Development of Grid Movement Algorithms Suitable for Aerodynamic Optimization

by

Anh H. Truong

A thesis submitted in conformity with the requirements
for the degree of Masters of Applied Science
Graduate Department of Aerospace Engineering
University of Toronto

Abstract

Development of Grid Movement Algorithms Suitable for Aerodynamic Optimization

Anh H. Truong
Masters of Applied Science
Graduate Department of Aerospace Engineering
University of Toronto
2005

Two new grid movement algorithms have been developed and implemented in the aerodynamic shape optimization software *Optima2D*, *OptimaMB* and *Optima3D* for 2- and 3-dimensional analysis. The algorithms are based on the spring analogy and linear elasticity methods. The former uses a stiffening parameter to enhance element stiffness near the deforming boundary while the latter uses inhomogeneous material properties with a stiffening mechanism based on element distortion to control mesh deformation. Additionally, a fast element stiffness matrix procedure is adopted to increase the efficiency of its computation. The displacements of the internal nodes are solved for using the conjugate-gradient method with an incomplete lower-upper (ILU) preconditioner.

Large deformation tests are performed to compare the efficiency and robustness of the two methods. The algorithms are also used in several optimization cases, and the results are compared with those obtained using the existing algebraic grid perturbation method. For all cases, the linear elasticity method is the most robust, although it is also the most computationally expensive.

Acknowledgements

I would like to thank Professor D. W. Zingg for giving me this research opportunity and the financial assistance to complete my thesis. Thank you for providing me with invaluable advice and guidance throughout my degree. You inspire me with your sound approach to research, innovative ideas, and insightful comments and feedbacks. I am extremely grateful for having such an excellent supervisor.

Special thanks to the UTIAS community of professors, staff, and students, particularly the students in the CFD lab, for making my academic life so enjoyable. I was fortunate to have the benefits of their experience, wisdom, encouragement and friendship. I would like to thank Professor J. S. Hansen for his patience and expert advice on solid mechanics and testing of finite element codes. Thanks to Marian Nemec for giving me useful tips on debugging and understanding *Optima*. I would also like to thank Jon Driver for sharing his *Optima* knowledge, always with eagerness and enthusiasm. I wish to give grateful acknowledgement to Peterson Wong for many times going out of his way to help me deal with various difficulties, and to Iordan Iordanov, Scott Northrup and James McDonald for all the technical help in computers and programming. Thank you, Iordan, for giving me the extra miles.

My greatest gratitude is due to my family and friends for their love and support. Without you I would be lost. Thank you for helping me overcome the adversities and challenges. I am grateful to my mom for teaching me determination and perseverance. To her I dedicate this thesis.

Anh Truong
University of Toronto Institute for Aerospace Studies
September 29, 2005

Contents

Abstract iii

Acknowledgments v

Contents vii

List of Figures ix

List of Symbols xi

1 Introduction 1

1.1 Motivation 1

1.2 Review of Grid Movement Algorithms 2

1.3 Objectives 4

2 Spring Analogy Method 5

3 Linear Elasticity Model 9

3.1 Equations of Linear Elasticity 9

3.2 Finite Element Formulation 11

3.3 Distortion Measure Strategy 14

3.3.1 Quadrilateral element 14

3.3.2 Hexahedral element 15

3.4 Element Stiffness Matrix Evaluation 16

3.4.1 Fast Element Stiffness Computation 17

3.5 Stiffening Algorithm 18

4 Integration **19**
4.1 Aerodynamic Optimization Procedure 19
4.2 Algebraic Grid Perturbation Method 21

5 Results **23**
5.1 2-Dimensional Test Cases 23
5.1.1 Single-Element Airfoils 23
5.1.2 Multi-Element Airfoil 24
5.2 Optimization Cases 31
5.2.1 Maximization of Lift-to-Drag Ratio 31
5.2.2 Lift-Constrained Drag Minimization 33
5.3 3-Dimensional Test Cases 35

6 Conclusions and Recommendations **41**
6.1 Conclusions 41
6.2 Recommendations 42

References **43**

List of Figures

1.1 Elements in structured meshes 4

2.1 Negative area produced by the spring analogy method 7

3.1 Quadrilateral (left) and hexahedral (right) elements in local, isoparametric coordinate system 12

5.1 Far- and near-field view of NACA 0012 mesh after applying a rotation of 20^o pitch-up at 50% chord using the linear elasticity method (left column) and the spring analogy method (right column) 25

5.2 Close-up view of the deformed mesh of the NACA 0012 airfoil at 20^o pitch-up angle at 50% chord, showing the leading and trailing edge, using the linear elasticity method (left column) and the spring analogy method (right column) 26

