correction for C_L with linear-linear basis on non-uniformly refined meshes; True Error[1] = 3.89%, True Error[2] = 0.364%.

Nodes	C_L	Error Correction	Corrected C_L	$\%$ Error after correction
42114	0.32904858	0.00751931	0.33656789	1.697
183796	0.34237983	0.00170301	0.34408284	-0.131
976503	0.34363090	–	–	–

Table 4.14: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Isotropic cubic spline weights correction for C_L with quadratic-quadratic basis on non-uniformly refined meshes; True Error[1] = 3.89%, True Error[2] = 0.364%.

Nodes	C_L	Error Correction	Corrected C_L	$\%$ Error after correction
42114	0.32904858	0.02045818	0.34950676	-2.081
183796	0.34237983	0.00223984	0.34461967	-0.287
976503	0.34363090	–	–	–

Table 4.15: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Isotropic cubic spline weights correction for C_L with linear-quadratic basis on non-uniformly refined meshes; True Error[1] = 3.89%, True Error[2] = 0.364%.

Nodes	C_L	Error Correction	Corrected C_L	% Error after correction
42114	0.32904858	0.00580444	0.33485302	2.198
183796	0.34237983	0.00323724	0.34561707	-0.578
976503	0.34363090	–	–	–

Table 4.16: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Isotropic cubic spline weights correction for C_L with quadratic-linear basis on non-uniformly refined meshes; True Error[1] = 3.89%, True Error[2] = 0.364%.

Nodes	C_L	Error Correction	Corrected C_L	% Error after correction
42114	0.32904858	0.00853922	0.33758780	1.40
183796	0.34237983	0.00101208	0.34339191	0.069
976503	0.34363090	–	–	–

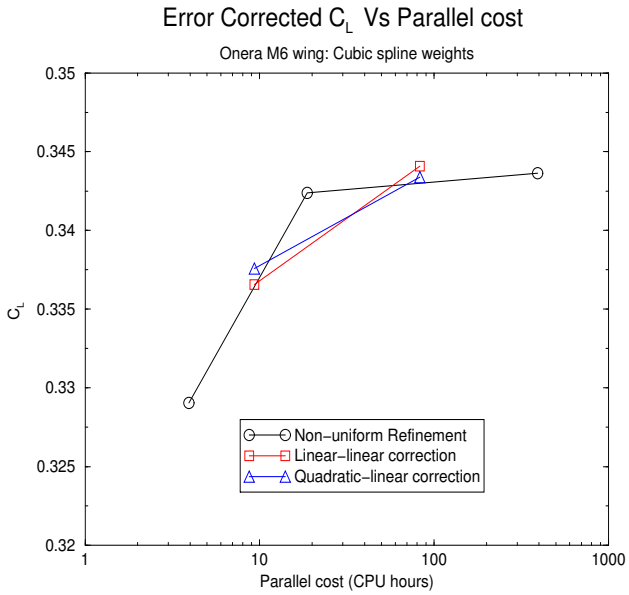


Figure 4.3: Inviscid flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Parallel cost for the error corrected C_L.

4.4.2 Viscous laminar flow

Mixed-element meshes are used for viscous cases with prisms in the boundary layer regions and, pyramids and tetrahedrals outside the boundary layer in the inviscid regions. The MLS support stencil is built with rectangular supports in the boundary layer regions and with circular supports in the inviscid regions. When isotropic weights are applied in the boundary layer regions, the rectangular supports are reconstructed to represent circular supports. Similarly, when tensor product weights are applied in the inviscid regions, the circular supports are reconstructed to represent rectangular supports. No modifications are needed for the mixed support, since, it employs tensor product weights in the boundary layer and isotropic weights in the inviscid regions.

4.4.2.1 C_D in a unit Cylinder at $M_\infty = 0.3$ and $Re = 100$

The first viscous test case is the laminar flow over a unit cylinder at a Mach number of 0.3 and Reynolds number of 100. The output function considered is the drag coefficient C_D. The cylinder is capped with symmetry planes at each end and the coarse-mesh contains 13,543 nodes, 1,792 surface quadrilaterals, 3,098 surface triangles, 20,916 prisms and 8,133 tetrahedrals. The coarse-mesh is uniformly refined to construct a h-refined embedded mesh containing 101,308 nodes, 7,168 surface quadrilaterals, 12,392 surface triangles, 167,328 prisms and 65,064 tetrahedrals. Two non-uniformly refined meshes are generated using AFLR3 [100] with the intermediate-mesh containing 56,808 nodes, 3,380 surface quadrilaterals, 13,452 surface triangles, 81,016 prisms and 66,474 tetrahedrals; and the fine-mesh containing 253,243 nodes, 6,704 surface quadrilaterals, 55,424 surface triangles, 313,807 prisms and 466,177 tetrahedrals.

The error correction results for C_D on a uniformly refined mesh is shown in Tables (4.17) and (4.18). The tables compare the cubic spline and inverse-distance weights for different MLS fits. With the exception of quadratic prolongation operators employing tensor product weights, all the MLS fits reduce the true error by 50%. From Table (4.18) for inverse-distance weights, it can be noticed that the quadratic-quadratic and quadratic-linear anisotropic MLS fits over-predict the correction. This can be attributed to a bad quadratic fit for the flow solution, when the weights are applied anisotropically. This is further confirmed by the excellent correction from linear-quadratic

anisotropic fit. Another possible explanation for the inferior corrections may be because of the use of rectangular supports in the inviscid regions. This can be verified from the quality corrections with a mixed MLS fit, which uses an anisotropic fit in the boundary layer and an isotropic fit in inviscid regions.

Tables (4.19) to (4.26) compare the different MLS operators over a series of non-uniformly refined meshes for mixed supports. Mixed supports are more appropriate in mixed element meshes and so, they are chosen for further investigation. For all the MLS fits, the true error^1 and true error^2 are reduced by more than 50% after applying correction. The remaining error in the corrected C_D from the different MLS fits are almost of the same order of magnitude. Fig.(4.4) shows the parallel CPU costs for the error corrected C_D. It can be noticed that significant savings in parallel cost can be achieved by performing error correction. The global L_2 norm error in the MLS fit for the flow and adjoint solutions on the coarse-mesh is shown in Tables (4.27) and (4.28). As expected, the L_2 norm errors in the inverse-distance weights are smaller, as the MLS fit behaves like an interpolant when the distance between data and seed points approach zero. The L_2 norm errors in the cubic spline quadratic MLS fits for the adjoint are one order of magnitude larger compared to the other cubic spline fits.

Table 4.17: Laminar flow over a Cylinder: $M_\infty = 0.3,\ Re = 100.$ Cubic spline weights correction from a coarse-mesh with $C_D = 2.271672$ to a uniformly refined mesh with $C_D = 2.0778713$; True Error = -9.325%.

MLS fit	Error Correction	Corrected C_D	% Error after correction
$Linear - Linear(isotropic)$	-0.08860659	2.18306541	-5.062
$Linear - Linear(anisotropic)$	-0.07975986	2.19191214	-5.488
$Linear - Linear(mixed)$	-0.08521173	2.18646027	-5.226
$Quadratic - Quadratic(isotropic)$	-0.09542261	2.17624939	-4.734
$Quadratic - Quadratic(anisotropic)$	-0.33037676	1.94129524	6.572
$Quadratic - Quadratic(mixed)$	-0.09161557	2.18005643	-4.917
$Linear - Quadratic(isotropic)$	-0.1363136	2.1353584	-2.767
$Linear - Quadratic(anisotropic)$	-0.1379030	2.1337690	-2.690
$Linear - Quadratic(mixed)$	-0.1313387	2.1403333	-3.006
$Quadratic - Linear(isotropic)$	-0.1209169	2.1507551	-3.507
$Quadratic - Linear(anisotropic)$	-0.1363500	2.1353220	-2.764
$Quadratic - Linear(mixed)$	-0.1221979	2.1494741	-3.445

Table 4.18: Laminar flow over a Cylinder: $M_\infty = 0.3,\ Re = 100.$ Inverse-distance weights correction from a coarse-mesh with $C_D = 2.271672$ to a uniformly refined mesh with $C_D = 2.0778713$; True Error = -9.325%.

MLS fit	Error Correction	Corrected C_D	% Error after correction
$Linear - Linear(isotropic)$	-0.09028313	2.18138887	-4.981
$Linear - Linear(anisotropic)$	-0.09316668	2.17850532	-4.843
$Linear - Linear(mixed)$	-0.09262002	2.17905198	-4.869
$Quadratic - Quadratic(isotropic)$	-0.10138606	2.17028594	-4.447
$Quadratic - Quadratic(anisotropic)$	-0.40984835	1.86182365	10.39
$Quadratic - Quadratic(mixed)$	-0.11318102	2.15849098	-3.88
$Linear - Quadratic(isotropic)$	-0.10326479	2.16840721	-4.357
$Linear - Quadratic(anisotropic)$	-0.12797833	2.14369367	-3.17
$Linear - Quadratic(mixed)$	-0.11352251	2.15814949	-3.863
$Quadratic - Linear(isotropic)$	-0.14136428	2.13030772	-2.523
$Quadratic - Linear(anisotropic)$	-0.42476962	1.84690238	11.11
$Quadratic - Linear(mixed)$	-0.16240366	2.10926834	-1.511

Table 4.19: Laminar flow over a Cylinder: $M_\infty = 0.3$, $Re = 100$. Mixed cubic spline weights correction for C_D with linear-linear basis on non-uniformly refined meshes; True Error^1 = -8.25%, True Error^2 = -8.06%.

Nodes	C_D	Error Correction	Corrected C_D	% Error after correction
13543	2.2716720	-0.05288262	2.21878938	-5.736
56808	2.0984108	-0.04865962	2.04975118	-5.558
253243	1.9418062	–	–	–

Table 4.20: Laminar flow over a Cylinder: $M_\infty = 0.3$, $Re = 100$. Mixed inverse-distance weights correction for C_D with linear-linear basis on non-uniformly refined meshes; True Error^1 = -8.25%, True Error^2 = -8.06%.

Nodes	C_D	Error Correction	Corrected C_D	% Error after correction
13543	2.2716720	-0.0745105	2.1971615	-4.705
56808	2.0984108	-0.07674662	2.02166418	-4.112
253243	1.9418062	–	–	–

Table 4.21: Laminar flow over a Cylinder: $M_\infty = 0.3$, $Re = 100$. Mixed cubic spline weights correction for C_D with quadratic-quadratic basis on non-uniformly refined meshes; True Error^1 = -8.25%, True Error^2 = -8.06%.

Nodes	C_D	Error Correction	Corrected C_D	% Error
13543	2.2716720	-0.07637621	2.19529579	-4.617
56808	2.0984108	-0.13209183	1.96631897	-1.262
253243	1.9418062	–	–	–

Table 4.22: Laminar flow over a Cylinder: $M_\infty = 0.3$, $Re = 100$. Mixed inverse-distance weights correction for C_D with quadratic-quadratic basis on non-uniformly refined meshes; True Error^1 = -8.25%, True Error^2 = -8.06%.

Nodes	C_D	Error Correction	Corrected C_D	% Error after correction
13543	2.2716720	-0.07336892	2.19830308	-4.76
56808	2.0984108	-0.08416162	2.01424918	-3.73
253243	1.9418062	–	–	–

Table 4.23: Laminar flow over a Cylinder: $M_\infty = 0.3$, $Re = 100$. Mixed cubic spline weights correction for C_D with linear-quadratic basis on non-uniformly refined meshes; True Error^1 = -8.25%, True Error^2 = -8.06%.

Nodes	C_D	Error Correction	Corrected C_D	$\%$ Error after correction
13543	2.2716720	-0.10212237	2.16954963	-3.390
56808	2.0984108	-0.06444012	2.03397068	-4.746
253243	1.9418062	–	–	–

Table 4.24: Laminar flow over a Cylinder: $M_\infty = 0.3$, $Re = 100$. Mixed inverse-distance weights correction for C_D with linear-quadratic basis on non-uniformly refined meshes; True Error^1 = -8.25%, True Error^2 = -8.06%.

Nodes	C_D	Error Correction	Corrected C_D	$\%$ Error after correction
13543	2.2716720	-0.09987072	2.17180128	-3.5
56808	2.0984108	-0.10708798	1.99132282	-2.55
253243	1.9418062	–	–	–

Table 4.25: Laminar flow over a Cylinder: $M_\infty = 0.3$, $Re = 100$. Mixed cubic spline weights correction for C_D with quadratic-linear basis on non-uniformly refined meshes; True Error^1 = -8.25%, True Error^2 = -8.06%.

Nodes	C_D	Error Correction	Corrected C_D	$\%$ Error after correction
13543	2.2716720	-0.10570873	2.16596327	-3.219
56808	2.0984108	-0.08306125	2.01534955	-3.787
253243	1.9418062	–	–	–

Table 4.26: Laminar flow over a Cylinder: $M_\infty = 0.3$, $Re = 100$. Mixed inverse-distance weights correction for C_D with quadratic-linear basis on non-uniformly refined meshes; True Error^1 = -8.25%, True Error^2 = -8.06%.

Nodes	C_D	Error Correction	Corrected C_D	$\%$ Error after correction
13543	2.2716720	-0.15995850	2.11171350	-0.634
56808	2.0984108	-0.08757107	2.01083973	-3.55
253243	1.9418062	–	–	–

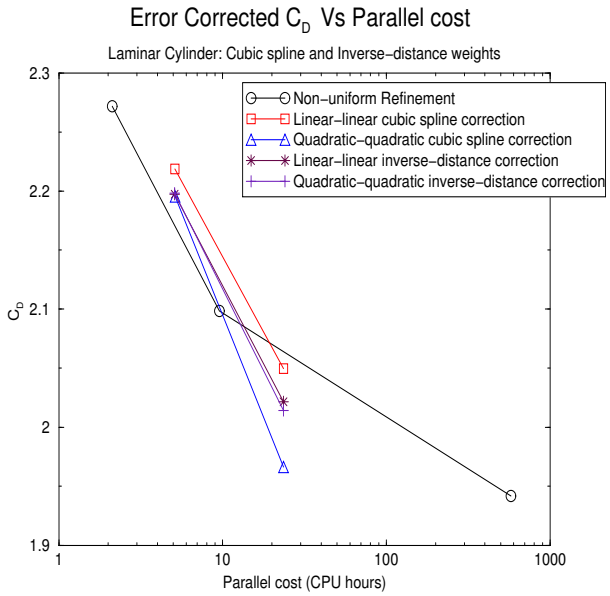


Figure 4.4: Laminar flow over a Cylinder: $M_\infty = 0.3,\ Re = 100$. Parallel cost for the error corrected C_D.

Table 4.27: Laminar flow over a Cylinder: $M_\infty = 0.3, Re = 100$. Global l_2 norm of error in MLS fit for flow solution

MLS fit	Cubic spline weights	Inverse-distance weights
$Linear - Linear(isotropic)$	$1.822E-03$	$8.454E-14$
$Linear - Linear(mixed)$	$1.814E-03$	$7.914E-14$
$Linear - Linear(anisotropic)$	$4.792E-04$	$2.069E-17$
$Quadratic - Quadratic(isotropic)$	$2.139E-03$	$8.165E-14$
$Quadratic - Quadratic(mixed)$	$2.025E-03$	$1.374E-14$
$Quadratic - Quadratic(anisotropic)$	$1.532E-03$	$3.154E-17$

Table 4.28: Laminar flow over a Cylinder: $M_\infty = 0.3, Re = 100$. Global l_2 norm of error in MLS fit for adjoint solution

MLS fit	Cubic spline weights	Inverse-distance weights
$Linear - Linear(isotropic)$	$3.086E-03$	$4.088E-13$
$Linear - Linear(mixed)$	$2.986E-03$	$1.629E-13$
$Linear - Linear(anisotropic)$	$9.407E-04$	$6.647E-18$
$Quadratic - Quadratic(isotropic)$	$8.855E-02$	$1.117E-11$
$Quadratic - Quadratic(mixed)$	$6.968E-02$	$4.561E-14$
$Quadratic - Quadratic(anisotropic)$	$6.964E-02$	$7.153E-18$

4.4.2.2 C_D in a Onera M6 wing at $M_\infty = 0.8395$, $Re = 5,000$ and $AOA = 3.06^0$

The second viscous test case is laminar flow over an Onera M6 wing at a Mach number of 0.8395, chord-based Reynolds number of $5,000$ and angle of attack of 3.06^0. The output function considered is the drag coefficient C_D on the wing. For this test case, two coarse-meshes are generated, a mixed-element and an all tetrahedral mesh, and error estimates are evaluated on a non-uniformly refined mixed-element fine-mesh. For the mixed-element meshes, prisms are used in the boundary layer. For the tetrahedral mesh, the boundary layer prisms are subdivided into tetrahedra in an advancing layer fashion. The mixed-element coarse-mesh contains $80,951$ nodes, 656 surface quadrilaterals, $23,664$ surface triangles, $87,009$ prisms, $3,561$ pyramids and $176,902$ tetrahedrals. The tetrahedral coarse-mesh contains $85,286$ nodes, $25,004$ surface triangles and $470,954$ tetrahedrals, among which $300,944$ are boundary layer tetrahedrals. The fine-mesh contains $516,834$ nodes, $2,607$ surface quadrilaterals, $78,088$ surface triangles, $731,774$ prisms, $23,413$ pyramids and $721,791$ tetrahedrals.

This case is performed to test the approximation power of the MLS prolongation operators between meshes of different element types. This will validate the mesh independent nature of the present MLS approximation. For the mixed-element coarse-mesh, the stencil is built with rectangular supports only in boundary layer and circular supports are used in rest of the regions. Supports are reconstructed when weights are applied isotropically or anisotropically. For the tetrahedral coarse-mesh, the support is either circular in all regions (boundary layer and inviscid) or rectangular in all regions. Tables (4.29) and (4.30) compare the cubic spline MLS fits for the mixed-element and tetrahedral coarse-meshes. Except for the linear-linear MLS fits on the mixed-element coarse-mesh, the remaining error in the corrected C_D is less than 40% of the true error. The MLS operators perform well on both the coarse-meshes, thereby, validating the meshless nature of the present MLS approximation.

Table 4.29: Laminar flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$, $Re = 5,000$. Cubic spline weights correction from a mixed-element coarse-mesh with $C_D = 0.08785503$ to a non-uniformly refined mixed-element mesh with $C_D = 0.08280842$; True Error = -6.09%.

MLS fit	Error Correction	Corrected C_D	% Error after correction
$Linear - Linear(isotropic)$	0.00066847	0.08852350	-6.901
$Linear - Linear(anisotropic)$	-0.00061292	0.08724211	-5.354
$Linear - Linear(mixed)$	-0.00059159	0.08726344	-5.379
$Quadratic - Quadratic(isotropic)$	-0.00380478	0.08405025	-1.50
$Quadratic - Quadratic(anisotropic)$	-0.00148669	0.08636834	-4.30
$Quadratic - Quadratic(mixed)$	-0.00466223	0.08319280	-0.464
$Linear - Quadratic(isotropic)$	-0.00285646	0.08499857	-2.644
$Linear - Quadratic(anisotropic)$	-0.00310203	0.08475300	-2.348
$Linear - Quadratic(mixed)$	-0.00296975	0.08488528	-2.508
$Quadratic - Linear(isotropic)$	-0.00389405	0.08396098	-1.391
$Quadratic - Linear(anisotropic)$	-0.00485201	0.08300302	-0.235
$Quadratic - Linear(mixed)$	-0.00417283	0.08368220	-1.055

Table 4.30: Laminar flow over a Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$, $Re = 5,000$. Cubic spline weights correction from a Tetrahedral coarse-mesh with $C_D = 0.08726068$ to a non-uniformly refined mixed-element mesh with $C_D = 0.08280842$; True Error = -5.37%.

MLS fit	Error Correction	Corrected C_D	% Error after correction
$Linear - Linear(isotropic)$	-0.00192239	0.08533829	-3.055
$Linear - Linear(anisotropic)$	-0.00217514	0.08508554	-2.75
$Quadratic - Quadratic(isotropic)$	-0.00225737	0.08500331	-2.650
$Quadratic - Quadratic(anisotropic)$	-0.00258690	0.08467378	-2.252
$Linear - Quadratic(isotropic)$	-0.00420017	0.08306051	-0.304
$Linear - Quadratic(anisotropic)$	-0.00391970	0.08334098	-0.643
$Quadratic - Linear(isotropic)$	-0.00240976	0.08485092	-2.466
$Quadratic - Linear(anisotropic)$	-0.00281541	0.08444527	-1.976

4.4.3 Turbulent flow

The one-equation turbulence model of Spalart and Allmaras [88] is used in the present study. The turbulence model is solved in a loosely coupled manner in both flow and sensitivity analysis. The turbulent contributions to the mean flow are added by the eddy viscosity μ_t. To better understand the influence of μ_t on the computed error estimates, the error correction for turbulent flows is evaluated in three ways: (1) μ_t is prolongated from the coarse-mesh; (2) μ_t is evaluated by Eqn.(2.22) using the prolongated turbulent quantity $\tilde{\nu}$ at the fine-mesh; (3) μ_t is floored to zero removing the turbulent contributions in mean flow.

4.4.3.1 C_D in a NACA 0012 rectangular wing at $M_\infty = 0.95$, $Re = 3,000,000$ and $AOA = 0^0$

The test case is turbulent flow over a NACA 0012 rectangular wing at a Mach number of 0.95, chord-based Reynolds number of $3,000,000$ and angle of attack of 0^0. The coarse-mesh contains $356,420$ nodes, $2,576$ surface quadrilaterals, $23,072$ surface triangles, $611,744$ prisms and $258,874$ tetrahedrals with a wall spacing of $8e^{-06}$ of mean aerodynamic chord. Error correction is performed on a non-uniformly refined fine-mesh containing $1,003,430$ nodes, $4,570$ surface quadrilaterals, $71,972$ surface triangles, $1,706,813$ prisms and $767,814$ tetrahedrals with the same wall spacing as coarse-mesh.

The error corrections results are presented in Tables (4.31) to (4.36) for cubic spline and inverse-distance weights. From the tables, it can be observed that the best corrections are observed in quadratic-quadratic and quadratic-linear MLS fits for mixed supports, with the remaining error in the corrected C_D less than 50% of the true error. Also, the error corrections evaluated with computed μ_t (option (2)) are significantly better than the interpolated and zeroed μ_t (options (1) and (3)). For the quadratic-quadratic fit, the isotropic supports (all circular) and the anisotropic supports (all rectangular) produce too small and too large corrections. This under-prediction and over-prediction of corrections may be attributed to the use of rectangular supports in inviscid regions for the anisotropic case and circular supports in the highly stretched boundary layer for the isotropic case.

All the linear-linear MLS fits produce inferior corrections with the wrong signs. This behavior is observed in all linear-quadratic fits also. The source of the inferior corrections can be attributed to a bad linear approximation for the flow solution. The highly stretched nature of the boundary layer elements with aspect ratios exceeding 10,000 may have resulted in an ill-conditioned MLS system that is converging to the wrong values for the linear approximation. Though the estimates are not shown here, the linear approximations are tried with the larger quadratic stencil, but, they also resulted in corrections with wrong signs. This behavior requires further investigation of the MLS support stencil for turbulent corrections and should be addressed in future research.

The parallel CPU costs for the error corrected C_D are shown in Fig.(4.5). It can be inferred that large savings in parallel cost may be realized by performing error correction. The quality of the MLS fits for the turbulent case is shown in Tables (4.37) to (4.40). The L_2 norm errors in the linear fits do not explain the behavior of the bad linear approximations. The quadratic approximations have a larger L_2 norm error, but, produced better corrections compared to the linear fits.

Table 4.31: Turbulent flow over a NACA0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Cubic spline weights correction from a coarse-mesh with $C_D = 0.06146953$ to a non-uniformly refined mesh with $C_D = 0.07064086$ with prolongated μ_t; True Error = 12.98%.

MLS fit	Error Correction	Corrected C_D	$\%$ Error after correction
$Linear - Linear(isotropic)$	-0.02044105	0.04102848	41.92
$Linear - Linear(anisotropic)$	-0.02603397	0.03543556	49.83
$Linear - Linear(mixed)$	-0.02049659	0.04097294	42.0
$Quadratic - Quadratic(isotropic)$	0.00066702	0.06213655	12.03
$Quadratic - Quadratic(anisotropic)$	0.01170592	0.07317545	-3.58
$Quadratic - Quadratic(mixed)$	0.00421005	0.06567958	7.02
$Linear - Quadratic(isotropic)$	-0.01892106	0.04254847	39.77
$Linear - Quadratic(anisotropic)$	-0.01778208	0.04368745	38.15
$Linear - Quadratic(mixed)$	-0.01914460	0.04232493	40.08
$Quadratic - Linear(isotropic)$	0.00314096	0.06461049	8.53
$Quadratic - Linear(anisotropic)$	-0.00642091	0.05504862	22.07
$Quadratic - Linear(mixed)$	0.00387947	0.06534900	7.5

Table 4.32: Turbulent flow over a NACA0012 rectangular wing: $M_\infty = 0.95,\ AOA = 0^0$, $Re = 3,000,000$. Inverse-distance weights correction from a coarse-mesh with $C_D = 0.06146953$ to a non-uniformly refined mesh with $C_D = 0.07064086$ with prolongated μ_t; True Error = 12.98%.

MLS fit	Error Correction	Corrected C_D	$\%$ Error after correction
$Linear - Linear(isotropic)$	-0.01976715	0.04170238	40.96
$Linear - Linear(anisotropic)$	-0.02684151	0.03462802	50.98
$Linear - Linear(mixed)$	-0.01986285	0.04160668	41.10
$Quadratic - Quadratic(isotropic)$	0.00284616	0.06431569	8.95
$Quadratic - Quadratic(anisotropic)$	0.01381020	0.07527973	-6.56
$Quadratic - Quadratic(mixed)$	0.00444236	0.06591189	6.69
$Linear - Quadratic(isotropic)$	-0.01863537	0.04283416	39.36
$Linear - Quadratic(anisotropic)$	-0.01992065	0.04154888	41.18
$Linear - Quadratic(mixed)$	-0.01702510	0.04444443	37.08
$Quadratic - Linear(isotropic)$	0.00996636	0.07143589	-1.12
$Quadratic - Linear(anisotropic)$	0.00931081	0.07078034	-0.20
$Quadratic - Linear(mixed)$	0.00611743	0.06758696	4.321

Table 4.33: Turbulent flow over a NACA0012 rectangular wing: $M_\infty = 0.95,\ AOA = 0^0$, $Re = 3,000,000$. Cubic spline weights correction from a coarse-mesh with $C_D = 0.06146953$ to a non-uniformly refined mesh with $C_D = 0.07064086$ with computed μ_t; True Error = 12.98%.

MLS fit	Error Correction	Corrected C_D	$\%$ Error after correction
$Linear - Linear(isotropic)$	-0.02053464	0.04093489	42.05
$Linear - Linear(anisotropic)$	-0.02575492	0.03571461	49.44
$Linear - Linear(mixed)$	-0.02063805	0.04083148	42.2
$Quadratic - Quadratic(isotropic)$	0.00073695	0.06220648	11.93
$Quadratic - Quadratic(anisotropic)$	0.01523128	0.07670081	-8.57
$Quadratic - Quadratic(mixed)$	0.00718791	0.06865744	2.81
$Linear - Quadratic(isotropic)$	-0.01913630	0.04233323	40.07
$Linear - Quadratic(anisotropic)$	-0.01776635	0.04370318	38.13
$Linear - Quadratic(mixed)$	-0.01946077	0.04200876	40.53
$Quadratic - Linear(isotropic)$	-0.00230264	0.05916689	16.24
$Quadratic - Linear(anisotropic)$	-0.00599742	0.05547211	21.47
$Quadratic - Linear(mixed)$	0.00606966	0.06753919	4.39

Table 4.34: Turbulent flow over a NACA0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Inverse-distance weights correction from a coarse-mesh with $C_D = 0.06146953$ to a non-uniformly refined mesh with $C_D = 0.07064086$ with computed μ_t; True Error = 12.98%.

MLS fit	Error Correction	Corrected C_D	$\%$ Error after correction
$Linear - Linear(isotropic)$	-0.01993086	0.04153867	41.20
$Linear - Linear(anisotropic)$	-0.02592073	0.03554880	49.67
$Linear - Linear(mixed)$	-0.01984222	0.04162731	41.07
$Quadratic - Quadratic(isotropic)$	0.00350393	0.06497346	8.02
$Quadratic - Quadratic(anisotropic)$	0.01395067	0.07542020	-6.76
$Quadratic - Quadratic(mixed)$	0.00458292	0.06605245	6.5
$Linear - Quadratic(isotropic)$	-0.01895162	0.04251791	39.81
$Linear - Quadratic(anisotropic)$	-0.02003803	0.04143150	41.35
$Linear - Quadratic(mixed)$	-0.01714216	0.04432737	37.25
$Quadratic - Linear(isotropic)$	0.01017355	0.07164308	-1.42
$Quadratic - Linear(anisotropic)$	0.011644282	0.07311381	-3.5
$Quadratic - Linear(mixed)$	0.00845078	0.06992031	1.02

Table 4.35: Turbulent flow over a NACA0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Cubic spline weights correction from a coarse-mesh with $C_D = 0.06146953$ to a non-uniformly refined mesh with $C_D = 0.07064086$ with μ_t floored to zero; True Error = 12.98%.

MLS fit	Error Correction	Corrected C_D	$\%$ Error after correction
$Linear - Linear(isotropic)$	-0.01843408	0.04303545	39.07
$Linear - Linear(anisotropic)$	-0.01819439	0.04327514	38.73
$Linear - Linear(mixed)$	-0.01861335	0.04285618	39.33
$Quadratic - Quadratic(isotropic)$	0.00037584	0.06184537	12.45
$Quadratic - Quadratic(anisotropic)$	0.01059496	0.07206449	-2.01
$Quadratic - Quadratic(mixed)$	0.00310038	0.06456991	8.6
$Linear - Quadratic(isotropic)$	-0.01847939	0.04299014	39.14
$Linear - Quadratic(anisotropic)$	-0.01330620	0.04816333	31.82
$Linear - Quadratic(mixed)$	-0.01891338	0.04255615	39.75
$Quadratic - Linear(isotropic)$	0.00025986	0.06172939	12.61
$Quadratic - Linear(anisotropic)$	-0.00635012	0.05511941	21.97
$Quadratic - Linear(mixed)$	0.00319125	0.06466078	8.46

Table 4.36: Turbulent flow over a NACA0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Inverse-distance weights correction from a coarse-mesh with $C_D = 0.06146953$ to a non-uniformly refined mesh with $C_D = 0.07064086$ with μ_t floored to zero; True Error = 12.98%.

MLS fit	Error Correction	Corrected C_D	$\%$ Error after correction
$Linear - Linear(isotropic)$	-0.01785474	0.04361479	38.25
$Linear - Linear(anisotropic)$	-0.01949098	0.04197855	40.57
$Linear - Linear(mixed)$	-0.01784394	0.04362559	38.24
$Quadratic - Quadratic(isotropic)$	0.00016784	0.06163737	12.74
$Quadratic - Quadratic(anisotropic)$	0.00727151	0.06874104	2.68
$Quadratic - Quadratic(mixed)$	0.00209401	0.06356354	10.01
$Linear - Quadratic(isotropic)$	-0.01825629	0.04321324	38.82
$Linear - Quadratic(anisotropic)$	-0.02082238	0.04064715	42.45
$Linear - Quadratic(mixed)$	-0.01791899	0.04355054	38.34
$Quadratic - Linear(isotropic)$	0.00752036	0.06898989	2.33
$Quadratic - Linear(anisotropic)$	0.00553917	0.06700870	5.14
$Quadratic - Linear(mixed)$	0.00232975	0.06379928	9.68

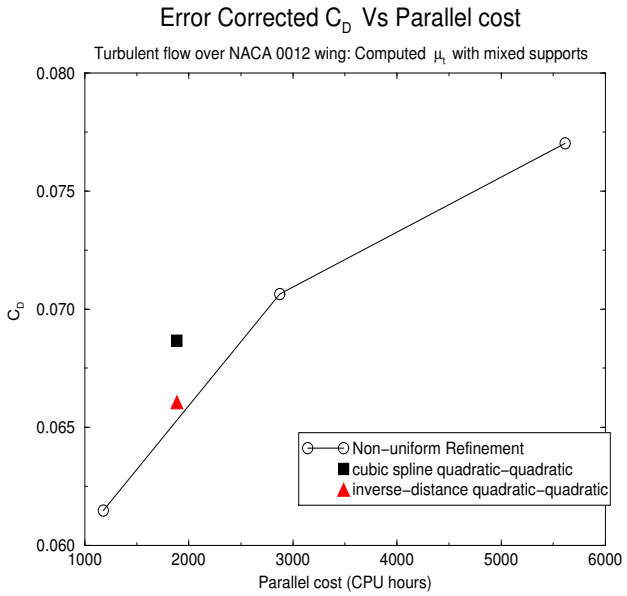


Figure 4.5: Turbulent flow over a NACA0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Parallel cost for the error corrected C_D with computed μ_t and mixed supports.

Table 4.37: Turbulent flow over NACA 0012 wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Global l_2 norm of error in MLS fit for flow solution

MLS fit	Cubic spline weights	Inverse-distance weights
$Linear - Linear(isotropic)$	$7.392E-04$	$4.880E-12$
$Linear - Linear(mixed)$	$6.905E-04$	$5.647E-13$
$Linear - Linear(anisotropic)$	$3.094E-04$	$2.839E-17$
$Quadratic - Quadratic(isotropic)$	$2.163E-03$	$1.302E-10$
$Quadratic - Quadratic(mixed)$	$1.367E-03$	$4.333E-14$
$Quadratic - Quadratic(anisotropic)$	$1.072E-03$	$3.216E-17$

Table 4.38: Turbulent flow over NACA 0012 wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Global l_2 norm of error in MLS fit for adjoint solution

MLS fit	Cubic spline weights	Inverse-distance weights
$Linear - Linear(isotropic)$	$5.210E-02$	$1.182E-09$
$Linear - Linear(mixed)$	$4.205E-03$	$7.823E-12$
$Linear - Linear(anisotropic)$	$1.789E-03$	$3.122E-16$
$Quadratic - Quadratic(isotropic)$	$1.719E-01$	$3.866E-08$
$Quadratic - Quadratic(mixed)$	$1.371E-01$	$2.228E-10$
$Quadratic - Quadratic(anisotropic)$	$9.972E-02$	$5.723E-14$

Table 4.39: Turbulent flow over NACA 0012 wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Global l_2 norm of error in MLS fit for turbulent quantity $\bar{\nu}$

MLS fit	Cubic spline weights	Inverse-distance weights
$Linear - Linear(isotropic)$	$1.257E+00$	$1.873E-09$
$Linear - Linear(mixed)$	$1.255E+00$	$1.745E-09$
$Linear - Linear(anisotropic)$	$9.502E-01$	$5.901E-20$
$Quadratic - Quadratic(isotropic)$	$4.102E+00$	$3.750E-09$
$Quadratic - Quadratic(mixed)$	$2.715E+00$	$1.374E-14$
$Quadratic - Quadratic(anisotropic)$	$1.056E+00$	$4.154E-17$

Table 4.40: Turbulent flow over NACA 0012 wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Global l_2 norm of error in MLS fit for turbulent-adjoint

MLS fit	Cubic spline weights	Inverse-distance weights
$Linear - Linear(isotropic)$	$1.649E-06$	$1.666E-14$
$Linear - Linear(mixed)$	$8.450E-07$	$7.823E-12$
$Linear - Linear(anisotropic)$	$5.986E-07$	$6.647E-18$
$Quadratic - Quadratic(isotropic)$	$9.804E-05$	$4.152E-12$
$Quadratic - Quadratic(mixed)$	$7.552E-05$	$4.561E-14$
$Quadratic - Quadratic(anisotropic)$	$6.713E-05$	$6.721E-18$

CHAPTER V

GRID ADAPTATION

This section demonstrates the final step in the present error correction methodology: implementation of an efficient and robust output-based grid adaptive scheme. The adjoint-based and feature-based adaptive approaches are introduced and the adaptive indicators are formulated. The adaptation mechanics is discussed and adaptation results are presented for inviscid, laminar and turbulent test cases.

5.1 Adjoint-based Approach

The present output-based adaptive strategy suggested by Venditti and Darmofal [48, 69–71] is based on the adjoint error correction procedure described in section 4.1. The adaptive procedure strives to improve the error estimates $\{\gamma_H^h\}^T R_h(Q_H^h)$ by reducing the level of error in the computable error correction. By including the error in computable estimates, Eqn.(4.9) can be written as

$$F(Q) \; - \; F_h(Q_H^h) \; \approx \; \{\gamma_H^h\}^T R_h(Q_H^h) \; + \; \left\{\gamma_h - \gamma_H^h\right\}^T \; R_h(Q_H^h) \tag{5.1}$$

In the above equation, the first term on the right hand side is the computable error correction and the second term is the error in computable correction. The relationship between the primal (flow) and dual (adjoint) problem gives rise to another expression for the second term. Neglecting the non-linear effects, the second term can be written as

$$\left(\gamma_h - \gamma_H^h\right)^T \; R_h(Q_H^h) \approx R_h^\gamma(\gamma_H^h) \; \left(Q_h - Q_H^h\right) \tag{5.2}$$

where $R_h^{\gamma}(\gamma_H^h)$ is the adjoint residual given by

$$R_h^{\gamma}(\gamma) \;=\; \left[\frac{\partial R_h}{\partial Q_h}\right]^T_{Q_H^h} \{\gamma\} \;+\; \left\{\frac{\partial F_h}{\partial Q_h}\right\}_{Q_H^h} \tag{5.3}$$

A conservative adaptive indicator can be formed by including the errors in computing the adjoint solution also in the formulation. The error indicator (E_I) formed by including both the primal and adjoint residual errors is

$$E_I \;=\; \frac{\left(\gamma_h - \gamma_H^h\right)^T \; R_h(Q_H^h) \;+\; R_h^{\gamma}(\gamma_H^h) \; \left(Q_h - Q_H^h\right)}{2} \tag{5.4}$$

5.1.1 Implementation

The error intensity (E_I) at each fine-mesh node *i* can be evaluated as

$$(E_I)_i \;=\; \frac{\left\{ \; \left|\left(\gamma_h - \gamma_H^h\right)_i^T \; R_h(Q_H^h)_i\right| \;+\; \left|R_h^{\gamma}(\gamma_H^h)_i \; \left(Q_h - Q_H^h\right)_i\right| \; \right\}}{2} \tag{5.5}$$

By approximating

$$\begin{aligned} [\gamma_h - \gamma_H^h] \;&\approx\; [(\gamma_H^h)^{HO} - (\gamma_H^h)^{LO}] \equiv [\gamma^{HO} - \gamma^{LO}] \\ [Q_h - Q_H^h] \;&\approx\; [(Q_H^h)^{HO} - (Q_H^h)^{LO}] \equiv [Q^{HO} - Q^{LO}] \end{aligned}$$

Eqn. (5.5) can be written as

$$(E_I)_i \;=\; \frac{\left|\left(\gamma^{HO} - \gamma^{LO}\right)_i^T R_h(Q^{HO})_i\right| \;+\; \left|R_h^{\gamma}(\gamma^{HO})_i \left(Q^{HO} - Q^{LO}\right)_i\right|}{2} \tag{5.6}$$

where γ^{HO}, Q^{HO} and γ^{LO}, Q^{LO} are the higher-order and lower-order prolongated adjoint and flow solutions. Higher-order prolongation is performed using quadratic basis and lower-order prolongation is achieved using linear basis. This has advantages computationally, as the need to solve for the adjoint solution at the fine-mesh is avoided and the accuracy is not affected

significantly. The only computational costs on this larger mesh are function evaluations, flow and adjoint residual evaluations, and dot product of vectors.

The formation of adaptation parameters at the coarse-mesh from the error intensities at the fine-mesh differs slightly for uniformly and non-uniformly refined fine-meshes, but, the underlying principle is essentially the same. The adaptation parameter A_p^1 for a coarse-mesh node *k* can be obtained from a uniformly refined fine-mesh by looping over all the coarse-mesh edges surrounding node *k*, and adding one-half of the error intensities (E_I) from each of the embedded fine-mesh nodes located at the midpoint of these edges. The adaptation parameter A_p^1 at node *k* is given by

$$(A_p^1)_k \;=\; \sum_{j=1}^{n(k)} \frac{(E_I)_j}{2} \tag{5.7}$$

where $n(k)$ is the summation over all the embedded fine-mesh nodes (which are at the midpoint of the edges surrounding node *k*). The adaptation parameters can be obtained from a non-uniformly refined fine-mesh by looping over all the coarse-mesh elements that contain fine-mesh nodes and split the error intensity E_I between all the nodes that make the element. The adaptation parameter A_p^1 at node *k* is given by

$$(A_p^1)_k \;=\; \sum_{i=1}^{e(k)} \sum_{j=1}^{n(i)} \frac{(E_I)_j}{d_i} \tag{5.8}$$

where $e(k)$ is the number of coarse-mesh elements incident at node *k*, $n(i)$ is the number of fine-mesh nodes contained by element i and d_i is the element size (number of nodes that make element i).

5.2 Feature-based Approach

The second adaptive strategy is based on feature detection [64–67]. This approach strives to identify and resolve the significant features of the flow. The adaptation parameter (A_p^2) can be defined as

$$A_p^2 \;=\; \{e_1, e_2, e_3\} \tag{5.9}$$

where e_1, e_2, e_3 are the error indicators given by

$$e_1 = max\left[-\frac{\mathbf{V}\cdot grad\,Q}{|\mathbf{V}|}, 0\right] \tag{5.10}$$

$$e_2 = max\left[+\frac{\mathbf{V}\cdot grad\,Q}{|\mathbf{V}|}, 0\right] \tag{5.11}$$

$$e_3 = \left| grad\,Q - \frac{\mathbf{V}}{|\mathbf{V}|}\left(\frac{\mathbf{V}\cdot grad\,Q}{|\mathbf{V}|}\right)\right| \tag{5.12}$$

$|\cdot|$ represent the magnitude, Q is any suitable flow property and $\mathbf{V}$ denotes the velocity vector. Each of these error indicators can isolate a particular type of feature. The first two error indicators represent expansions and compressions in the flow direction and the third represents gradients normal to the flow direction [65]. At viscous boundaries, A_p^2 is defined only by the magnitude of the gradient of Q ($|\,grad\,Q|$) because of the no-slip boundary conditions ($\mathbf{V}$=0).

5.3 Adaptation Mechanics

Simple adaptation mechanics are employed in the present study. The adaptation procedure employed in the present study constitutes two stages: (1) formation of element-adaptation flags; (2) h-refinement.

The adaptation parameters A_p^1 or A_p^2 given in Eqns.(5.7), (5.8) and (5.9) are formed at all the coarse-mesh nodes. These nodal values are transferred to the elements by a simple averaging. The mean (μ) and standard deviation (σ) of the adaptation parameters over all the coarse-mesh elements are computed and elements are flagged for refinement, if their adaptation parameter is greater than a proposed error limit (e_{lim}) given by

$$e_{lim} = \mu + crel * \sigma \tag{5.13}$$

where *crel* is a relaxation factor usually greater than 0.5.

An unstructured mesh refinement module is developed using Python and C++ to perform adaptation. The adaptive mesh library is based on [84–86] and performs isotropic h-refinement of the elements. The refinement template [87] controls the pattern of subdivision of the mesh

elements. The h-refinement templates used in the present study are shown in Appendix B. In the present study, only node addition is considered and mesh coarsening is not implemented. H-refinement is performed in 4 steps:

1. isotropically refine all flagged elements; the isotropic refinement pattern for each element type is shown in Appendix B.

2. loop over all elements and identify the refinement pattern for each element by adding new nodes if needed; this step should be repeated till no new nodes are added and all the elements have valid refinement patterns.

3. split the element based on its refinement pattern.

4. perform boundary projection.

For h-refined tetrahedral meshes, quality improvement by local reconnection and Laplacian smoothing is performed using AFLR3 [100]. No quality improvement is performed on mixed-element meshes. In the present study, the mixed-element meshes from AFLR3 [100] typically have prisms in the boundary layer, tetrahedra in the inviscid regions and pyramids in the transition region. Anisotropic refinement of prisms resulted in poor quality pyramids and tetrahedrals in the boundary layer. This is not desirable and hence, anisotropic refinement of prisms is not supported in the present study. Because of this, no refinement is allowed in the streamwise direction of boundary layer, especially for prisms. If streamwise refinement is allowed, this resulted in the refinement of the whole layer to preserve the shape of prism.

The adaptation process creates boundary nodes at the midpoint of the edges, by a simple averaging of the node coordinates. Boundary node projection is accomplished by using a transfinite, visually continuous, triangular interpolant explained in [60, 101]. The interpolant is based on side-vertex interpolation in triangles [101]. The cubic Hermite interpolant uses outward surface normals to reconstruct the surface and the resulting reconstructed surface is a G^1 representation with a continuously varying outward normal vector. The reader is referred to

[60] for a derivation of the interpolant. This is a slightly better approximation than the simple averaging of the nodes. In boundary layer regions if the projected boundary node resulted in negative volumes, it is replaced with the averaged value.

5.4 Results and Discussions

Grid adaptation is performed using adjoint-based and feature-based approaches and the improvement in functional estimates is observed. For adjoint-based adaptation, the adaptive indicators are evaluated by prolongating the flow and adjoint solutions to a fine-mesh using the MLS approximation described in chapter 4. The fine-mesh may be uniformly or non-uniformly refined, though, uniform refinement should be the best choice if multiple iterations of adaptation are performed. In the present study, the number of adaptive iterations is limited to one or two and so, non-uniformly refined meshes are also used, wherever seemed appropriate.

5.4.1 Inviscid flow

The first test case is inviscid flow over an Onera M6 wing at an angle of attack of 3.06^0 and Mach number of 0.8395. A lambda shock is typical of these flow conditions and the estimates of C_L and C_D are largely dependent on the accurate prediction of the shock and its location. Adjoint-based adaptation is performed for both lift (C_L) and drag (C_D) coefficients on the wing and compared with feature-based adaptation. For the adjoint-based adaptation, the non-uniformly refined fine-mesh with $976,503$ nodes, $286,728$ surface triangles and $5,372,918$ tetrahedrals is used to establish the adaptive indicators.

5.4.1.1 C_D in a Onera M6 wing at $M_\infty = 0.8395$ and $AOA = 3.06^0$

Figs.(5.1a) and (5.1b) show the convergence of C_D after two adaptive iterations. The adjoint-based C_D adaptation reaches the finest-mesh estimate in two iterations with less than $200,000$ nodes compared to $976,503$ nodes at the finest-mesh; a factor 5 reduction in mesh size for the same level of C_D accuracy. Also, adjoint adaptation with correction achieves super-convergent estimates and have probably converged to the asymptotic value for C_D. The feature-based

adaptation is converging to an erroneous value significantly larger than the finest-mesh C_D. From Fig.(5.1a), it can be observed that the C_D accuracy from adjoint-based C_L adaptation is better than feature-based adaptation, but, slightly inferior to adjoint-based C_D adaptation. From the parallel CPU costs shown in Fig.(5.1b), it can be noticed that large savings in parallel cost may be realized by performing adjoint adaptation.

The initial and adjoint-adapted (C_D) grids are shown in Figs.(5.2a) and (5.2b). There is significant refinement near the leading and trailing edges and at the lambda shock location. Fig.(5.3) shows the initial grid on the upper wing surface. The surface grids from adjoint and feature adaptations are shown in Figs.(5.4) and (5.5). The feature-based adapted grid has excessive refinement near the leading and trailing edges, but, have poor shock resolution. The poor C_D results from the feature approach may be attributed to the failure to resolve the lambda shock. The adaptation parameter contours for the adjoint and feature-based approaches on the upper wing surface of the initial grid are shown in Figs.(5.6a) to (5.6d). The adjoint error indicators have identified the lambda shock, whereas, the feature error indicators have failed to identify the lambda shock properly. A look at the initial density contours in Fig.(5.7) can explain this behavior. The initial grid has resolved the shock poorly and the feature-based approach may need a well resolved initial grid to identify the features. The resolution of the initial grid did not pose a problem for the adjoint approach. The density contours on the upper wing surface of the adjoint and feature-adapted grids are shown in Figs.(5.8) and (5.9). The lambda shock is clearly visible and captured well by the adjoint approach. However, the feature approach has failed to resolve the shock in two iterations of adaptation.