5.3 The RAE 2822 airfoil undergoing large deformation 27

5.4 Deformed mesh of the RAE 2822 airfoil, showing the near-field region and the leading and trailing edge of the airfoil, using the linear elasticity method (left column) and the spring analogy method (right column) . . . 28

5.5 NLR 7301 airfoil 29

5.6 Close-up view of the leading and trailing edge of the NLR 7301 airfoil. Mesh obtained using the linear elasticity method (left) and the spring analogy method (right) 30

5.7 Original and optimized airfoil shapes obtained using different grid perturbation methods 31

5.8 Original and deformed mesh of the NACA 0012 airfoil, showing the major differences in the deformed airfoil shape at about 25% chord 34

5.9 Re-generated grid (green) and perturbed grid using the linear elasticity method (red) 36
5.10 Original H-topology mesh of the NACA 0012 airfoil 37
5.11 ONERA M6 wing - coarse mesh 38
5.12 ONERA M6 wing - fine mesh 39

List of Symbols

Alphanumeric Symbols

$\mathbf{B}$	strain-displacement matrix
$\mathbf{C}$	elasticity material matrix
$\mathbf{D}$	symmetric gradient operator
$\mathbf{f}$	external force vector
$\mathbf{F}_i$	vector of nodal forces at node i
$\mathbf{J}$	*Jacobian* transformation matrix
$\mathbf{K}_e$	element stiffness matrix
$\mathbf{u}$	displacement vector of an element
$\mathcal{G}$	gradient of the objective function
$\mathcal{J}$	objective function (scalar)
$\mathcal{N}$	element shape function
$\mathcal{R}$	vector of discrete residual equations
$\mathcal{X}$	design variable
A	area of a face of an element
C	constraint equations
c	chord length of airfoil
c_D	coefficient of drag

c_L coefficient of lift

d distance from boundary

E modulus of elasticity or Young's modulus

f_d wall distance function

K global stiffness matrix

k_{ij} spring stiffness of an edge with end nodes i and j

l_{ij} length of an edge with end nodes i and j

N grid node location

N_D number of design variables

N_I number of interior nodes in the mesh

n_i number of neighbours node i has

N_c number of constraint equations

p number of Gauss points

Q vector of conservative flow variables

R radius of the smallest circumscribed sphere

r radius of the largest inscribed sphere

S total surface area of an element

U total strain energy

u, v, w x, y, z components of $\mathbf{u}$

V element volume

x, y, z coordinates in the physical domain

Greek Symbols

α angle of attack

β spring length exponent

$\boldsymbol{\delta}$ vector of nodal displacements

γ power-law model exponent

ν Poisson's ratio

Ω feasible spatial domain in the design space

Φ measure of element quality

ϕ stiffness enhancement parameter

Ψ vector of adjoint variables

σ normal stress component

$\boldsymbol{\sigma}$ stress vector

τ shear stress component

$\boldsymbol{\varepsilon}$ strain vector

ζ, η, μ coordinates in the isoparametric domain

Superscripts

$\tilde{}$ normalized variable

T transpose

Subscripts

Δ triangular element

e generic element

H hexahedral element

T tetrahedral element

Chapter 1

Introduction

1.1 Motivation

Dynamic meshes are required in the simulation of flow problems involving moving boundaries. These problems occur in many engineering applications, including blood flow in arteries, induced vibrations of suspension bridges, skyscrapers, and offshore structures, oscillating airfoils, aeroelastic response of wings, and unsteady motion of rotor blades in forward flight [8, 1, 2]. More recently, with rapid advancements in processing speed and computing power and the extended use of computational fluid dynamics software as a design tool, dynamic meshes have also been used in the process of aerodynamic shape optimization. In either area of application, a new mesh has to be generated at each time step or iteration to fit the deformed surface, or the existing mesh has to be allowed to move with the computational domain. Allowing the existing mesh to evolve with the computational domain is clearly more efficient than generating a new mesh. In shape optimization, the boundary surface undergoes many small changes; it would be too time consuming to regenerate the mesh in response to these deformations. Re-meshing usually requires manual adjustments for complex geometries, and the projection of the solution from the old mesh to the new one, since the new mesh may not have the same number of nodes and connectivity [27]. A mesh perturbation approach, on the other hand, will inherently preserve the original grid connectivity, hence assuring the consistency of any mesh-induced errors in the flow solution (i.e. due to truncation error) between the initial and deformed grid, provided that a suitable mesh quality is maintained. It also assures continuity in the sensitivity derivatives. The robustness and efficiency of the mesh deformation tool is particularly important in gradient-based optimization because

any changes in the grid topology can have significant effects on these derivatives [17]. Most importantly, mesh movement algorithms have the potential to significantly reduce engineering cost by allowing the design process to be automated.

1.2 Review of Grid Movement Algorithms

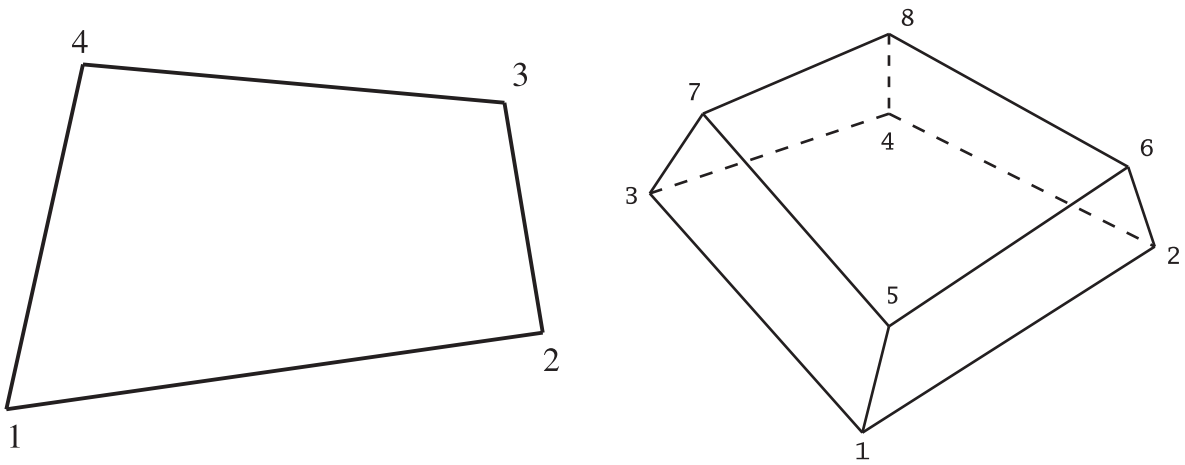
Currently, there are several categories of mesh moving techniques available. The most common technique is to model the mesh as a network of fictitous lineal springs whose stiffness is inversely proportional to the spring length. The displacements of the nodes attached to the ends of the springs are governed by the spring equilibrium condition. The spring analogy algorithm is known to be very efficient [3, 8, 17, 20], although it lacks robustness, failing to maintain adequate element quality for moderate to large geometric changes. This problem was addressed by Farhat *et al.* [8] with the introduction of additional torsional springs to monitor and control the amount of angular deformation an element undergoes. They successfully applied the modified method to mesh movements in two-dimensions. An extended three-dimensional version was developed and implemented by Farhat [7], Murayama *et al.* [19] and Burg [5]. Although this approach improved the robustness of the spring analogy method, its use is limited to unstructured meshes.

Recently, Samareh [27] extended the applicability of the spring analogy method to more generalized meshing schemes by using quaternions to propagate both the angular and translational changes of the boundary to the interior nodes. By propagating the rotational changes of the boundary into the field mesh, the method preserved the element quality and orthogonality at the deforming surface, which is desired for high-Reynolds-number viscous flow simulations. Quaternion algebra was used to represent rotations instead of the traditional Euler angles used by Morton *et al.* [18] because it leads to a simpler formulation and reduces computational cost. Additionally, the "gimbal lock" problem with the Euler angle method - the phenomenon of two rotational axes of an object pointing in the same direction - can be avoided. Samereh demonstrated that this method works well in two-dimensions for both structured and unstructured meshes. However, the applicability and feasibility of this method in three-dimensions have not been studied.

Another approach to mesh perturbation is to model the mesh as a continuum of elastic solid whose properties are defined by the modulus of elasticity, also known as Young's modulus, and Poisson's ratio. Nodal movements are governed by the equations

of linear elasticity, and mesh distortion can be controlled through the elements in the elastic matrix [28]. This method has been found to be much more robust than the spring analogy method, although relatively inefficient [20, 2, 17, 31]. Like the spring analogy technique, the use of the linear elasticity method usually requires some additional mesh stiffening mechanism to minimize the distortion of small elements, often by means of nonlinear material properties and/or nonlinear geometric deformation. Tezduyar *et al.* [29] introduced the technique of controlling the element deformation based on its size. This is achieved by assigning the modulus of elasticity to be inversely proportional to the cell volume. This way smaller elements will be more rigid and more resistant to deformation. Stein *et al.* [28] and Bar-Yoseph *et al.* [2] augmented this method to include a stiffening mechanism that stiffens the mesh with increasing mesh displacement for cases with even larger amplitudes of displacement and rotation.