5.4.1.2 C_L in a Onera M6 wing at $M_\infty = 0.8395$ and $AOA = 3.06^0$

Fig.(5.10) shows the convergence of C_L for the adjoint and feature approaches. From Fig.(5.10), it can be noticed that C_L is better predicted by the adjoint approach, and the feature approach is converging to a lower estimate of C_L. More adaptive iterations are needed for the adjoint approach to attain the same level of accuracy as the finest-mesh estimate. However, the

error corrected C_L estimate from adjoint adaptation has achieved the finest-mesh accuracy in two iterations. Also, the C_L accuracy from adjoint-based C_L and C_D adaptations compare favorably. Fig.(5.11a) shows the C_L adaptation contours on the upper wing surface for the first adaptation. The adapted grid after two C_L adaptations is shown in Fig.(5.11b) for the upper wing surface. It can be observed that there is considerable refinement near the lambda shock and near the leading and trailing edges of the wing.

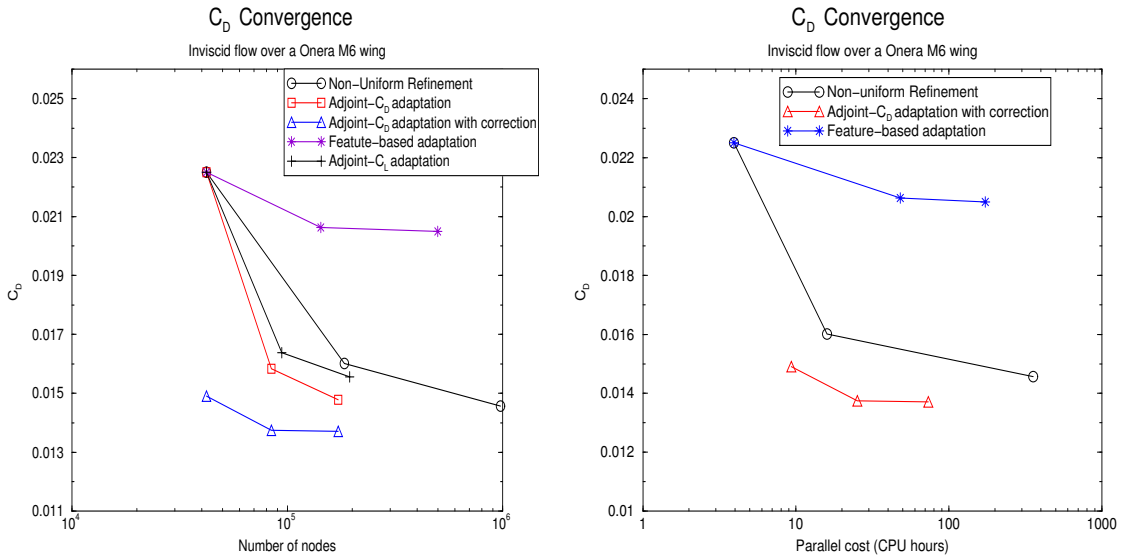
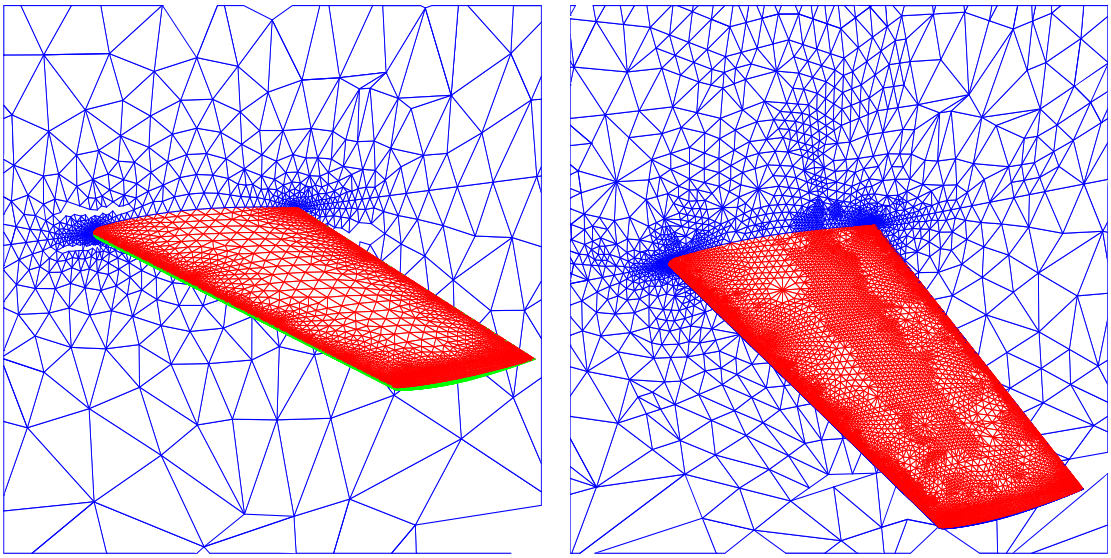


Figure 5.1: Inviscid flow over Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. C_D convergence. Correction computed with isotropic cubic spline weights and linear-quadratic basis. (a) C_D Vs Number of Nodes. (b) C_D Vs Parallel cost.



(a) (b)

Figure 5.2: Inviscid flow over Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. (a) Initial Onera M6 wing grid with $42,114$ nodes. (b) Onera M6 wing grid after two adjoint-C_D adaptations with $172,299$ nodes.

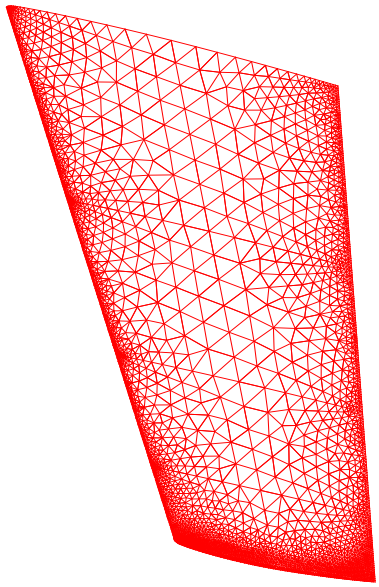


Figure 5.3: Inviscid flow over Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Initial surface grid on upper wing (Total Nodes: $42,114$).

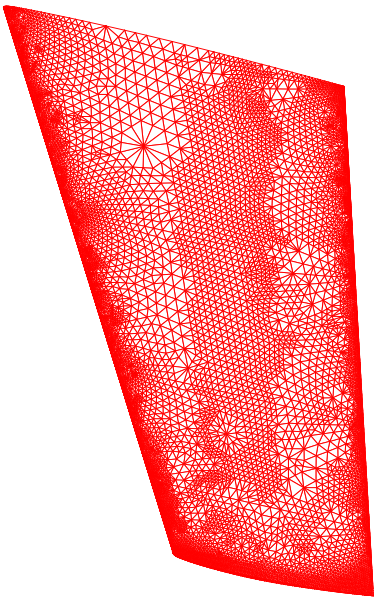


Figure 5.4: Inviscid flow over Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Surface grid on upper wing after two adjoint-C_D adaptations (Total Nodes: $172,299$).

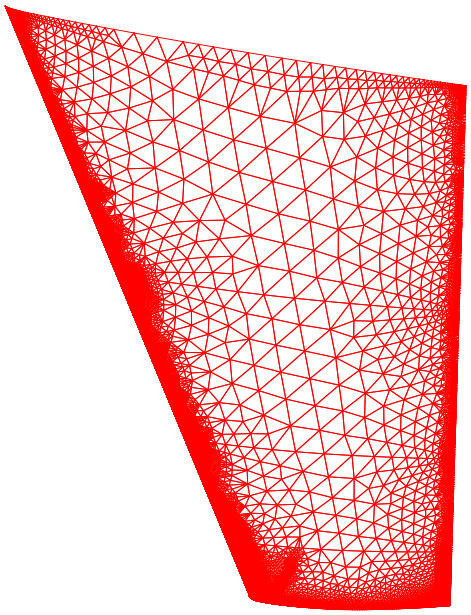
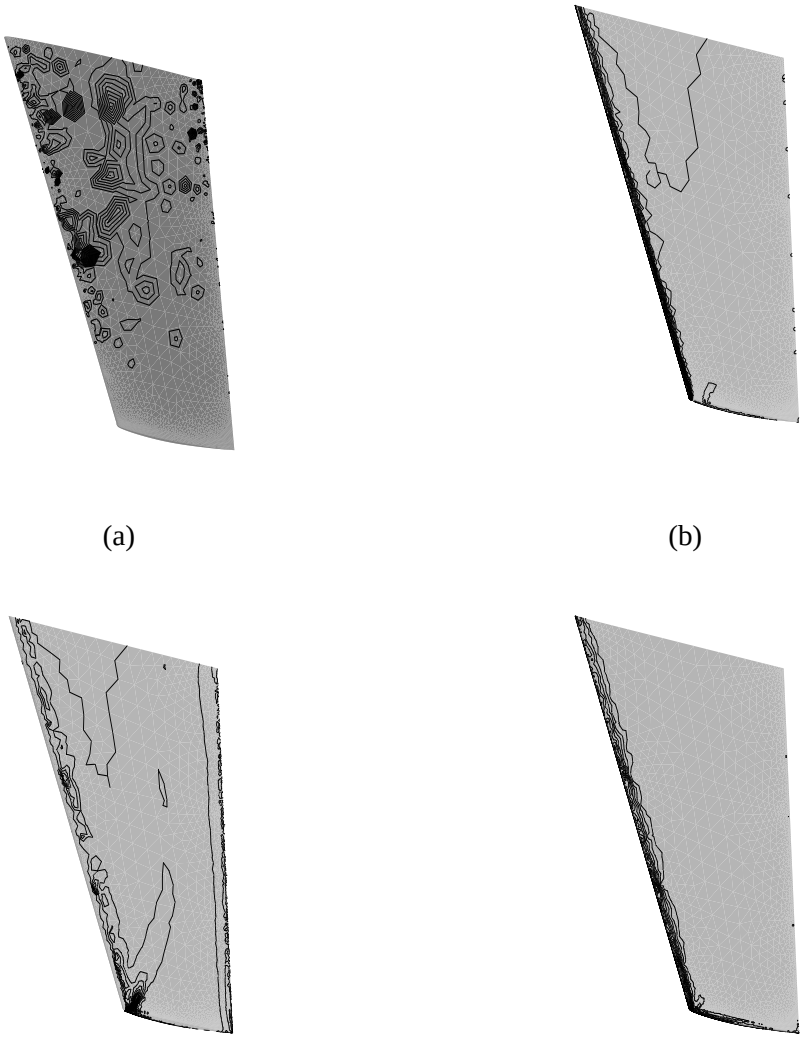


Figure 5.5: Inviscid flow over Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Surface grid on upper wing after two feature adaptations (Total Nodes: $498,863$).



(a) (b)

(c) (d)

Figure 5.6: Inviscid flow over Onera M6 wing: $M_\infty = 0.8395,\ AOA = 3.06^0$. Adaptation parameters on upper wing surface in initial grid. (a) Adjoint-C_D (Eqn.(5.8)); (b) Feature1 (Eqn.(5.10)); (c) Feature2 (Eqn.(5.11)); (d) Feature3 (Eqn.(5.12)).

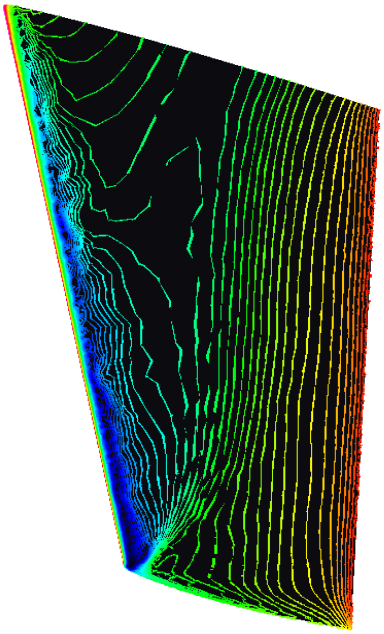


Figure 5.7: Inviscid flow over Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Density contours on upper wing for initial grid.

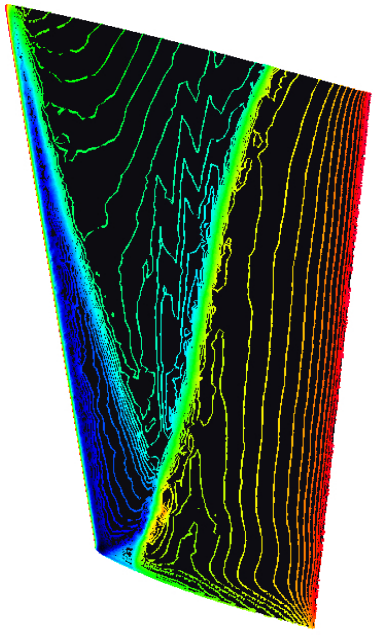


Figure 5.8: Inviscid flow over Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Density contours on upper wing for adjoint-C_D adapted grid.

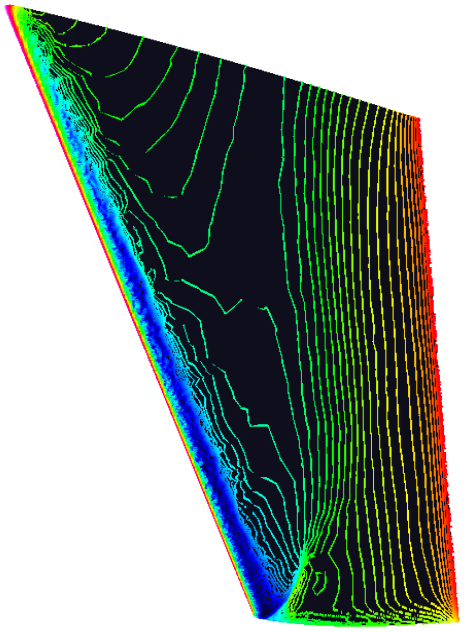


Figure 5.9: Inviscid flow over Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. Density contours on upper wing for feature-adapted grid.

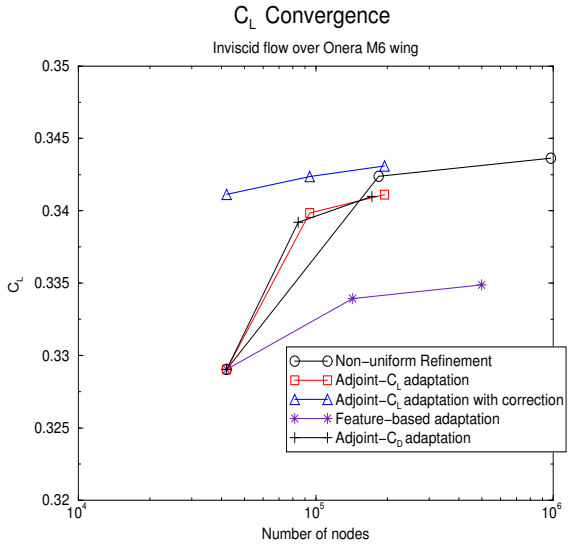
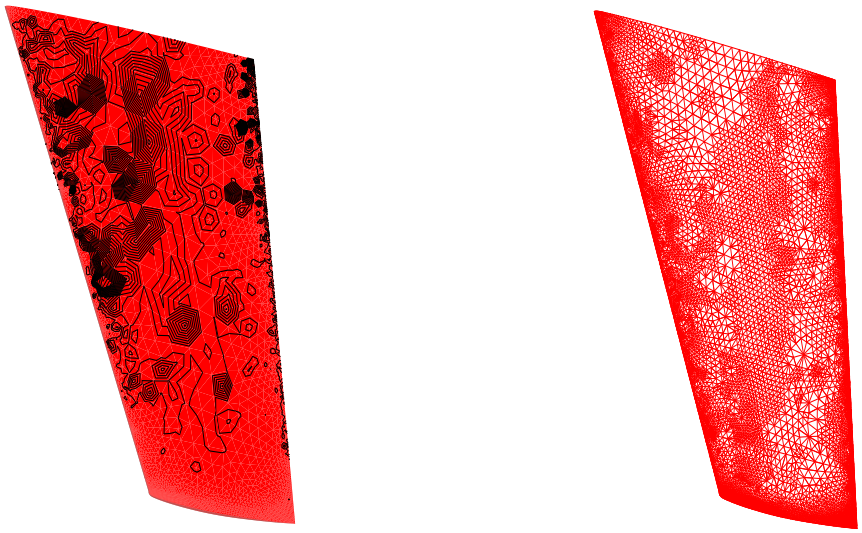


Figure 5.10: Inviscid flow over Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. C_L convergence. Correction computed with isotropic cubic spline weights and quadratic-quadratic basis.



(a) (b)

Figure 5.11: Inviscid flow over Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$. (a) Adjoint-C_L adaptation parameters on upper wing surface in the initial grid. (b) Surface grid on upper wing after two adjoint-C_L adaptations (Total nodes: $177,540$).

5.4.2 Viscous laminar flow

For the viscous cases, only adjoint adaptation is performed and uniformly refined grids are used to evaluate the adaptive indicators.

5.4.2.1 C_D in a unit Cylinder at $M_\infty = 0.1$ and $Re = 20$

The first viscous test case is laminar flow over a unit cylinder at a Mach number of 0.1 and Reynolds number of 20. The cylinder is capped with symmetry planes at both ends and has a height of half its diameter. The initial cylinder grid shown in Fig.(5.12) contains 22,242 nodes, 4,608 surface quadrilaterals, 2,520 surface triangles, 36,864 prisms and 5,618 tetrahedrals. Adaptation is performed for C_D using the adjoint approach.

The convergence of C_D is shown in Figs.(5.13a) and (5.13b). The adjoint-based adaptation converges to the finest-mesh estimate in two iterations. The adapted grid has $126,812$ nodes compared to $303,152$ nodes for the finest-mesh; a factor 2 reduction in mesh size is achieved for the same level of C_D accuracy. A more accurate estimate of C_D is attained by combining error correction with adaptation. The parallel CPU costs are shown in Fig.(5.13b), and it can be noticed that considerable savings in parallel cost is obtained by performing adaptation.

The initial and final adapted symmetry plane grids are shown in Figs.(5.14a) and (5.14b). Near field views of the symmetry plane grids are shown in Figs.(5.15a) and (5.15b). From Fig.(5.14b), it can be observed that there is significant refinement in the front and wake of cylinder. The wake regions are always a source of drag, and it can be noticed that the adjoint-based adaptation has identified these regions for enrichment. The leading edge stagnation point and the regions of flow acceleration near the top and bottom of the cylinder have been considerably refined. These are the regions of the flow where pressure changes rapidly in the streamwise direction. This is confirmed by a look at the pressure contours on the symmetry plane of the initial grid in Fig.(5.16). The pressure contours on the symmetry plane of the final adapted grid is shown in Fig.(5.17), and it can be noticed that the pressure contours are symmetric and are better resolved. The velocity magnitude ($U = \sqrt{u^2 + v^2 + w^2}$) contours on the farther symmetry plane for the initial and adapted grids are shown in Figs.(5.18) and (5.19). From the Figs.(5.18) and (5.19), it can be

observed that the shear layer is better resolved in the adapted grid in the wake regions. The tangential component of velocity vector for the initial and adapted grids are shown in Figs.(5.20) and (5.21). The attached symmetric vortices behind the cylinder can be observed.

5.4.2.2 C_L in a Onera M6 wing at $M_\infty = 0.8395$, $Re = 5,000$ and $AOA = 3.06^0$

The second case is laminar flow over an Onera M6 wing at a Mach number of 0.8395, chord-based Reynolds number of $5,000$ and angle of attack of 3.06^0. The initial grid contains $85,286$ nodes, $25,004$ surface triangles and $470,954$ tetrahedrals. Out of $470,954$ tetrahedra, $300,944$ are boundary layer tetrahedra that are arranged in an advancing layer fashion. For this case, the flow separates from the suction side of the wing at approximately 75% chord lengths, while remaining attached on the lower surface. The output function considered is the lift coefficient C_L on the wing. Because of the highly stretched tetrahedral elements in the boundary layer, only one iteration of adaptation is performed, and the improvement in C_L is observed.

The convergence of C_L is shown in Fig.(5.22). The C_L estimate from the adapted grid with $224,997$ nodes is better than the estimate on the uniformly refined mesh with $654,047$ nodes; a factor 3 reduction in mesh size is achieved with an improved level of accuracy. The initial and adapted upper wing surface grids are shown in Figs.(5.23a) and (5.23b). Figs.(5.24) and (5.25) show the initial and adapted symmetry plane grids. The initial grid was generated with a large boundary layer growth factor using AFLR3 [100] to keep the mesh size small. It can be noticed that the first few layers of boundary layer have been refined to compensate for the large growth factor. Also, the region where the flow separates on the upper surface (approximately 75% of chord length) has been considerably refined. The regions upstream of the leading edge (outside the boundary layer), and the wake regions also have significant refinement. The Mach number contours on the symmetry plane for the initial and adapted grids are shown in Figs.(5.26) and (5.27). From the plots, it can be noticed that the shear layer is resolved better in the adapted grid.

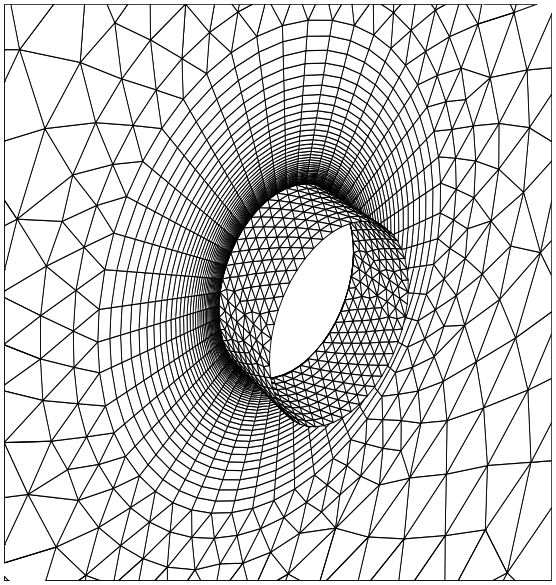


Figure 5.12: Laminar flow over a cylinder: $M_\infty = 0.1$, $Re = 20$. Initial cylinder grid with the near symmetry plane removed.

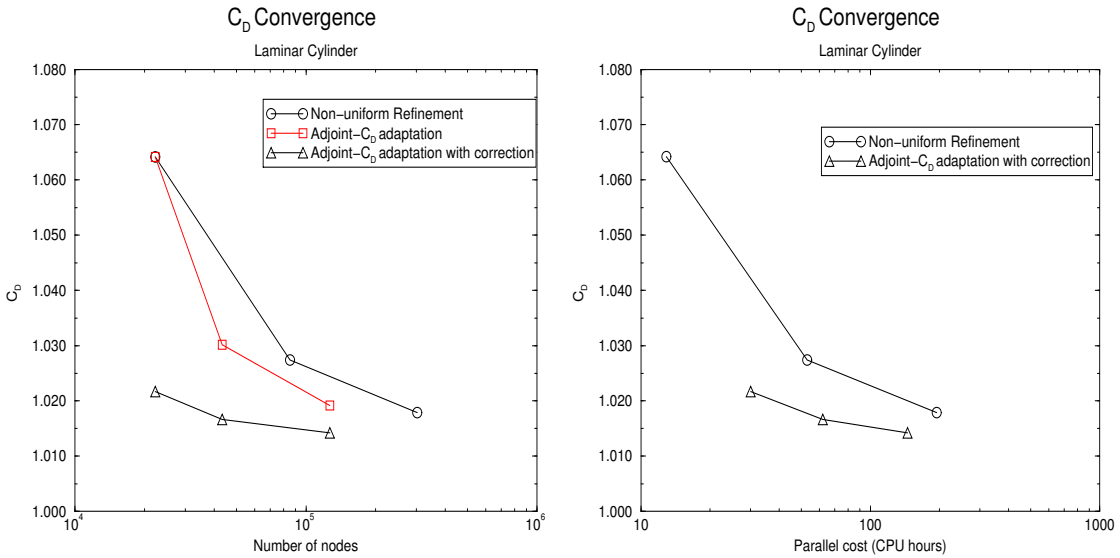
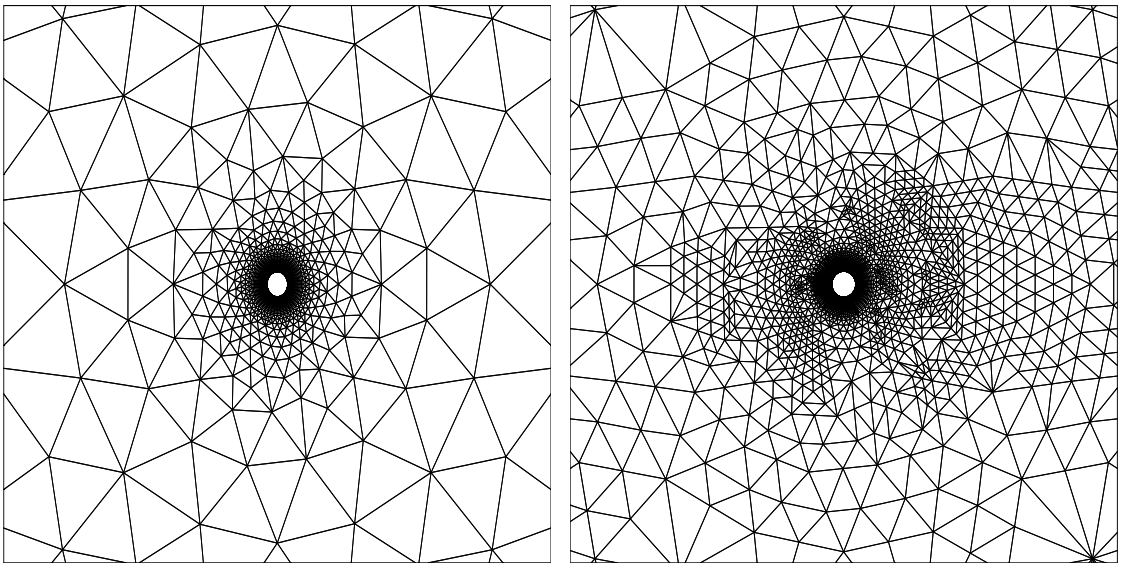
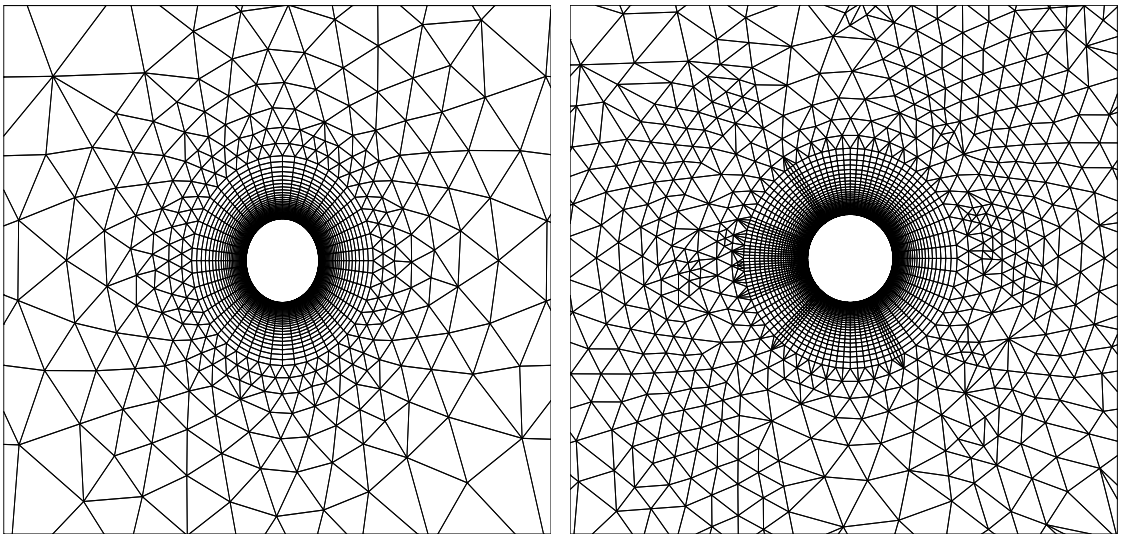


Figure 5.13: Laminar flow over a cylinder: $M_\infty = 0.1$, $Re = 20$. C_D convergence. Correction computed with mixed cubic spline weights and quadratic-quadratic basis (a) C_D Vs Number of Nodes. (b) C_D Vs Parallel cost.



(a) (b)

Figure 5.14: Laminar flow over a cylinder: $M_\infty = 0.1$, $Re = 20$. (a) Initial symmetry plane grid (Total Nodes: $22,242$). (b) Symmetry plane grid after two adjoint-C_D adaptations (Total Nodes: $126,812$).



(a) (b)

Figure 5.15: Laminar flow over a cylinder: $M_\infty = 0.1$, $Re = 20$. (a) Near field view of initial symmetry plane grid. (b) Near field view of adjoint-adapted symmetry plane grid.

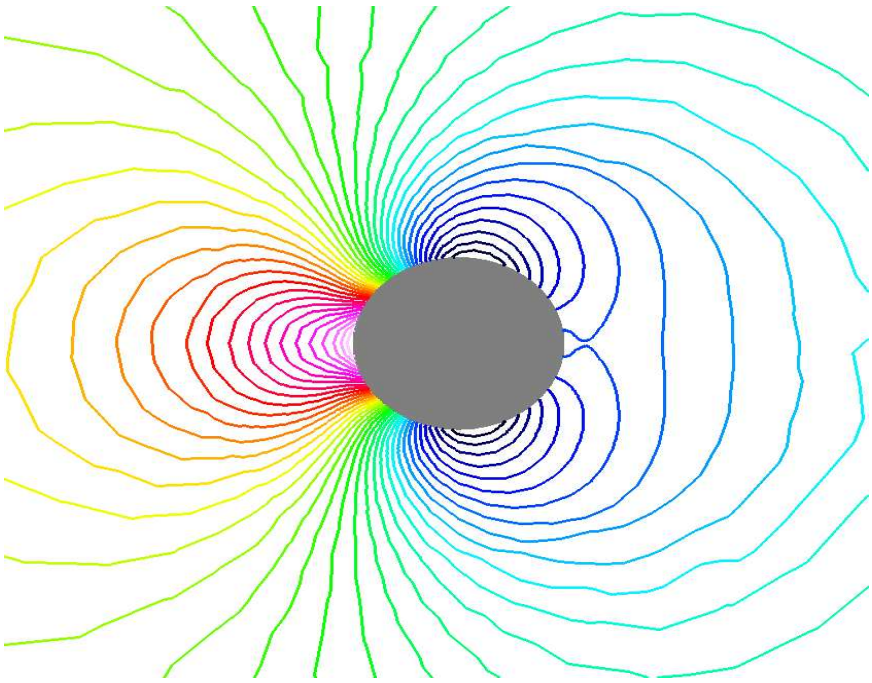


Figure 5.16: Laminar flow over a cylinder: $M_\infty = 0.1,\ Re = 20.$ Pressure contours on the symmetry plane for initial grid.

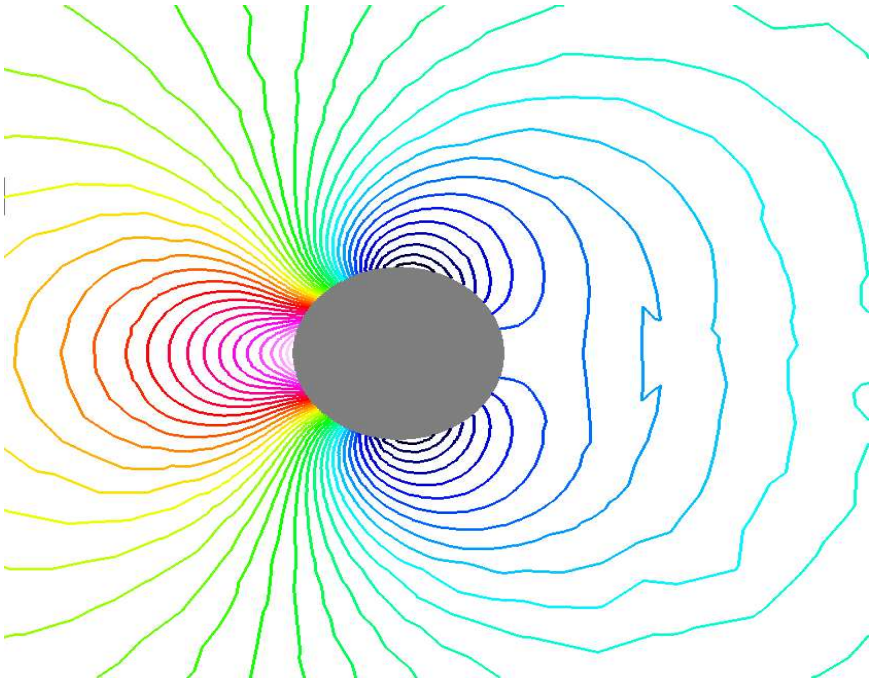


Figure 5.17: Laminar flow over a cylinder: $M_\infty = 0.1,\ Re = 20.$ Pressure contours on the symmetry plane for adjoint-adapted grid.

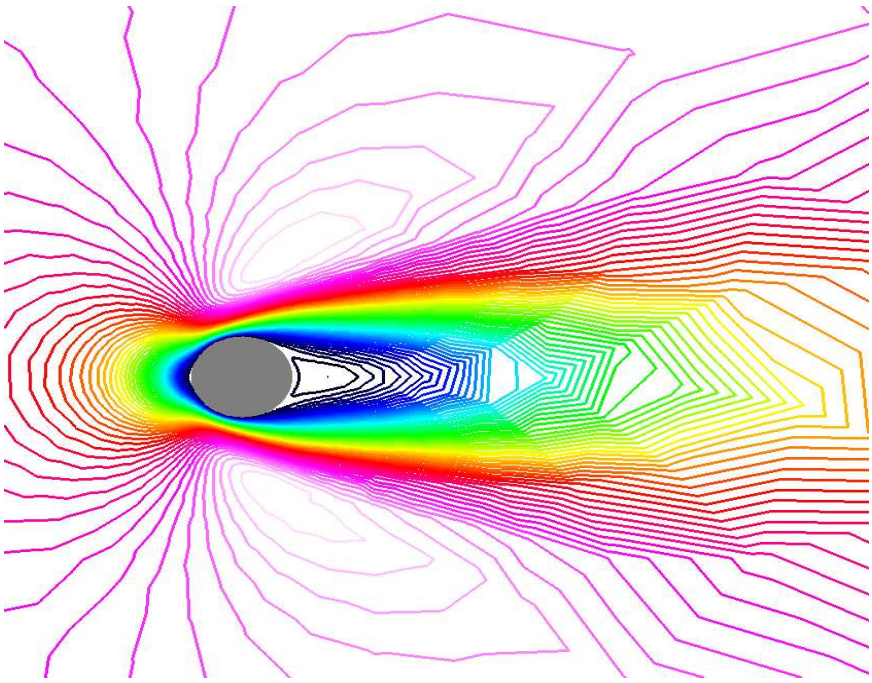


Figure 5.18: Laminar flow over a cylinder: $M_\infty = 0.1$, $Re = 20$. Velocity magnitude contours on the symmetry plane for initial grid.

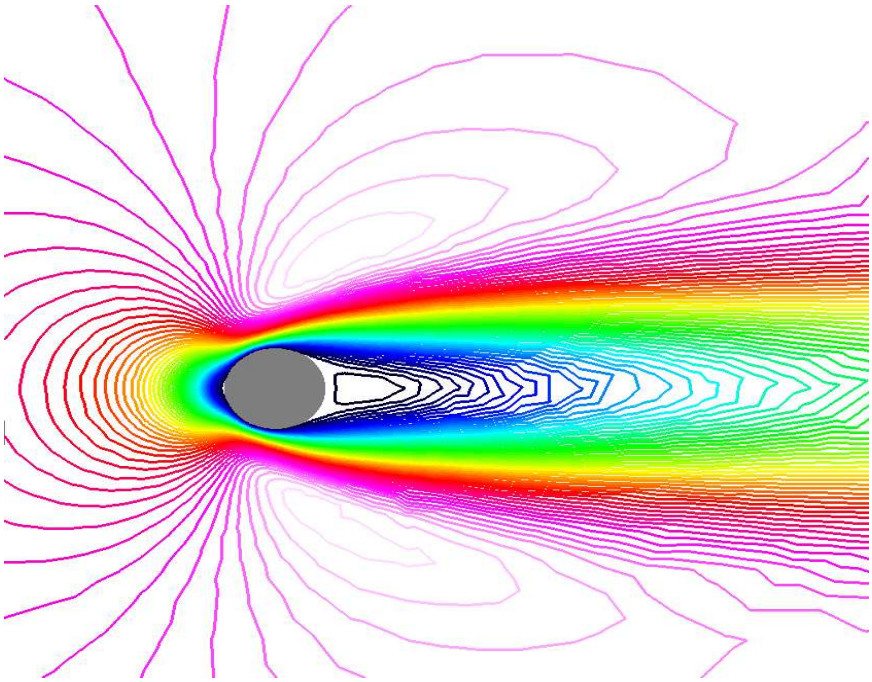


Figure 5.19: Laminar flow over a cylinder: $M_\infty = 0.1$, $Re = 20$. Velocity magnitude contours on the symmetry plane for adjoint-adapted grid.

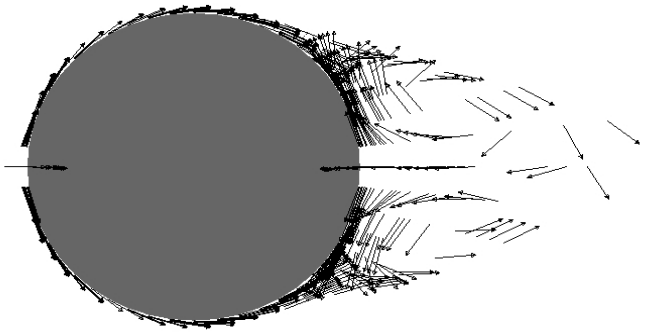


Figure 5.20: Laminar flow over a cylinder: $M_\infty = 0.1,\ Re = 20$. Tangential component of velocity vector on the symmetry plane for initial grid.

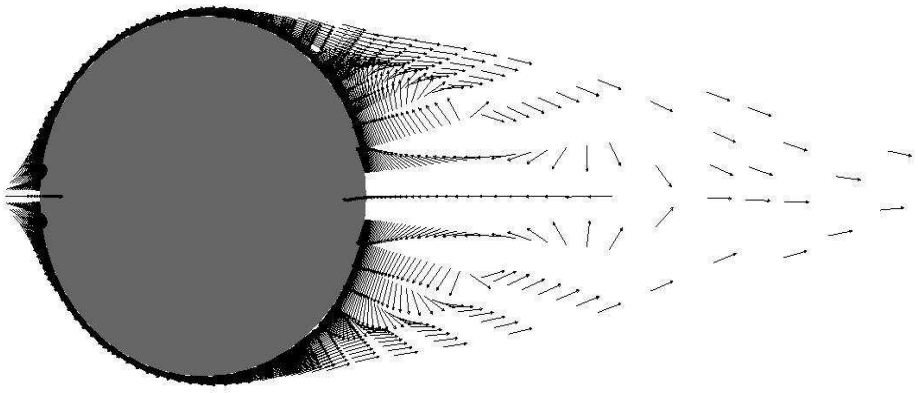


Figure 5.21: Laminar flow over a cylinder: $M_\infty = 0.1,\ Re = 20$. Tangential component of velocity vector on the symmetry plane for adjoint-adapted grid.

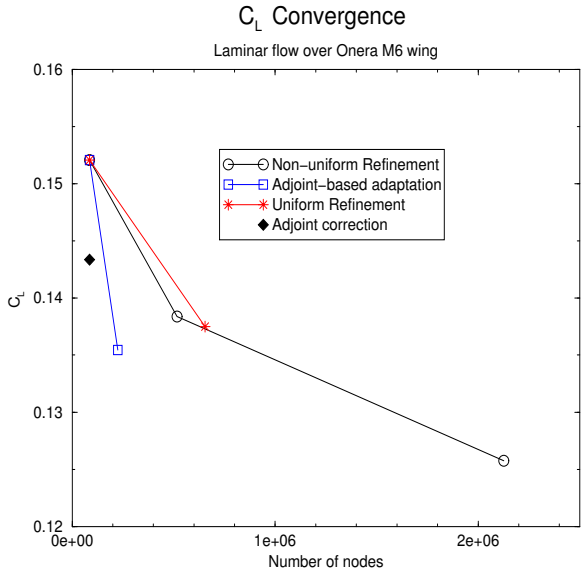
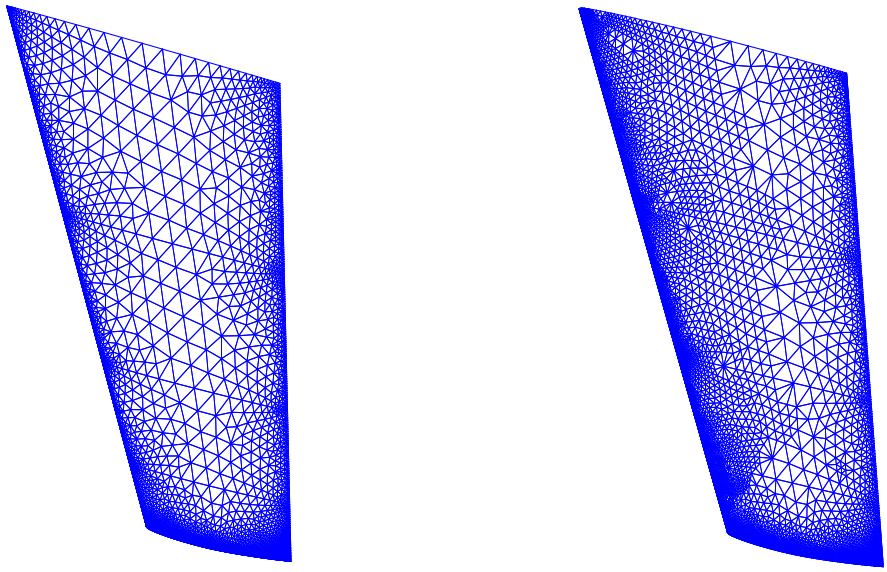


Figure 5.22: Laminar flow over Onera M6 wing: $M_\infty = 0.8395$ and $AOA = 3.06^0$. C_L convergence. Correction computed with anisotropic cubic spline weights and quadratic-quadratic basis.



(a) (b)

Figure 5.23: Laminar flow over Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$, $Re = 5,000$. (a) Initial surface grid on upper wing. (b) Surface grid on upper wing after one adjoint-C_L adaptation

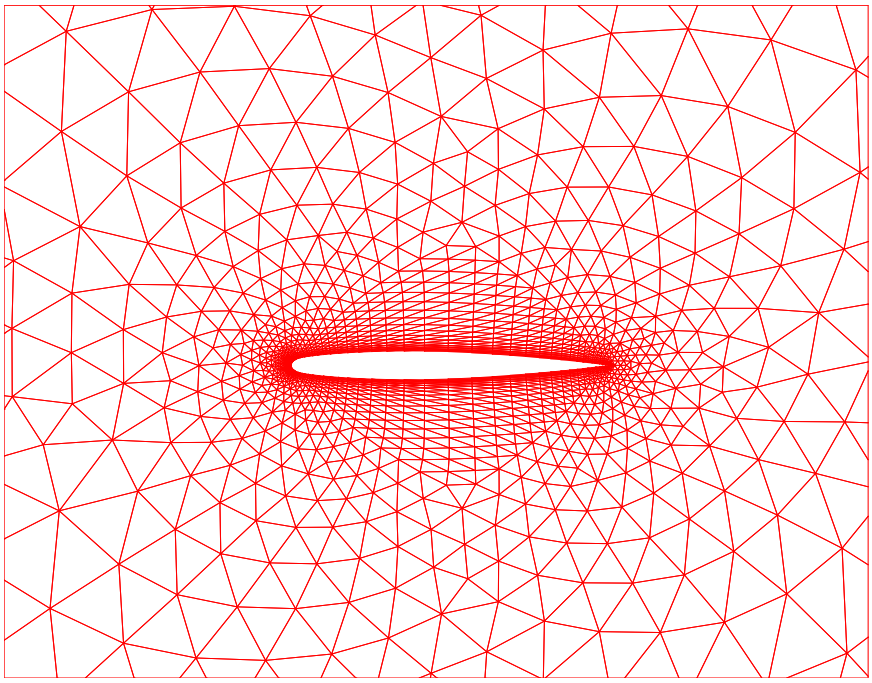


Figure 5.24: Laminar flow over Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$, $Re = 5,000$. Initial symmetry plane grid (Total Nodes: $85,286$).

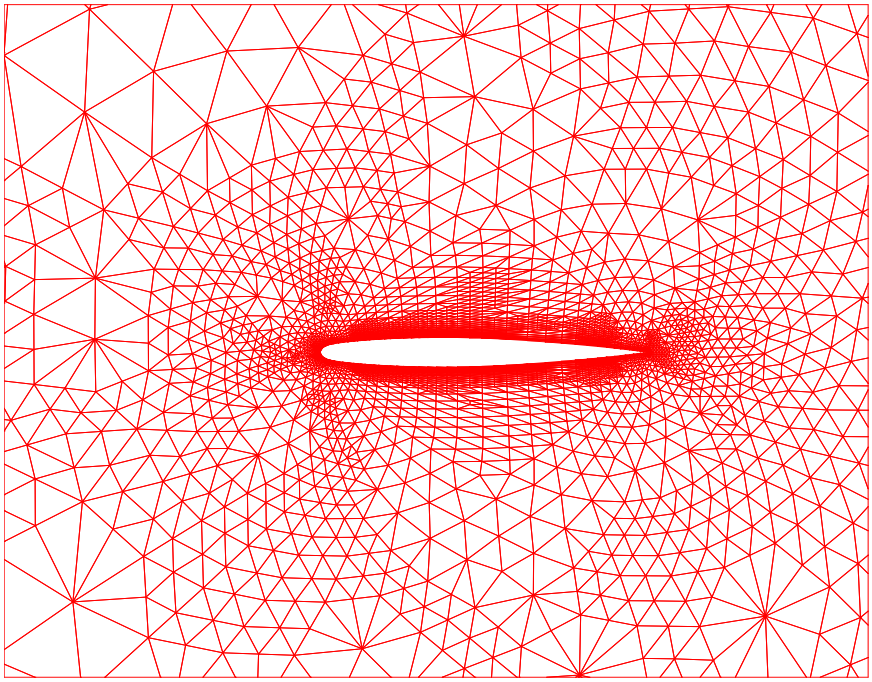


Figure 5.25: Laminar flow over Onera M6 wing: $M_\infty = 0.8395$, $AOA = 3.06^0$, $Re = 5,000$. Symmetry plane grid after one adjoint-C_L adaptation (Total Nodes: $224,997$).

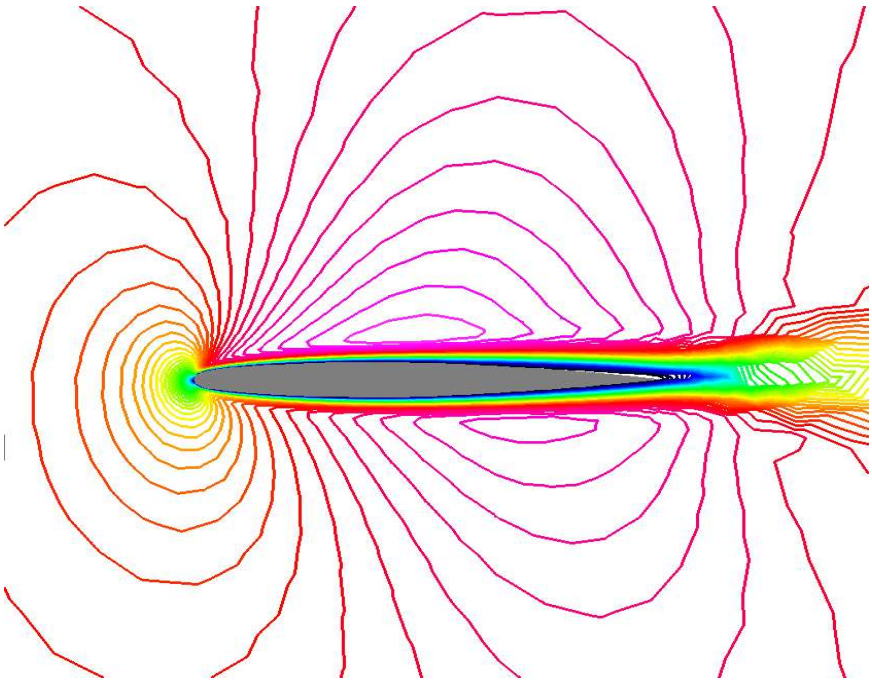


Figure 5.26: Laminar flow over Onera M6 wing: $M_\infty = 0.8395,\ AOA = 3.06^0,\ Re = 5,000.$ Mach number contours on the symmetry plane for initial grid.

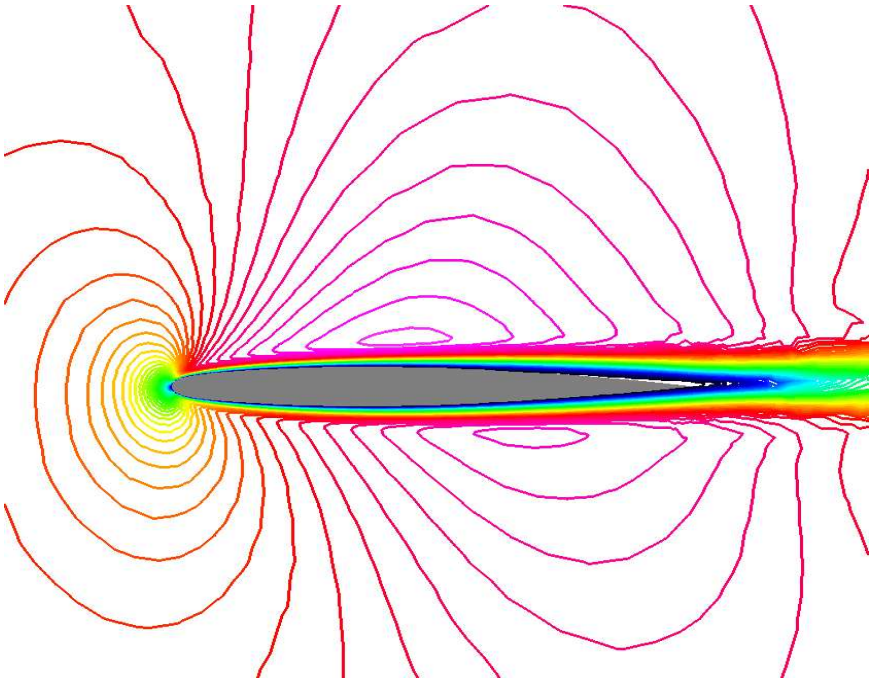


Figure 5.27: Laminar flow over Onera M6 wing: $M_\infty = 0.8395,\ AOA = 3.06^0,\ Re = 5,000.$ Mach number contours on the symmetry plane for adjoint-adapted grid.

5.4.3 Turbulent flow

Because of memory constraints, only a single iteration of adaptation is performed for the turbulent case.

5.4.3.1 C_D in a NACA 0012 rectangular wing at $M_\infty = 0.95$, $Re = 3,000,000$ and $AOA = 0^0$

Turbulent flow is simulated over a NACA 0012 rectangular wing at a Mach number of 0.95, chord-based Reynolds number of $3,000,000$ and angle of attack of 0^0. The test case is a supercritical flow with strong shocks on the upper and lower surfaces of the wing. There is shock/boundary layer interaction and because of the presence of strong shocks, the boundary layer thickens and separates on the upper and lower wing surfaces. The output function considered is the drag coefficient C_D on the wing. The estimates of C_D are largely dependent on the accurate prediction of the upper and lower wing shocks, especially their locations, and adequate resolution of the separation zones in the boundary layer. The initial grid contains $356,420$ nodes, $2,576$ surface quadrilaterals, $23,072$ surface triangles, $611,744$ prisms and $258,874$ tetrahedrals with a wall spacing of 8×10^{-06} of the mean aerodynamic chord. For the adjoint-based adaptation, the non-uniformly refined intermediate fine-mesh with $1,003,430$ nodes, $4,570$ surface quadrilaterals, $71,972$ surface triangles, $1,706,813$ prisms and $767,814$ tetrahedrals is used to establish the adaptive indicators. The wall spacing on the fine-mesh is same as the initial grid.

The convergence of C_D is shown in Figs.(5.28a) and (5.28b). The estimate of C_D from the adjoint-adapted grid with $478,952$ nodes is better than the estimates from the non-uniformly refined fine-mesh with $1,003,430$ nodes, and the uniformly refined mesh with 2.35×10^6 nodes. Also, note that the adapted grid C_D has surpassed the estimate of the fine-mesh employed to form the adaptive indicators. The feature-based approach has failed to make any improvements to C_D and in fact, there is loss of accuracy in C_D from the feature-adapted grid with $1,135,637$ nodes. From Fig.(5.28b) for the parallel CPU costs, it can be inferred that large savings in parallel cost may be attained with adjoint-based adaptation.

The surface grids on the upper wing surface for the initial, adjoint-adapted and feature-adapted grids are shown in Figs.(5.29), (5.30) and (5.33). The initial, adjoint-adapted and feature-adapted symmetry plane grids are shown in Figs.(5.31),(5.32) and (5.34). For the adjoint-adapted grid shown in Figs.(5.30) and (5.32), besides refinement near the leading edge, trailing edge and surface of the wing, there is moderate refinement in the wake regions, regions upstream of leading edge (outside boundary layer) and regions near the outer edge of boundary layer. The feature-adapted grid in Figs.(5.33) and (5.34) has considerable refinement near the leading and trailing edges of the wing and has moderate surface refinement. There is no significant refinement in rest of the regions.

The initial density contours on the upper and lower surfaces of the wing are shown in Figs.(5.35a) and (5.35b) and it can be noticed that the shock is smeared over lot of points. From the density contours of the adjoint-adapted grid in Figs.(5.36a) and (5.36b), it can be observed that the curvature of the shock is captured well and there is a crisper shock compared to the initial grid. However, in the density contours of the feature-adapted grid shown in Figs.(5.37a) and (5.37b), the shock curvature is less captured, and there is no significant reduction in the smearing of shock compared to the initial grid. The pressure contours on the symmetry plane of the initial grid is shown in Fig.(5.38). The shocks are poorly resolved outside the boundary layer and in the inviscid regions. In the pressure contours of the adjoint-adapted grid shown in Fig.(5.39), there is less smearing of the shocks and their resolution in the inviscid regions have improved greatly. There is no improvement in the pressure contours of the feature-adapted grid shown in Fig.(5.40). The feature adaptation has been handicapped by the poor resolution on the initial grid. However, this does not pose a problem for the adjoint adaptation. A better resolved initial grid may be needed for the feature adaptation.

Figs.(5.41), (5.42) and (5.43) show the Mach number contours on the symmetry plane for the initial, adjoint-adapted and feature-adapted grids. The Mach number contours are plotted here to see the resolution of the separation zone in the boundary layer. From Figs.(5.41), (5.42) and (5.43), the thickening of the boundary layer because of the shocks can be noticed. Figs.(5.44) and (5.45)

show the Mach number contours near the trailing edge for the adjoint-adapted and feature-adapted grids. It can be observed that the separation zone is better resolved by the adjoint adaptation, while, it is poorly resolved in the feature approach. The feature approach predicts a large separation with the flow separating immediately after the shock. But, the actual separation occurs further downstream close to the trailing edge as predicted by the adjoint adaptation. The loss of accuracy in the feature-adapted grid may be attributed to the poor resolution of the separation zone.

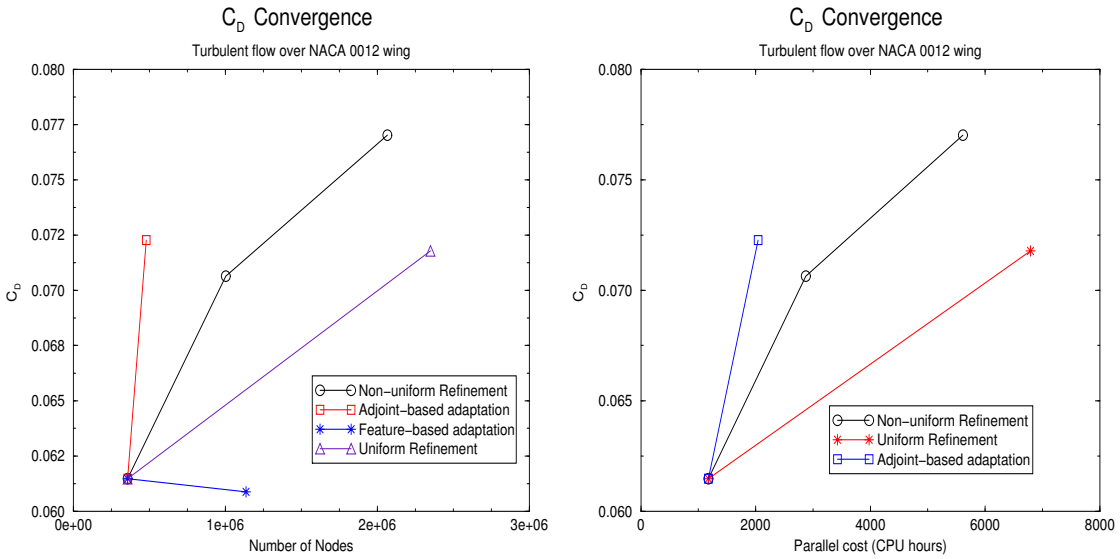


Figure 5.28: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. C_D convergence. (a) C_D Vs Number of Nodes. (b) C_D Vs Parallel cost

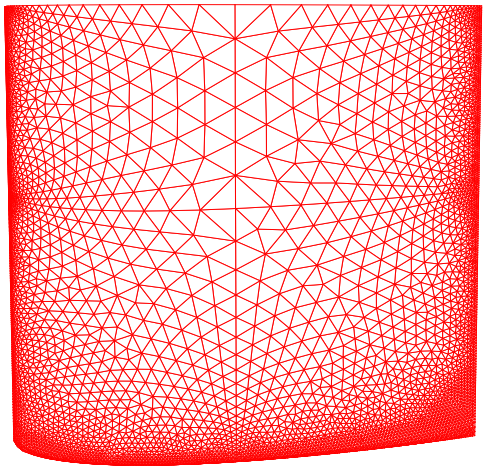


Figure 5.29: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Initial surface grid on upper wing (Total Nodes: $356,420$).

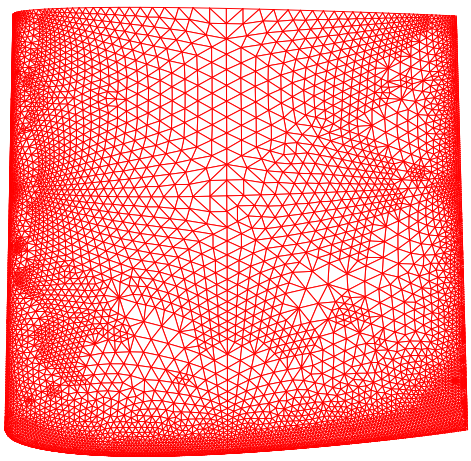


Figure 5.30: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Surface grid on upper wing after one adjoint-C_D adaptation (Total Nodes: $478,952$).

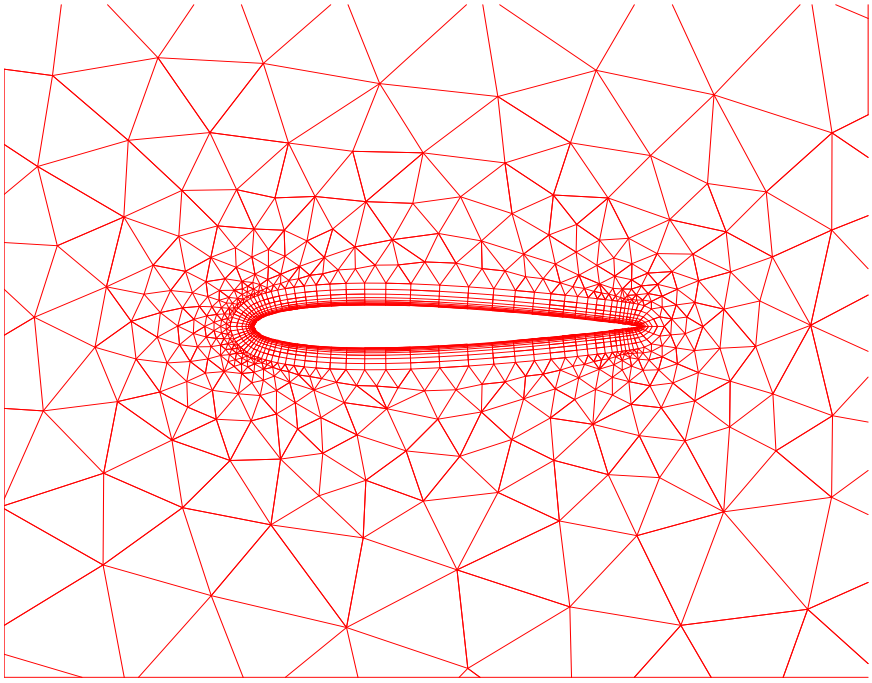


Figure 5.31: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Initial symmetry plane grid (Total Nodes: $356,420$).

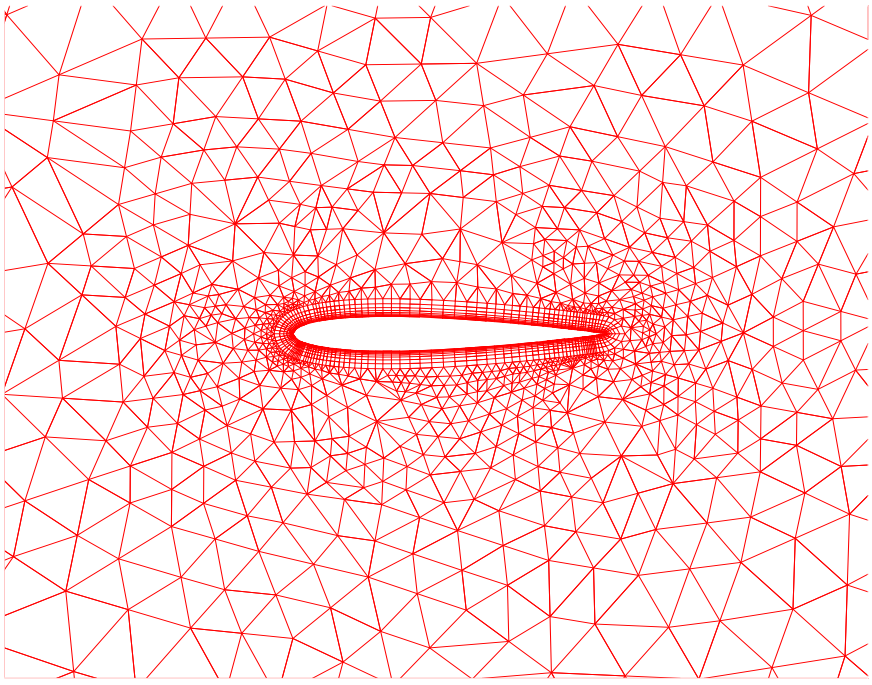


Figure 5.32: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Symmetry plane grid after one adjoint-C_D adaptation (Total Nodes: $478,952$).

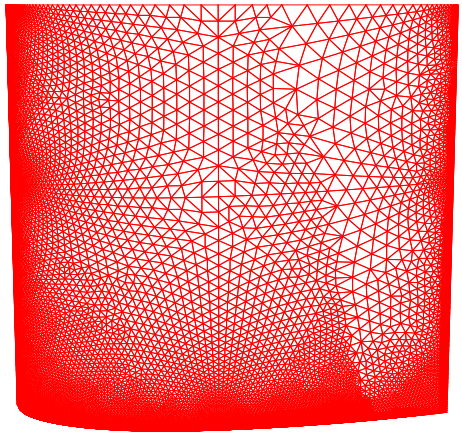


Figure 5.33: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Surface grid on upper wing after one feature adaptation (Total Nodes: $1,135,637$).

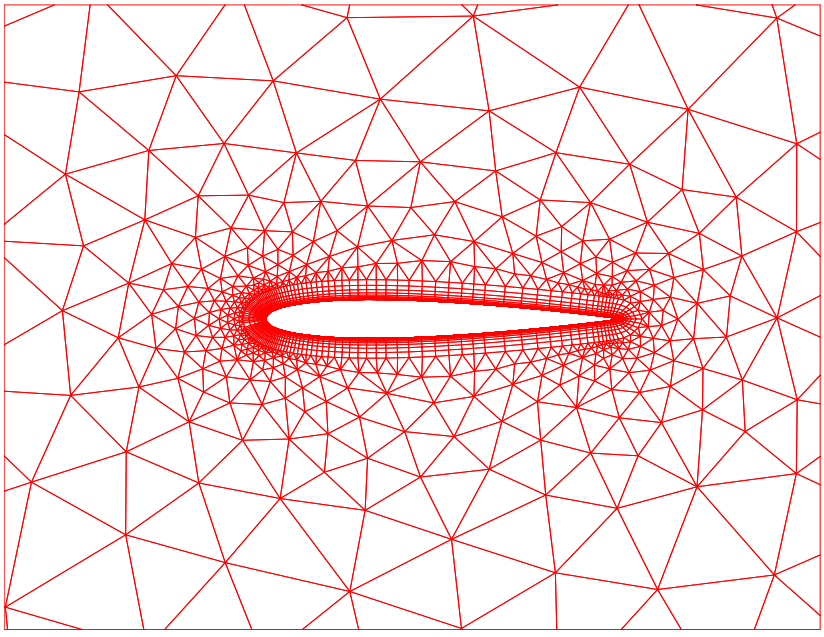
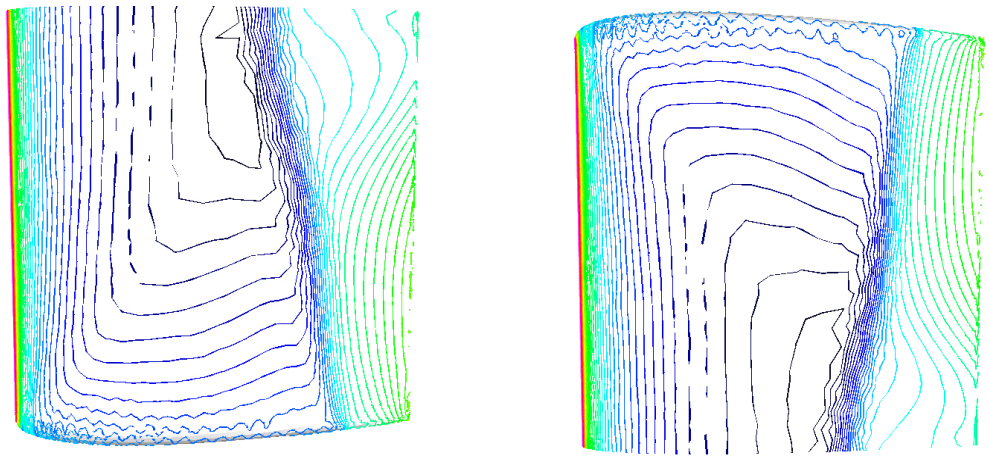
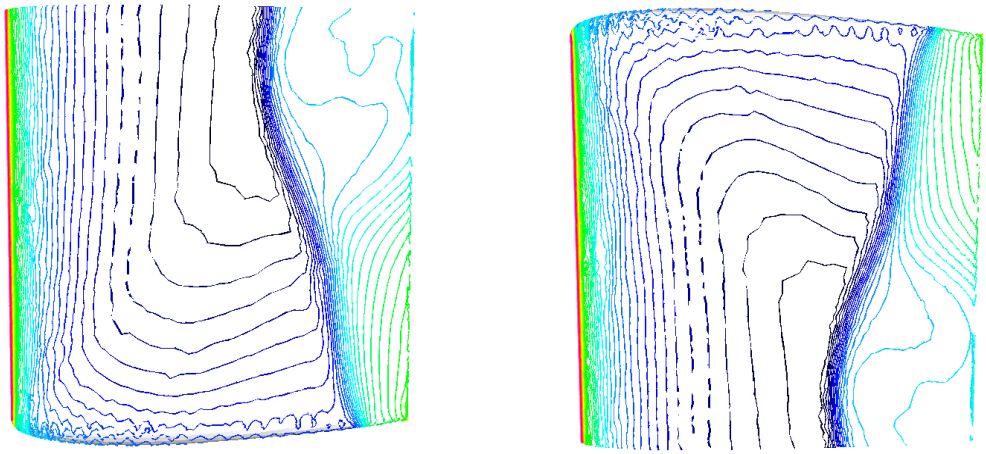


Figure 5.34: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Symmetry plane grid after one feature adaptation (Total Nodes: $1,135,637$).



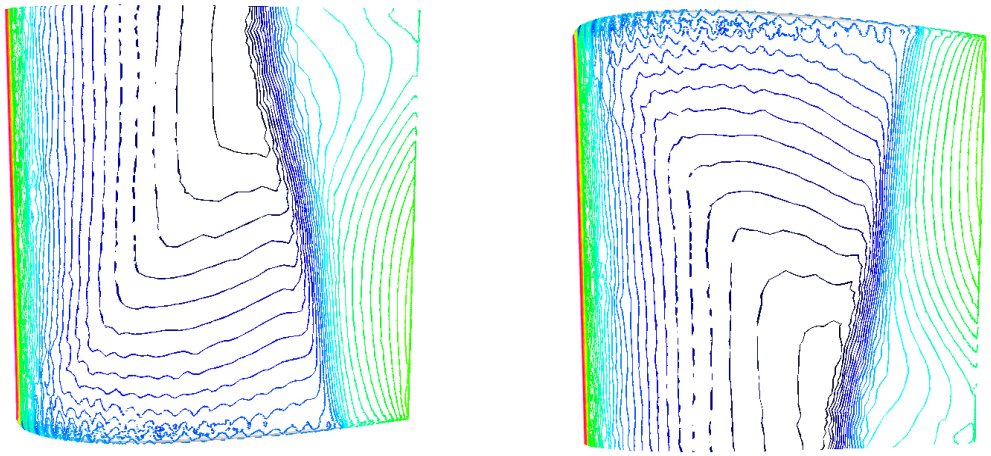
(a) (b)

Figure 5.35: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Initial grid. (a) upper wing surface density contours; (b) lower wing surface density contours.



(a) (b)

Figure 5.36: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Adjoint-adapted grid. (a) upper wing surface density contours; (b) lower wing surface density contours.



(a) (b)

Figure 5.37: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Feature-adapted grid. (a) upper wing surface density contours; (b) lower wing surface density contours.

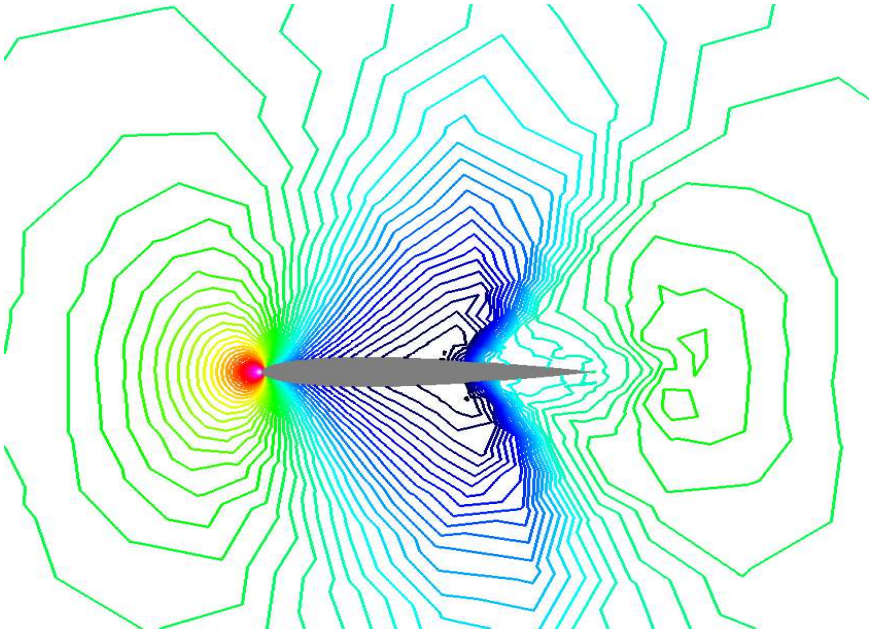


Figure 5.38: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Pressure contours on the symmetry plane for the initial grid.

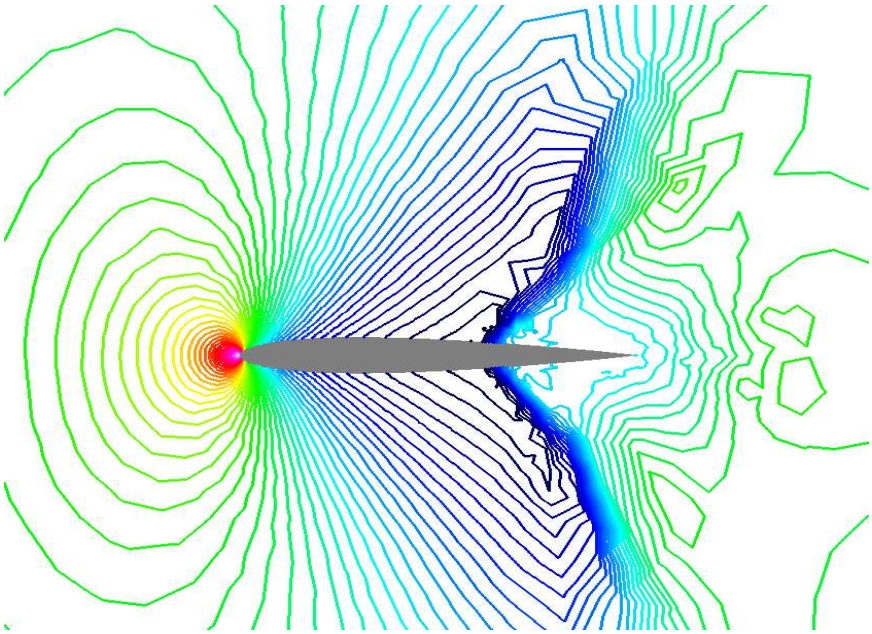


Figure 5.39: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Pressure contours on the symmetry plane for the adjoint-adapted grid.

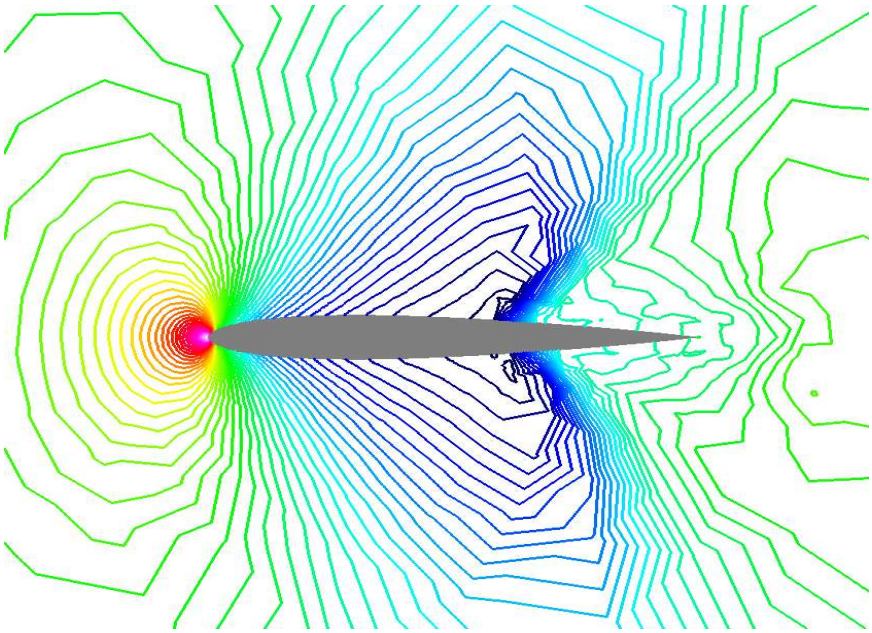


Figure 5.40: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.52$, $AOA = 0^0$, $Re = 3,000,000$. Pressure contours on the symmetry plane for the feature-adapted grid.

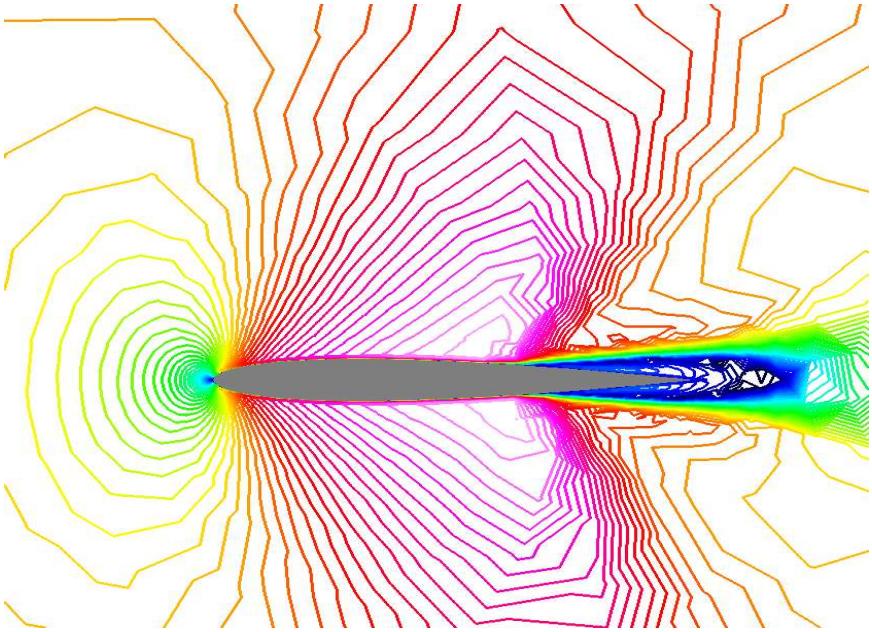


Figure 5.41: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Mach number contours on the symmetry plane for the initial grid.

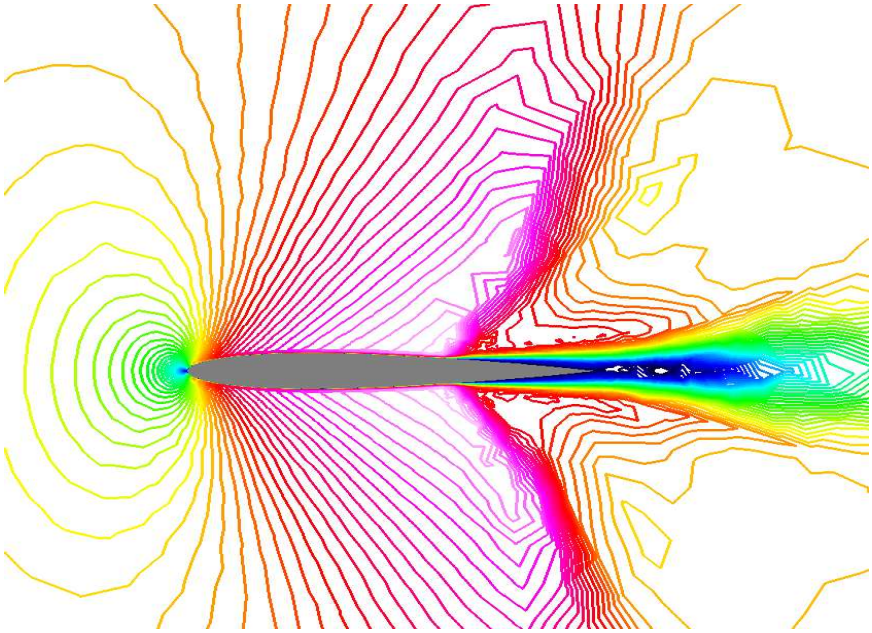


Figure 5.42: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Mach number contours on the symmetry plane for the adjoint-adapted grid.

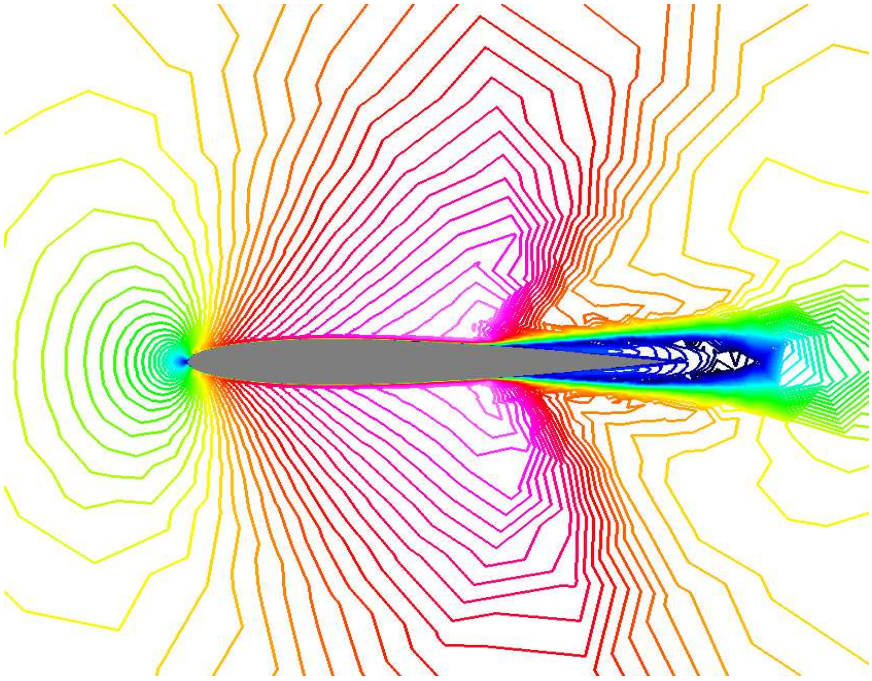


Figure 5.43: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Mach number contours on the symmetry plane for the feature-adapted grid.

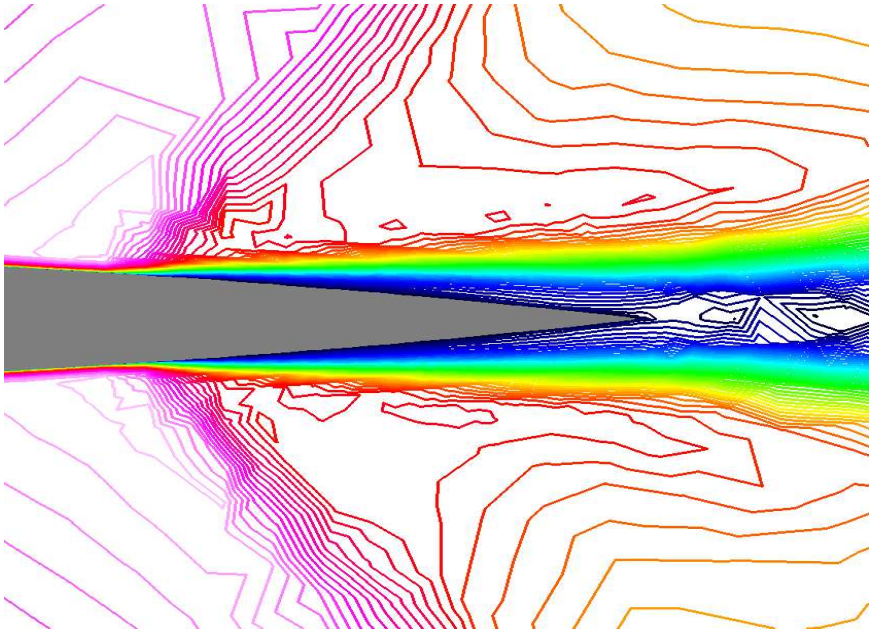


Figure 5.44: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Mach number contours near the trailing edge on the symmetry plane for the adjoint-adapted grid.

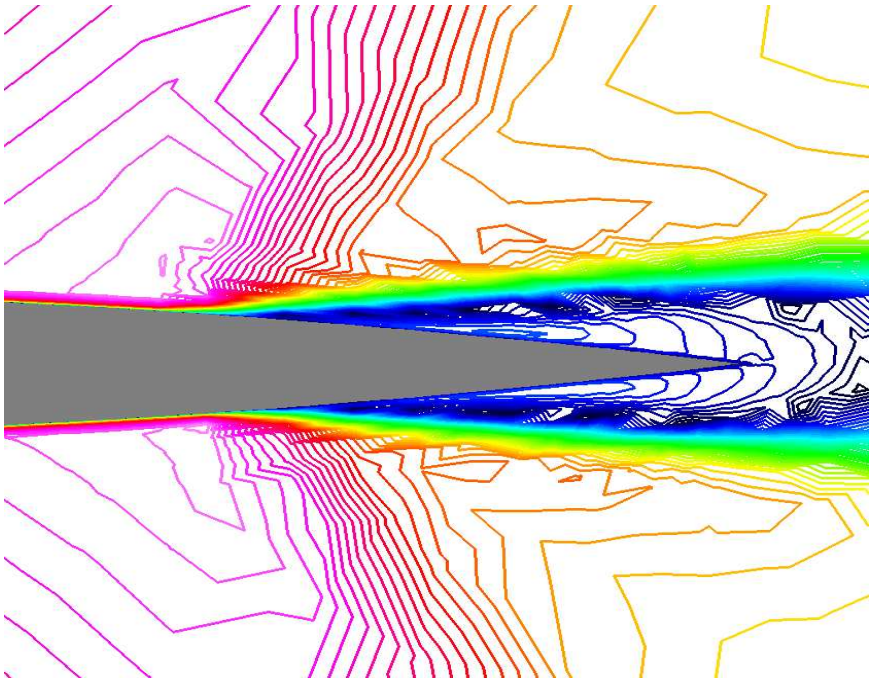


Figure 5.45: Turbulent flow over NACA 0012 rectangular wing: $M_\infty = 0.95$, $AOA = 0^0$, $Re = 3,000,000$. Mach number contours near the trailing edge on the symmetry plane for the feature-adapted grid.

CHAPTER VI

SUMMARY AND RECOMMENDATIONS

An adjoint-based error estimation methodology has been presented that provides a quantitative measure of the error in computed outputs and improves the computed accuracy of functional outputs. The error estimates relate the local residual errors to the global error in output function via adjoint variables as weight functions. The major steps of the error estimation methodology: (1) development of adjoint sensitivity analysis capabilities; (2) development of an efficient and robust error estimation procedure; (3) implementation of an output-based grid adaptive scheme have been accomplished in this work.

In the first step, parallel discrete direct and adjoint sensitivity analysis capabilities have been developed for variable Mach number flows on mixed-element unstructured meshes. The parallel implementation is based on coarse-grained domain decomposition and has been shown to scale well. Results for several cases validate the consistency of linearization over sequential and parallel runs. A modification has been proposed to the current class of unstructured flux limiters to make them piecewise continuous and suitable for discrete sensitivity analysis. The modified limiters are essentially a weak form of the original limiters and have been found to be more dispersive from numerical tests. The current study is perhaps the first work to accomplish a consistent and complete linearization of limiters for discrete sensitivity analysis. An improved version of Symmetric Gauss Seidel scheme and its exact dual algorithm have been implemented. The algorithms exhibit identical asymptotic convergence rates, demonstrating numerical equivalence between the direct and adjoint discretizations.

The second step required prolongation of flow and adjoint solutions from a coarse-mesh to a fine-mesh to compute the error estimates. Smooth reconstruction of the coarse-mesh solutions has been accomplished using the meshless Moving Least Squares (MLS) approximation. The MLS

procedure has been found to compute highly accurate corrections on the fine-mesh. A thorough investigation of the error correction procedure has been performed based on the choice of (a) basis functions: linear basis for both flow and adjoint solutions (linear-linear), quadratic basis for both flow and adjoint solutions (quadratic-quadratic), linear basis for flow solution and quadratic basis for adjoint solution (linear-quadratic), quadratic basis for flow solution and linear basis for adjoint solution (quadratic-linear); (b) weight functions: cubic spline or inverse-distance; and (c) type of support: circular, rectangular or mixed.

The linear-quadratic and quadratic-linear MLS fits have been helpful to understand the quality of linear/quadratic approximations for the flow and adjoint solutions and to identify the source of bad approximations in linear-linear and quadratic-quadratic fits. The cubic spline weights has been found to produce smoother approximations compared to the inverse-distance weights, resulting in slightly better corrections. Circular supports are found effective in inviscid regions (typically, tetrahedrals and pyramids) and rectangular supports are found effective in boundary layer regions (typically, prisms and hexahedrals). Circular supports should be the choice for tetrahedral meshes and mixed (circular-rectangular) supports should be the choice for mixed-element meshes. Error correction results presented for inviscid, laminar and turbulent flows demonstrate the robustness of the developed error estimation procedure in improving functional accuracy. Also, this is the first work to demonstrate the error correction procedure on mixed-element unstructured meshes and both uniformly and non-uniformly refined fine-meshes.

In the final step, adjoint-based and feature-based adaptive strategies have been implemented to improve the accuracy of the chosen output to a prescribed tolerance. Grid adaptation results presented for inviscid, laminar and turbulent flows demonstrate the robustness of the adjoint-based approach over the feature-based approach. In all the adjoint adaptation cases presented, the same level of functional accuracy has been accomplished with a much smaller mesh size (typically a factor of 3 to 5 reduction in mesh size) compared to the uniformly and non-uniformly refined fine-meshes. Also, significant savings in parallel CPU cost has been achieved by performing adjoint adaptation. The feature approach has suffered by a poor resolution of the initial grid and failed to

make significant improvements to functional accuracy. However, the resolution of the initial grid did not pose a problem for the adjoint adaptation. A better resolved initial grid is needed by the feature-based approach.

The first recommendation for future work is to improve the stability of the adjoint solver for turbulent flows. In the present study, the loosely coupled implementation of the turbulence model in the flow solver sometimes resulted in stalled convergence or limit-cycle oscillations. This has been detrimental to the convergence of sensitivity analysis solver. Future work may be to develop a tightly coupled turbulence model along the lines of [21] or stabilize the adjoint calculations by using the standard solver as a preconditioner with outer GMRES (Generalized Minimal Residual) or RPM (Recursive Projection Method) iterations along the lines of [102, 103].

In the present study, the error correction procedure is sequential, and because of memory constraints, complex geometries and large mesh sizes could not be handled. The second recommendation for future work is to perform error estimation in parallel to handle large real life applications. Parallel error estimation can be implemented by uniformly refining the mesh in each individual partition and performing error estimation locally in each partition with due care taken at the block boundaries. The error estimates in each individual partition can then be post-processed based on the needs of the adaptation module.

The final recommendation for future work is to develop a unified framework for direct interface of the CAD models to CFD software. By facilitating direct access to the CAD model, mesh movement and mesh adaptation can be easily integrated with the CFD software, and this will greatly improve design optimization capabilities. Furthermore, at a larger level, multidisciplinary analysis and design can be performed with the CAD model serving as a common geometry description.

REFERENCES

[1] E. Turkel, "Review of preconditioning methods for fluid dynamics," *Applied Numerical Mathematics*, vol. 12, pp. 257–284, 1993.

[2] E. Turkel, "Preconditioning techniques in computational fluid dynamics," *Annual Review of Fluid Mechanics*, vol. 31, pp. 385–416, 1999.

[3] W. R. Briley, H. McDonald, and S. J. Shamroth, "A low Mach number Euler formulation to time iterative LBI schemes," *AIAA Journal*, vol. 21, no. 10, pp. 1467–1469, 1983.

[4] W. R. Briley, L. K. Taylor, and D. L. Whitfield, "High-resolution viscous flow simulations at arbitrary Mach number," *Journal of Computational Physics*, vol. 184, pp. 79–105, 2003.

[5] W. Squire and G. Trapp, "Using complex variables to estimate derivatives of real functions," *SIAM Rev.*, vol. 40, no. 1, pp. 110–112, 1998.

[6] J. C. Newman, III, D. L. Whitfield, and W. K. Anderson, "Multidisciplinary Sensitivity Derivatives Using Complex Variables," tech. rep., Mississippi State University, July 1998. Tech. Rep. MSSU-COE-ERC-98-08.

[7] J. C. Newman, III, D. L. Whitfield, and W. K. Anderson, "A step-size independent approach for multidisciplinary sensitivity analysis," *Journal of Aircraft*, vol. 40, no. 3, pp. 566–573, 2003.

[8] J. R. R. A. Martins, I. M. Kroo, and J. J. Alonso, "An automated method for sensitivity analysis using complex variables," 2000. AIAA 2000-0689.

[9] J. C. Newman, III, A. C. Taylor, III, R. C. Barnwell, P. A. Newman, and G. J.-W. Hou, "Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations," *Journal of Aircraft*, vol. 36, no. 1, pp. 87–96, 1999.

[10] A. Jameson, "Aerodynamic design via control theory," *Journal of Scientific Computing*, vol. 3, pp. 233–260, 1988.

[11] B. Soemarwoto, "The Variational Method for Aerodynamic Optimization using the Navier-Stokes Equations," tech. rep., ICASE, Institute for Computer Applications in Science and Engineering, MS 132C, NASA Langley Research Center, Hampton, VA 23681-0001., December 1997. Report no. 97-71.

[12] W. K. Anderson and V. Venkatakrishnan, "Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation," 1997. AIAA 97-0643.

[13] S. K. Nadarajah and A. Jameson, "A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization," 2000. AIAA 2000-0667.

[14] S. K. Nadarajah and A. Jameson, "Studies of the continuous and discrete adjoint approach to viscous automatic aerodynamic shape optimization," 2001. AIAA 2001-2530.

[15] M. B. Giles, "Adjoint methods for aeronautical design," tech. rep., ECCOMAS, September 2001. Computational Fluid Dynamics Conference.

[16] G. J.-W. Hou, A. C. Taylor, III, and V. M. Korivi, "Discrete Shape Sensitivity Equations for Aerodynamic Problems," *International Journal of Numerical Methods in Engineering*, vol. 37, pp. 2251–2266, 1994.

[17] J. C. Newman, III, *Integrated Multidisciplinary Design Optimization Using Discrete Sensitivity Analysis for Geometrically Complex Aeroelastic Configurations*. PhD thesis, Virginia Polytechnic Institute and State University, July 1997.

[18] C. O. E. Burg and J. C. Newman, III, "Computationally efficient, numerical exact design space derivatives via the complex Taylor's series expansion method," *Computers and Fluids*, vol. 32, pp. 373–383, 2003.

[19] E. J. Nielsen, *Aerodynamic Design Sensitivities on an Unstructured Mesh using the Navier-Stokes Equations and a Discrete Adjoint Formulation*. PhD thesis, Virginia Polytechnic Institute and State University, December 1998.

[20] E. J. Nielsen and W. K. Anderson, "Recent Improvements in Aerodynamic Design Optimization On Unstructured Meshes," *AIAA Journal*, vol. 40, no. 6, pp. 1155–1163, 2002. Also, AIAA-2001-0596, January 2001.

[21] E. J. Nielsen, J. Lu, M. A. Park, and D. L. Darmofal, "An implicit, exact dual adjoint solution method for turbulent flows on unstructured grids," *Computers and Fluids*, vol. 33, pp. 1131–1155, 2004.

[22] E. J. Nielsen and M. A. Park, "Efficient Construction of Discrete Adjoint Operators on Unstructured Grids by Using Complex Variables," 2005. AIAA 2005-0324.

[23] E. J. Nielsen and M. A. Park, "Using An Adjoint Approach to Eliminate Mesh Sensitivities in Computational Design," 2005. AIAA 2005-0491.

[24] S. K. Nadarajah, *Discrete Adjoint Formulation*. PhD thesis, Department of Aeronautics and Astronautics, Stanford University, January 2003.

[25] M. B. Giles, M. C. Dutta, and J.-D. Müller, "Adjoint Code Developments Using the Exact Discrete Approach," 2001. AIAA 2001-2596.

[26] M. B. Giles, M. C. Dutta, and J.-D. Müller, "Algorithm developments for discrete adjoint methods," *AIAA Journal*, vol. 41, no. 2, pp. 198–205, 2003.

[27] D. J. Mavriplis, "Formulation and Multigrid Solution of the Discrete Adjoint for Optimization Problems on Unstructured Meshes," 2005. AIAA 2005-0319.

[28] R. Balasubramanian and J. C. Newman, III, "Discrete Direct and Adjoint Sensitivity Analysis for Variable Mach number Flows," *International Journal of Numerical Methods in Engineering*. Accepted for publication, September 2005. Early View: *http://www3.interscience.wiley.com/cgi-bin/jissue/108567238*. DOI: 10.1002/nme.1558.

[29] R. Balasubramanian and J. C. Newman, III, "Discrete Sensitivity Analysis for Variable Mach number Flows." SIAM Conference on Computational Science and Engineering, February 2005.

[30] C. Burdyshaw and W. K. Anderson, "A General and Extensible Unstructured Mesh Adjoint Method," 2005. AIAA 2005-0335.

[31] G. Strang and G. Fix, *An analysis of the Finite Element Method*. Prentice Hall, 1973.

[32] I. Babuška and W. C. Rheinboldt, "Error estimates for adaptive finite element computations," *SIAM Journal of Numerical Analysis*, vol. 15, no. 4, pp. 736–754, 1978.

[33] I. Babuška and A. Miller, "The post-processing approach in the finite element method - Part 1: calculation of displacements, stresses and other higher derivatives of the displacements," *International Journal of Numerical Methods in Engineering*, vol. 20, pp. 1085–1109, 1984.

[34] R. Becker and R. Rannacher, "Weighted A Posteriori Error Control in Finite Element Methods," tech. rep., Universitat Heidelberg, 1996. Preprint No. 1996-1. *http://gaia.iwr.uni-heidelberg.de*.

[35] R. Becker and R. Rannacher, "A Feed-Back Approach to Error Control in Finite Element Methods: Basic Analysis and Examples," tech. rep., Universitat Heidelberg, 1996. Preprint No. 1996-8. *http://gaia.iwr.uni-heidelberg.de*.

[36] R. Becker and R. Rannacher, "An optimal control approach to a posteriori error estimation in finite element methods," tech. rep., Universitat Heidelberg, 2001. Preprint No. 2001-3. *http://gaia.iwr.uni-heidelberg.de*.

[37] E. Süli, "A posteriori error analysis and global error control for adaptive finite element approximations of hyperbolic problems," in *Proceeding of the 16th Biennial Conference in Numerical Analysis*, 1996.

[38] E. Süli and P. Houston, *Lecture Notes in Computational Science and Engineering: Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, ch. Adaptive finite element approximations of hyperbolic problems. Springer, 2000. Eds: T. J . Barth and H. Deconinck.

[39] J. Peraire and A. T. Patera, "Bounds for linear-functional outputs of coercive partial differential equations: Local indicators and adaptive refinement," in *Proc. of the Workshop On New Advances in Adaptive Computational Methods in Mechanics*, Elsevier, 1997. Eds. P.Ladeveze and J.T. Oden.

[40] M. Paraschivoiu, J. Peraire, and A. T. Patera, "A posteriori finite element bounds for linear-functional outputs of elliptic partial differential equations," *Comput. Methods Appl. Mech. Engrg.*, vol. 150, pp. 289–312, 1997.

[41] M. B. Giles, “On adjoint equations for error analysis and optimal grid adaptation in cfd,” tech. rep., Oxford University Computing Laboratory, Numerical Analysis Group, Oxford, England, 1997. Tech. rep. 97/11.

[42] M. B. Giles and N. A. Pierce, “Improved lift and drag estimates using adjoint Euler equations,” 1999. AIAA 1999-3293.

[43] M. B. Giles and N. A. Pierce, “Adjoint recovery of superconvergent functionals from pde approximations,” *SIAM Rev.*, vol. 42, no. 2, pp. 247–264, 2000.

[44] M. B. Giles and N. A. Pierce, “Adjoint error correction for integral outputs,” tech. rep., Oxford University Computing Laboratory, Numerical Analysis Group, Oxford, England, 2001. Tech. rep. 01/18.

[45] M. B. Giles, N. A. Pierce, and E. Süli, “Progress in adjoint error correction for integral functionals,” *Computing and Visualisation in Science*, vol. 6, pp. 113–121, 2004.

[46] D. A. Venditti and D. L. Darmofal, “A multilevel error estimation and grid adaptive strategy for improving the accuracy of integral outputs,” 1999. AIAA 1999-3292.

[47] D. A. Venditti and D. L. Darmofal, “Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow,” *Journal of Computational Physics*, vol. 164, pp. 204–227, 2000.

[48] D. A. Venditti, *Grid adaptation for functional outputs of compressible flow simulations*. PhD thesis, Massachusetts Institute of Technology, June 2002.

[49] M. A. Park, “Adjoint-based, three-dimensional error prediction and grid adaptation,” 2002. AIAA 2002-3286.

[50] M. A. Park, “Three-Dimensional Turbulent RANS Adjoint-Based Error Correction,” 2003. AIAA 2003-3849.

[51] R. Balasubramanian, “Error estimation and grid adaptation for functional outputs using discrete adjoint sensitivity analysis,” Master’s thesis, Mississippi State University, December 2002.

[52] M. B. Giles, *Computational Fluid Dynamics 2000*, ch. Defect and adjoint error correction. Springer-Verlag, 2001. Ed. N. Satofuka.

[53] M. B. Giles and N. A. Pierce, *Lecture Notes in Computational Science and Engineering: Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, ch. Adjoint error correction for functional outputs. Springer, 2000. Eds: T. J . Barth and H. Deconinck.

[54] N. A. Pierce and M. B. Giles, “Adjoint and defect bounding and correction for functional estimates,” *Journal of Computational Physics*, vol. 200, pp. 769–794, 2004.

[55] J. F. Thompson, B. K. Soni, and N. P. Weatherill, *Handbook of Grid Generation*. CRC Press, 1999.

[56] T. J. Barth, "Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations," in *VKI Lecture Series 1994-05*, (von Karman Institute for Fluid Dynamics), 1994.

[57] S. Arabshahi and D. L. Whitfield, "A multi-block approach to the three-dimensional unsteady Euler equations about a wing-pylon-store configuration," 1989. AIAA 1989-3401.

[58] D. J. Mavriplis, "Unstructured mesh generation and adaptivity," tech. rep., ICASE, Institute for Computer Applications in Science and Engineering, MS 132C, NASA Langley Research Center, Hampton, VA 23681-0001., December 1997. Report no. 95-26.

[59] D. J. Mavriplis, "Unstructured grid techniques," *Annu. Rev. Fluid Mech.*, pp. 473–514, 1997.

[60] N. P. Weatherill, O. Hassan, D. L. Marcum, and M. J. Marchant, "Grid Generation by the Delaunay Triangulation," in *26th Computational Fluid Dynamics Lecture Series*, (von Karman Institute for Fluid Dynamics), January 1994.

[61] T. J. Baker, "Mesh Adaptation Strategies for Problems in Fluid Dynamics," *Finite Elements in Analysis and Design*, vol. 25, pp. 243–273, 1997.

[62] N. P. Weatherill, O. Hassan, M. J. Marchant, and D. L. Marcum, "Adaptive inviscid flow solutions for aerospace geometries on efficiently generated unstructured tetrahedral meshes," 1993. AIAA 1993-3390.

[63] M. J. Marchant and N. P. Weatherill, "Adaptive techniques for compressible inviscid flows," *Journal of Numerical Methods in Applied Mechanics and Engineering*, vol. 106, pp. 83–106, 1993.

[64] N. P. Weatherill and M. J. Marchant, "Grid adaptation using a distribution of sources applied to inviscid compressible flow simulations," *International Journal of Numerical Methods in Fluids*, vol. 19, pp. 739–764, 1994.

[65] D. L. Marcum and N. P. Weatherill, "A Procedure for Efficient Generation of Solution Adapted Unstructured Grids," in *Proceedings of the 4th International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fluids*, pp. 639–650, 1994. Eds. N. P. Weatherill and P. R. Eiseman and J. Hauser and J. F. Thompson.

[66] D. L. Marcum, "Adaptive Unstructured Grid Generation for Viscous Flow Applications," *AIAA Journal*, vol. 34, pp. 2440–2453, 1996.

[67] D. L. Marcum and P. G. Kelly, "Solution Adaptive Unstructured Grid Generation using Pseudo-Pattern Recognition Techniques," 1997. AIAA 1997-1860.

[68] S. Z. Pizadeh, "An adaptive unstructured grid method by grid subdivision, local remeshing and grid movement," 1999. AIAA 1999-3255.

[69] D. A. Venditti and D. L. Darmofal, "Grid adaptation for functional outputs of 2-D compressible flow simulations," 2000. AIAA 2000-2244.

[70] D. A. Venditti and D. L. Darmofal, "Grid adaptation for functional outputs: Application to two-dimensional inviscid flows," *Journal of Computational Physics*, vol. 176, pp. 40–69, 2002.

[71] D. A. Venditti and D. L. Darmofal, "Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows," *Journal of Computational Physics*, vol. 187, pp. 22–46, 2003.

[72] J.-D. M$\ddot{u}$ller and M. B. Giles, "Solution adaptive mesh refinement using adjoint error analysis," 2001. AIAA 2001-2550.

[73] D. G. Hyams, *An Investigation of Parallel Implicit Solution Algorithms for Incompressible Flows on Unstructured Topologies*. PhD thesis, Mississippi State University, May 2000.

[74] C. Sheng, J. C. Newman, III, M. Remotigue, J. P. Chen, D. L. Marcum, and D. L. Whitfield, "Development of unstructured computational capabilities to MSU-TURBO with an arbitrary Mach algorithm," tech. rep., Mississippi State University, 2002. Tech. Rep. MSSU-COE-ERC-02-16.

[75] T. J. Barth and D. C. Jespersen, "The Design and Application of Upwind Schemes on Unstructured meshes," 1989. AIAA 1989-0366.

[76] V. Venkatakrishnan, "On the Accuracy of Limiters and Convergence to Steady State Solutions," 1993. AIAA 1993-0880.

[77] D. L. Whitfield, "Lecture notes on Computational Fluid Dynamics II," Spring semester 2001. ASE 8423, Mississippi State University.

[78] P. Lancaster and K. Salkauskas, "Surfaces Generated by Moving Least Squares Methods," *Mathematics of Computation*, vol. 37, no. 155, pp. 141–158, 1981.

[79] R. Franke, "Scattered Data Interpolation: Tests of Some Methods," *Mathematics of Computation*, vol. 38, no. 157, pp. 181–200, 1982.

[80] R. Farwig, "Rate of Convergence of Shepard's Global Interpolation Formula," *Mathematics of Computation*, vol. 46, no. 174, pp. 577–590, 1986.

[81] R. Farwig, "Multivariate interpolation of arbitrarily spaced data by moving least squares methods," *Journal of Computational and Applied Mathematics*, vol. 16, pp. 79–93, 1986.

[82] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, "Meshless Methods: An Overview and Recent Developments," *Computer Methods in Applied Mechanics and Engineering*, vol. 139, pp. 3–47, 1996. special issue on Meshless Methods.

[83] D. Levin, "The approximation power of moving least squares," *Mathematics of Computation*, vol. 67, no. 224, pp. 1517–1531, 1998.

[84] M. W. Beall and M. S. Shephard, "A general topology-based mesh data structure," *Int. Journal of Num. Meth. Engng.*, vol. 40, no. 9, pp. 1573–1596, 1997.

[85] M. W. Beall, *An Object-Oriented Framework for the Reliable Automated Solution of Problems in Mathematical Physics*. PhD thesis, Rensselaer Polytechnic Institute, May 1999.

[86] J.-F. Remacle, B. K. Karamete, and M. S. Shephard, "Algorithm Oriented Mesh Database," tech. rep., SCOREC, Rensselaer Polytechnic Institute, 2000. Rep. 2000-5. *http://www.scorec.rpi.edu/cgi-bin/reports/GetByYear.pl?Year=2000*.

[87] D. J. Mavriplis, "Adaptive Meshing Techniques for Viscous Flow Calculations on Mixed Element Unstructured Meshes," tech. rep., ICASE, 1997. NASA CR-201675, ICASE Report No. 97-20.

[88] P. R. Spalart and S. R. Allmaras, "A One-Equation Turbulence Model for Aerodynamic Flows," 1992. AIAA 1992-0439.

[89] R. Balasubramanian and J. C. Newman, III, "Adjoint Error Estimation for Functional Outputs using Moving Least Squares Approximation," *AIAA Journal*, 2005. under preparation.

[90] R. Balasubramanian and J. C. Newman, III, "Grid Adaptation for Functional Outputs," *International Journal of Numerical Methods in Fluids*, 2005. In review.

[91] F. M. White, *Viscous Fluid Flow*. Mcgraw-Hill, 1974.

[92] P. R. Spalart and M. Shur, "On the Sensitization of Turbulence Models to Rotation and Curvature," *Aerospace Science and Technology*, pp. 297–302, May 1997.

[93] P. Roe, "Approximate riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, vol. 23, pp. 357–372, 1981.

[94] V. M. Korivi, A. C. Taylor, III, P. A. Newman, G. J. W. Hou, , and H. E. Jones, "An approximate factored incremental strategy for calculating consistent discrete cfd sensitivity derivatives," *Journal of Computational Physics*, vol. 113, pp. 336–346, 1994.

[95] G. J.-W. Hou, V. Maroju, A. C. Taylor, III, V. M. Korivi, and P. A. Newman, "Transonic Turbulent Airfoil Design Optimization Using Automatic Differentiation in Incremental Iterative Form," 1995. AIAA 1995-1692.

[96] L. L. Sherman, A. C. Taylor, III, L. L. Green, P. A. N. amd G. J.-W. Hou, and V. M. Korivi, "First- and Second-Order Aerodynamic Sensitivity Derivatives Via Automatic Differentiation with Incremental Iterative Methods," *Journal of Computational Physics*, vol. 129, pp. 307–331, 1994.

[97] V. Schmitt and F. Charpin, "Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers," tech. rep., 1979. Experimental Data Base for Computer Program Assessment. Report of the Fluid Dynamics Panel Working Group 04, AGARD AR 138, May 1979. *http://www.grc.nasa.gov/WWW/wind/valid/m6wing/m6wing.html*.

[98] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal of Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.

[99] G. Karypis and V. Kumar, "METIS: Family of multilevel partitioning algorithms." *http://www-users.cs.umn.edu/ karypis/metis/index.html*.

[100] D. L. Marcum, "Advancing Front/Local Reconnection Grid Generator." Mississippi State University.

[101] G. M. Nielson, "The side-vertex method for interpolation in triangles," *J Approx. Theory*, vol. 25, pp. 318–336, 1979.

[102] M. S. Campobasso and M. B. Giles, *Computational Fluid Dynamics 2002*, ch. Stabilization of a linearized Navier-Stokes solver for turbomachinery aeroelasticity. Springer-Verlag, 2003.

[103] M. S. Campobasso and M. B. Giles, "Effect of flow instabilities on the linear analysis of turbomachinery aeroelasticity," *AIAA Journal of Propulsion and Power*, vol. 19, no. 2, 2003.

[104] W. H. Press, S. A. Teukolsky, W. T. Vellerling, and B. P. Flannery, *Numerical Recipes in C: The Are of Scientific Computing*. Cambridge University Press, 1992.

[105] P. S. Patel, *Automatic CAD Model Processing for Downstream Applications*. PhD thesis, Mississippi State University, December 2005.

APPENDIX A

MOVING LEAST SQUARES APPROXIMATION

The moving least squares (MLS) procedure was proposed by Lancaster and Salkauskas [78] for performing smooth interpolation of scattered data. The idea is to start with a weighted least squares fit for an arbitrary data point in the domain, and then move this point over the entire parameter domain, where a weighted least squares fit is computed and evaluated for each point individually. By choosing appropriate weight functions, this local weighted least squares fit results in a global approximation with good smoothing properties. A detailed study of the interpolating property of the MLS methods can be found in [79–81]. More recently, MLS approximations have become popular in *meshless* or *element free Galerkin* methods. An excellent overview of the MLS approximation to meshless methods can be found in [82]. The MLS procedure used in the present study is based on the meshless approximation explained in [82]. In the following sections, the MLS procedure is derived, the weight functions are introduced and the approach to build a compact support is explained.

A.1 MLS Formulation

Consider the following form of approximation, where $\mathbf{x}$ represent the co-ordinates of the points and u, the data at these points. The MLS approximation can be written as

$$u^h(\mathbf{x}) = \sum_{i=1}^{m} p_i(\mathbf{x}) a_i(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{A.1}$$

Here, $u^h(\mathbf{x})$ is the MLS approximation at $\mathbf{x}$, $p_i(\mathbf{x})$ are the monomial basis functions, m is the number of terms in the polynomial basis, and $a_i(\mathbf{x})$ are the coefficients. The common basis functions in three dimensions are

linear basis

$$\mathbf{p}^T(\mathbf{x}) = (1, x, y, z) \tag{A.2}$$

quadratic basis

$$\mathbf{p}^T(\mathbf{x}) = \left(1, x, y, z, xy, yz, zx, x^2, y^2, z^2\right) \tag{A.3}$$

A local approximation of $u^h(\mathbf{x})$ can be derived based on [78, 82]

$$u^h(\mathbf{x}, \mathbf{x}_I) = \sum_{i=1}^{m} p_i(\mathbf{x}_I) a_i(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}_I)\mathbf{a}(\mathbf{x}) \tag{A.4}$$

and the least squares problem can be defined as

$$\begin{aligned} F &= \sum_I w(\mathbf{x} - \mathbf{x}_I) \left(u^h(\mathbf{x}, \mathbf{x}_I) - u(\mathbf{x}_I) \right)^2 \\ &= \sum_I w(\mathbf{x} - \mathbf{x}_I) \left(\sum_i p_i(\mathbf{x}_I)\, a_i(\mathbf{x}) - u_I \right)^2 \end{aligned} \tag{A.5}$$

In the above equation, the polynomial basis $\mathbf{p}^T$ is defined at $\mathbf{x}_I$ and the unknown coefficients a_i are at $\mathbf{x}$ obtained from the weighted least squares fit for the local approximation. $w(\mathbf{x}-\mathbf{x}_I)$ are the weight functions with compact support given in section A.2. Writing Eqn.(A.5) as a least squares minimization problem:

find $\mathbf{a} = \{a_0, a_1, \cdots, a_n\}$ such that F(**a***) = min F(**a**). The coefficients $\mathbf{a}(\mathbf{x})$ can be obtained as

$$\frac{\partial F}{\partial \mathbf{a}} = 0 \tag{A.6}$$

$$\sum_I w(\mathbf{x} - \mathbf{x}_I) \left[\mathbf{p}(\mathbf{x}_I)\mathbf{p}^T(\mathbf{x}_I)\, \mathbf{a}(\mathbf{x}) - \mathbf{p}(\mathbf{x}_I) u(\mathbf{x}_I) \right] = 0 \tag{A.7}$$

Equation (A.7) can be written in matrix form as

$$\mathbf{P}^T\, \mathbf{W}\, \mathbf{P}\, \mathbf{a}(\mathbf{x}) - \mathbf{P}^T\, \mathbf{W}\mathbf{u} = 0 \tag{A.8}$$

where

$$\mathbf{u}^T = (u_0, u_1, \cdots, u_n) \tag{A.9}$$

$$\mathbf{P} = \begin{bmatrix} p_1(\mathbf{x}_1) & p_2(\mathbf{x}_1) & \cdots & p_m(\mathbf{x}_1) \\ p_1(\mathbf{x}_2) & p_2(\mathbf{x}_2) & \cdots & p_m(\mathbf{x}_2) \\ : & : & : & : \\ p_1(\mathbf{x}_n) & p_2(\mathbf{x}_n) & \cdots & p_m(\mathbf{x}_n) \end{bmatrix} \tag{A.10}$$

and

$$\mathbf{W} = \begin{bmatrix} w(\mathbf{x}-\mathbf{x}_1) & 0 & \cdots & 0 \\ 0 & w(\mathbf{x}-\mathbf{x}_2) & \cdots & 0 \\ : & : & : & : \\ 0 & 0 & \cdots & w(\mathbf{x}-\mathbf{x}_n) \end{bmatrix} \tag{A.11}$$

Equation (A.8) can be written as

$$\mathbf{P}^T\,\mathbf{W}\,\mathbf{P}\,\mathbf{a}(\mathbf{x}) = \mathbf{P}^T\,\mathbf{W}\mathbf{u} \tag{A.12}$$

$$[M]\,\mathbf{a}(\mathbf{x}) = [B]\,\mathbf{u}(\mathbf{x}_I) \tag{A.13}$$

where $[M] = \mathbf{P}^T\,\mathbf{W}\,\mathbf{P}$ is the moment matrix and $[B] = \mathbf{P}^T\,\mathbf{W}$. The coefficients can be obtained by a simple inverse of $[M]$ as

$$\mathbf{a}(\mathbf{x}) = [M]^{-1}\,[B]\,\mathbf{u} \tag{A.14}$$

By defining $\Phi^k(\mathbf{x}) = \{\phi_1^k(\mathbf{x}), \phi_2^k(\mathbf{x}), \cdots, \phi_n^k(\mathbf{x})\}$ as the MLS shape functions of order k, the MLS approximation in Eqn.(A.1) can be cast as

$$u^h(\mathbf{x}) = \sum_{I=1}^{n} \phi_I^k\, u_I \tag{A.15}$$

and the shape functions are given by

$$\Phi^k(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\,[M]^{-1}\,[B] \tag{A.16}$$

It is more common to use a scaled and locally defined polynomial basis instead of the globally defined $\mathbf{p}^T(\mathbf{x}_I)$ given by Eqns.(A.2) and (A.3). By defining the scaled coordinates as

$$\bar{x} = \frac{(x_I - x)}{\rho_1}, \quad \bar{y} = \frac{(y_I - y)}{\rho_2}, \quad \bar{z} = \frac{(z_I - z)}{\rho_3} \tag{A.17}$$

Eqns.(A.2) and (A.3) can be written as

linear basis

$$\mathbf{p}^T(\bar{\mathbf{x}}) = (1, \bar{x}, \bar{y}, \bar{z}) \tag{A.18}$$

quadratic basis

$$\mathbf{p}^T(\bar{\mathbf{x}}) = \left(1, \bar{x}, \bar{y}, \bar{z}, \bar{x}\bar{y}, \bar{y}\bar{z}, \bar{z}\bar{x}, \bar{x}^2, \bar{y}^2, \bar{z}^2\right) \tag{A.19}$$

Here, ρ_1, ρ_2 and ρ_3 define the lengths of the support explained in section A.2. To account for the scaling, the Eqn.(A.16) for the MLS shape functions is now written as

$$\Phi^k(\mathbf{x}) = \mathbf{p}^T(\mathbf{0})\,[M]^{-1}\,[B] \tag{A.20}$$

The MLS problem defined by Eqn.(A.5) typically results in an overdetermined system with more equations than unknowns. Also, the moment matrix $[M]$ in Eqn.(A.20) may be severely ill-conditioned on highly stretched meshes. To avoid numerical instability and have a stable algorithm, the least squares problem is solved using Singular Value Decomposition (SVD) [104].

A.2 Weight Functions

The MLS weight functions $w(\mathbf{x} - \mathbf{x}_I)$ are typically defined to have compact support; i. e., they are non-zero over a relatively small part of the entire domain. The sub-domain that defines the support is called the domain of influence. The most commonly used supports are circular (discs or balls) and rectangular (rectangles or bricks). Mixed (rectangular-circular) supports may also be used. Figs.(A.1), (A.2) and (A.3) show the circular, rectangular and mixed supports for a node I

in two dimensions. In Figs.(A.1), (A.2) and (A.3), only the nodes inside the sub-domain Ω_I will have influence on node I.

The commonly used weight functions in meshless approximations [82] are employed in this study. For circular supports, the weight functions are isotropic. By defining $s = \|\mathbf{x} - \mathbf{x}_I\|$; and $\bar{s} = s/\rho$, where $\rho = s_{max}$ is the radius of the support, the weights functions [82] can be defined as

cubic spline

$$w(\bar{s}) = \begin{cases} \frac{2}{3} - 4\bar{s}^2 + 4\bar{s}^3 & ; \bar{s} \leq \frac{1}{2} \\ \frac{4}{3} - 4\bar{s} + 4\bar{s}^2 - \frac{4}{3}\bar{s}^3 & ; \frac{1}{2} < \bar{s} \leq 1 \\ 0 & ; \bar{s} > 1 \end{cases} \tag{A.21}$$

inverse-distance

$$w(s) = \begin{cases} \frac{1}{s^2+\epsilon} & ; s \leq s_{max} \\ 0 & ; s > s_{max} \end{cases} \tag{A.22}$$

The above cubic spline and inverse-distance weights have been constructed to possess C^2 continuity. For, inverse-distance weights, ϵ is a penalty factor that forces the MLS fit to interpolate the data for $s = 0$. In the present study, a penalty factor of $\epsilon = 1e^{-12}$ is used. For rectangular supports, tensor product weights (also referred as anisotropic weights) can be defined as

$$w(\mathbf{x} - \mathbf{x}_I) = w(\frac{\|x - x_I\|}{\rho_1})\, w(\frac{\|y - y_I\|}{\rho_2})\, w(\frac{\|z - z_I\|}{\rho_3}) \tag{A.23}$$

where $\rho_1 = \|x - x_I\|_{max}$, $\rho_2 = \|y - y_I\|_{max}$ and $\rho_3 = \|z - z_I\|_{max}$ are the lengths of the rectangular support.

A.3 Optimal Selection of Support Stencil

Optimal selection of nodes for the support stencil play a crucial role in producing a good MLS approximation. The three main factors that influence the quality of MLS fit are: (1) support stencil must contain nodes that produce the best possible approximation; (2) support stencil must

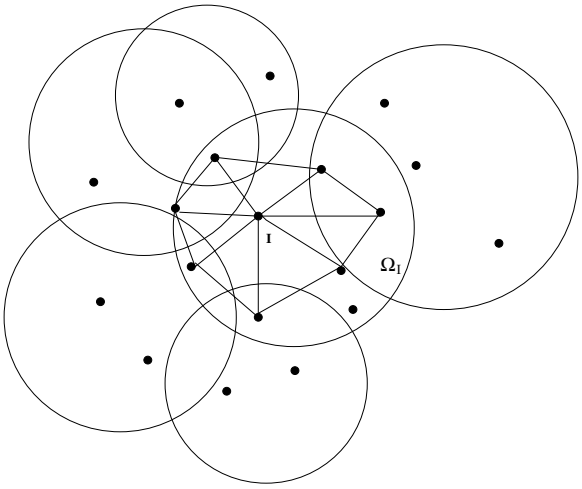


Figure A.1: Circular support for the MLS weight functions.

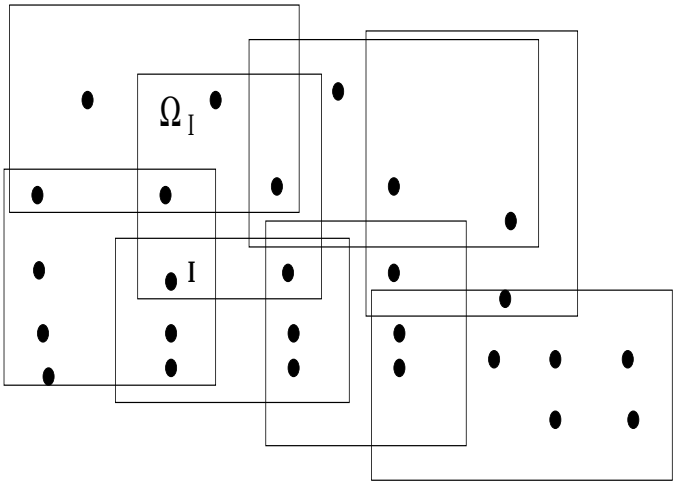


Figure A.2: Rectangular support for the MLS weight functions.

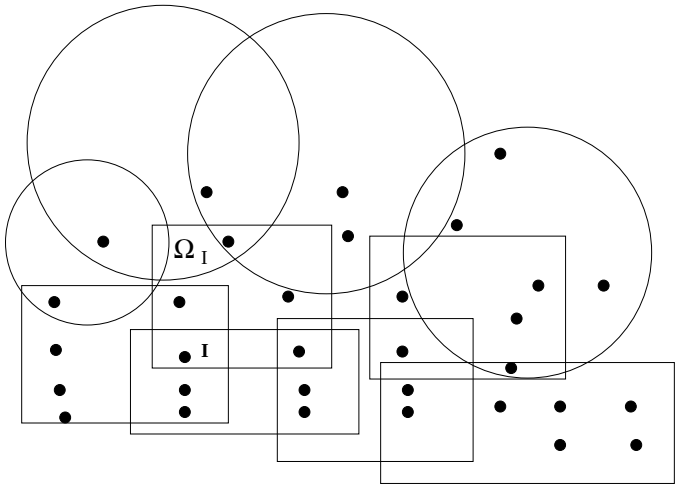


Figure A.3: Mixed (Circular-Rectangular) support for the MLS weight functions.

have enough nodes to avoid an ill-conditioned MLS system; (3) the supports must have sufficient overlap to ensure continuity of the approximation. In the present study, data values are interpolated from a coarse-mesh to fine-mesh and therefore, (3) is satisfied most of the times. The best way to satisfy (1) is to construct a stencil that includes nodes in the immediate vicinity of the node (say '*i*'), for which the approximation is sought. To accomplish (2), it is easy to expand the stencil with a large number of nodes. However, an extremely overdetermined system dilutes the approximation and results in a poor fit, thereby, not satisfying (1). The best way to satisfy (1) and (2) is to select nodes based on local configurations; test the stencil for ill-conditioned components using Singular Value Decomposition (SVD); and increase the stencil size if needed, to accomplish the requested order of approximation. However, this is extremely expensive and is not done in the present study.

In the present study, a global stencil size is chosen for building the support. Typically, a stencil size that is atleast twice the size of basis function produces a well-conditioned MLS system. For circular support, a stencil size of 9 is chosen for the linear basis, and a stencil size of 21 is chosen for the quadratic basis. For rectangular support, a stencil size of 6 is chosen for the linear basis, and a stencil size of 24 is chosen for the quadratic basis. These stencil sizes need not be optimum and they are chosen solely for a robust way to build the stencil. The stencil is built the following way for the circular and rectangular supports using a octree data structure developed by Patel [105].

For circular supports:

1. build a octree for the coarse-mesh coordinates.
2. perform a octree search to find the closest node to '*i*'.
3. using this distance as starting radius, perform a radial search and add nodes till the required stencil size is met.

For rectangular supports:

1. build a octree for the coarse-mesh element centroids.
2. perform a octree search to find the closest centroid to '*i*'.
3. with the closest element centroid as start, do a local neighborhood search to find the element containing '*i*'. If the containing element is not found by the local search, the closest element centroid is used as the containing element.
4. add all the nodes of containing element to the stencil; search stops here for linear basis.
5. do a directional search using the distances in x, y and z and add the adjacent element nodes till the required stencil size is met.

Since, the rectangular supports are in the boundary layer region, a 2:1 y-directional bias is used when building the support for velocities and turbulent quantity. For density and pressure, a stencil biased in x and z is built.

APPENDIX B

H-REFINEMENT TEMPLATES

Hierarchical element subdivision, widely know as h-refinement is one of the easiest ways to perform grid adaptation. Mesh enrichment is accomplished by subdividing the elements to form smaller children elements based on predetermined refinement patterns. Typically, subdivision rules are formulated to ensure valid element types are created during refinement. The isotropic refinement of different elements is shown in Figs.(B.1), (B.2), (B.3) and (B.4). The isotropic refinement of a tetrahedral, prism and hexahedral element results in 8 similar but smaller tetrahedrals, prisms and hexahedrals respectively. The isotropic refinement of a pyramid element results in 6 pyramids and 4 tetrahedral elements. In the present study, uniformly refined meshes are generated by the isotropic refinement of the elements in the mesh based on Figs.(B.1), (B.2), (B.3) and (B.4).

For adaptive mesh subdivision, only a limited number of refinement patterns are allowed. Figs.(B.5) and (B.6) show the permitted subdivision types for triangular and quadrilateral faces. The permitted subdivision types for tetrahedral elements is shown in Fig.(B.7). A tetrahedra may be divided into 2, 4, or 8 children. The subdivision of tetrahedra into 2 or 4 elements helps to stop additional mesh refinement. Fig.(B.8) shows the permitted subdivision types for prisms. A prism may be divided into 2 or 4 children. Isotropic refinement of a prism is not performed and instead, it is split into 4 new prisms. The refinement is allowed to propagate till the edge of the boundary layer, but, streamwise propagation is stopped by the subdivision of neighboring prisms into 2 new prisms. Though, the h-refinement module developed in the present study supports all the refinement patterns/templates described in [87], only subdivision types used in the present study are shown here.

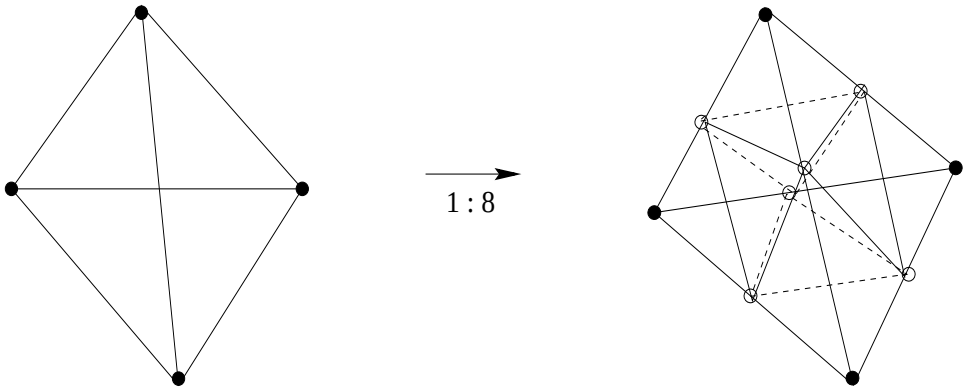


Figure B.1: Isotropic Refinement of Tetrahedra.

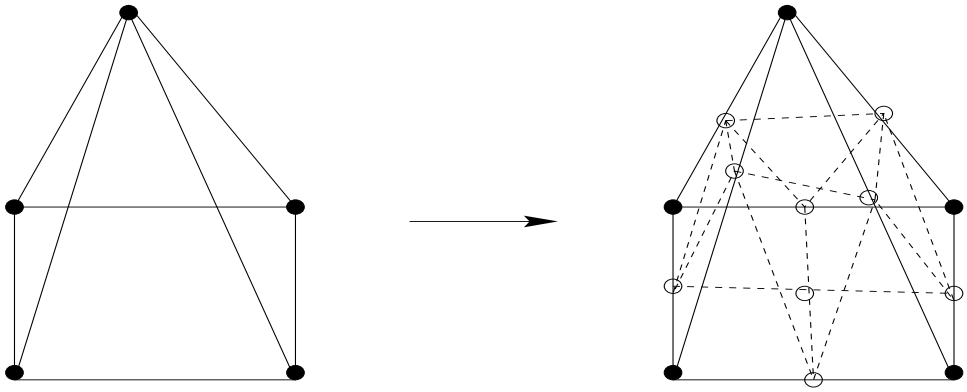


Figure B.2: Isotropic Refinement of Pyramid.

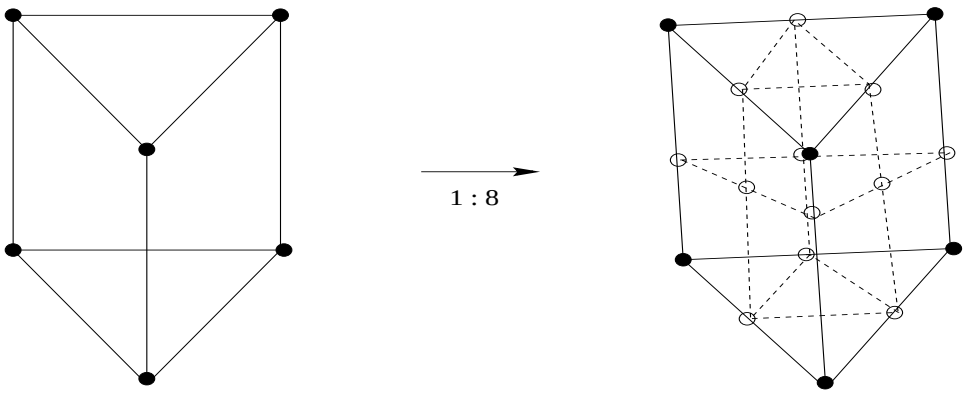


Figure B.3: Isotropic Refinement of Prism.

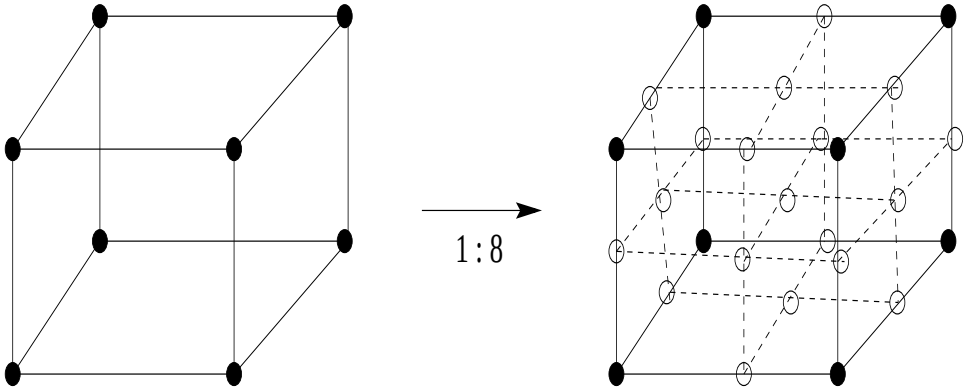


Figure B.4: Isotropic Refinement of Hexahedra.

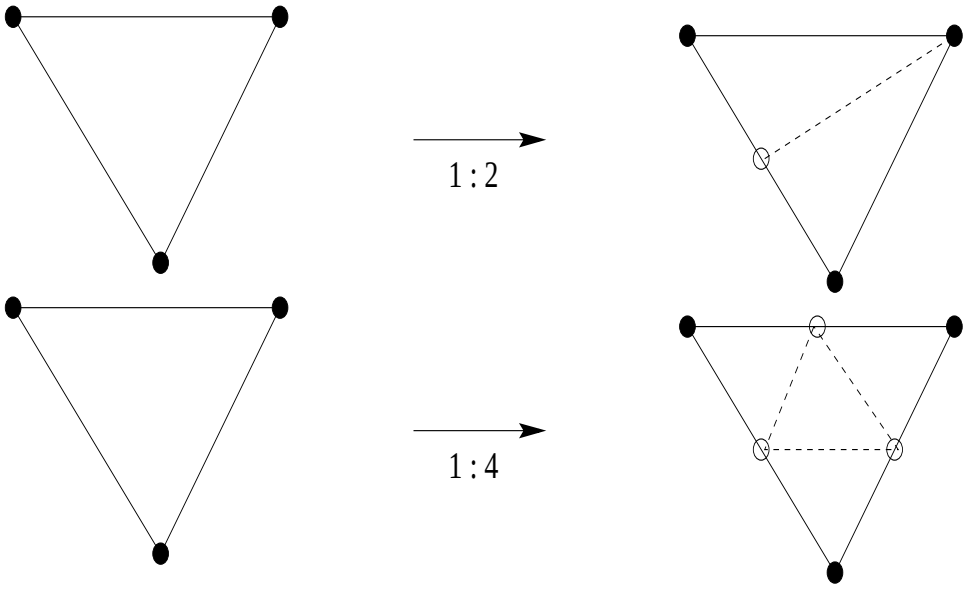


Figure B.5: Permitted subdivision types for Triangles in mesh adaptation.

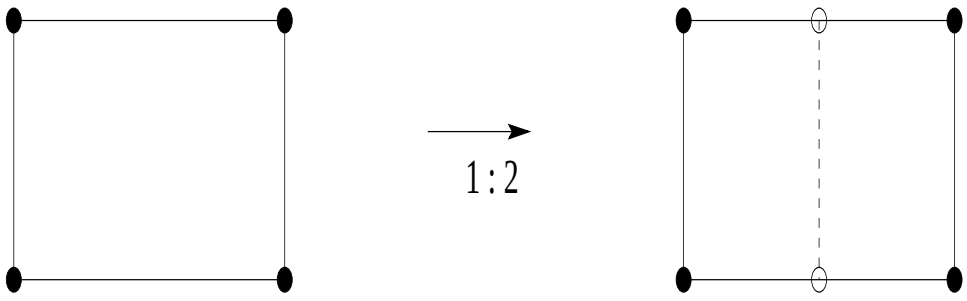


Figure B.6: Permitted subdivision types for Quadrilaterals in mesh adaptation.

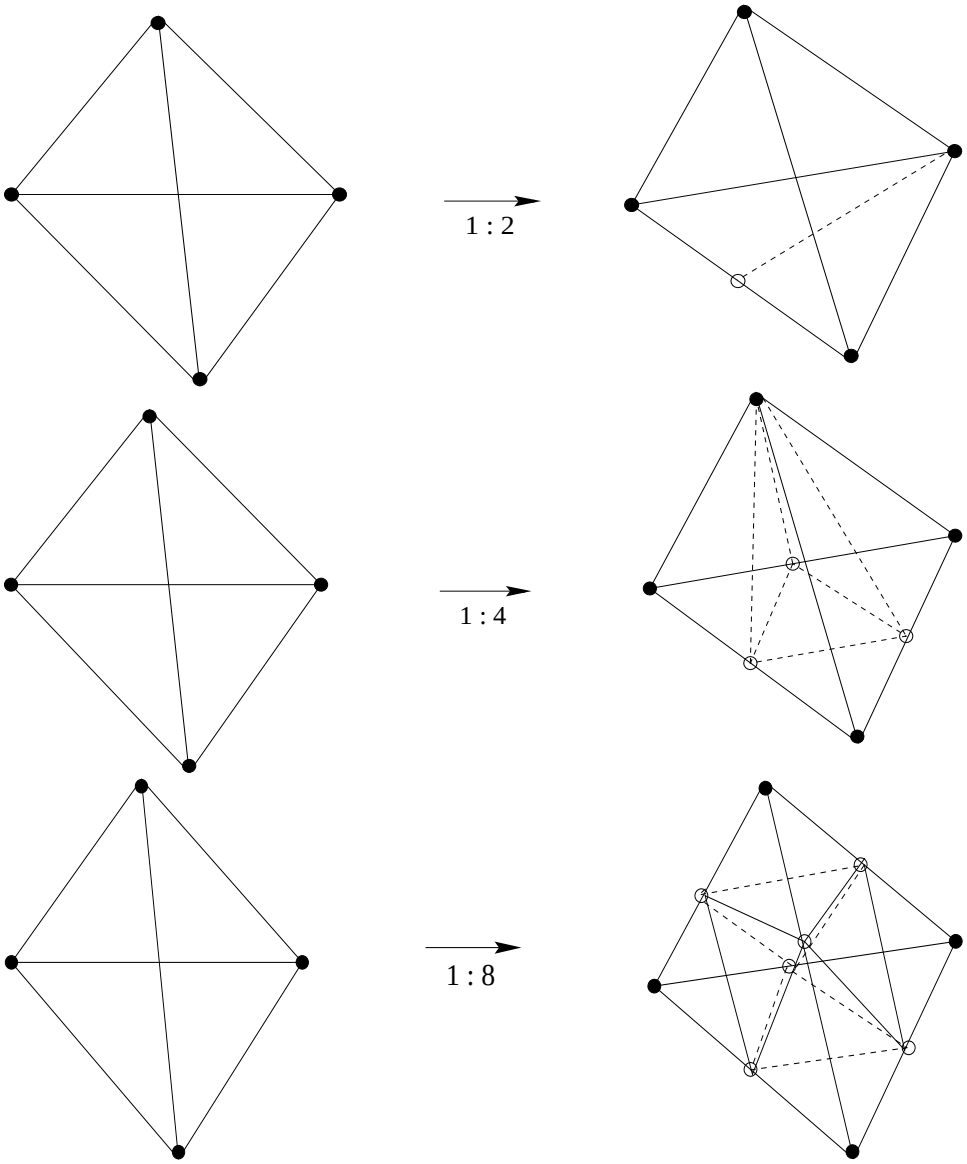


Figure B.7: Permitted subdivision types for Tetrahedral elements in mesh adaptation.

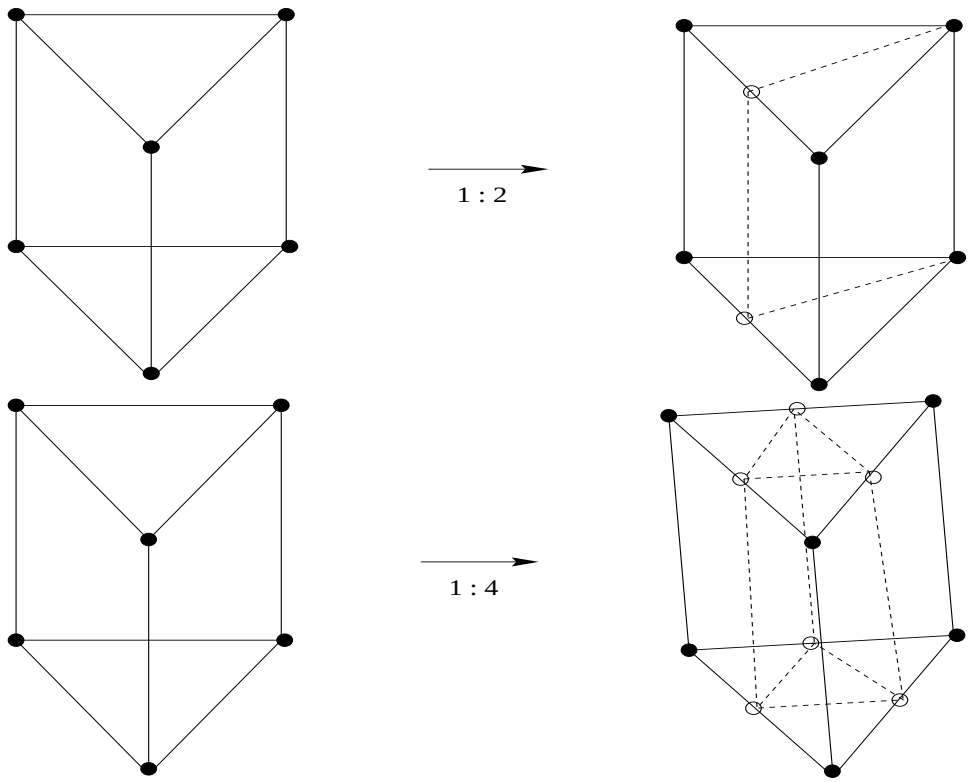


Figure B.8: Permitted subdivision types for Prism elements in mesh adaptation.