The third and simplest approach to grid perturbation is the analytical method in which the coordinates of the nodes in the deformed mesh can be obtained by solving some analytical functions. This approach, however, works only on structured meshes. Liu *et al.* [16] solved Laplace's differential equation which maps the node space into the real space; Benson [4] used a finite-difference mesh relaxation stencil that is derived from inverting Laplace's equation, forming equipotential lines that correspond to the lines on a regular mesh. Giuliani [12] formulated a mesh movement algorithm that repositions a nodal point in relation with its neighbouring nodes such that the shape of the elements sharing that node remains regular. Allen [1] and Gadala *et al.* [11] used variations of the transfinite interpolation method to solve for the new grid node locations. One other alternative method is to utilize algebraic interpolation methods. Burgreen and Baysal [6] proposed a formulation that relocates the internal nodes in the direction normal to the deforming boundary through a distance that is proportional to its position with respect to this boundary. New nodal coordinates are generated by moving the original nodes on an indexing line of a structured mesh that projects from the deforming surface. This method was found to have a varying degrees of success depending on the blending function used to obtain the final coordinate positions. This formulation assumes that the normalized distribution is locally invariant, and the far-field boundary nodes are spacially fixed. Although some modifications were made by various researchers [25, 21, 22] to improve the orthogonality of the grid, the scheme breaks down for large amplitude displacements, and in cases where two intersecting surfaces (such as wing-body or nacelle-pylon intersections) are allowed to move simultaneously [26].



(a) quadrilateral (b) hexahedron

Figure 1.1: Elements in structured meshes

1.3 Objectives

The objective of this research is to develop a grid movement algorithm based on the spring analogy and the linear elasticity method for 2- and 3-dimensional structured meshes suitable for use in aerodynamic shape optimization. The perturbation procedure is differentiable, which means that the grid obtained has to be unique to a given set of design variables, which in turn means that the nodes have to be moved with respect to their original positions, not relative to those obtained from the previous optimization iteration. The algorithm is integrated with the existing two-dimensional aerodynamic optimization codes *Optima2D* and *OptimaMB*, developed by Nemec and Zingg [23, 21], for single and multi-block computational domains; and the three-dimensional structured multi-block aerodynamic optimization code *Optima3D* that is currently under development. Comparisons of the efficiency and robustness between the two algorithms are made for a variety of different meshes, consisting of quadrilateral and hexahedral elements, as shown in Fig. 1.1.

Chapter 2

Spring Analogy Method

There are two approaches to the spring analogy method: vertex springs or segment springs [32]. Vertex springs are always under tension, the magnitude of which depends on the length of the spring. Segment springs, on the other hand, have zero tension at some equilibrium length. The vertex model is mostly used in mesh smoothing algorithms, since it will result in a more uniform distribution of nodal points. The segment spring model allows better adjustment of the nodes to the deformed boundaries and is more commonly used in grid perturbation algorithms.

The segment spring method assumes that the edges of the mesh elements are springs whose stiffness is determined by the formula [32]:

$$k_{ij} = \phi \left[(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right]^{\beta} \tag{2.1}$$

where k_{ij} is the stiffness of an edge with end nodes i and j, $\beta = -\frac{1}{2}$ is the spring length exponent, and ϕ is the stiffness enhancement parameter and usually has a value of 1. Near the boundaries, where the elements are smaller and thus the spring stiffness needs to be higher to prevent excessive deformation, the values of ϕ is increased to 5 or more. Another variation is the use of the wall distance function $f_d \propto 1/d$ as a value for ϕ. This ensures high stiffness of the grid near the wall so that the deformation of the grid in this region is minimal [19].

When the boundary nodes move, the interior nodes also have to adjust to new positions. As the nodes are connected by the springs, there is a force generated on the node

when it experiences a displacement. The force, $\mathbf{F}$, at node i is

$$\mathbf{F}_i = \sum_{j=1}^{n_i} k_{ij} \left(\boldsymbol{\delta}_j - \boldsymbol{\delta}_i \right) \tag{2.2}$$

where n_i is the number of neighbouring nodes of i, and $\boldsymbol{\delta}_i$ is the displacement of node i from equilibrium. The **bold font** indicates that the quantity is a vector.

$$\begin{aligned} \boldsymbol{\delta} &= [\Delta x \;\; \Delta y \;\; \Delta z]^T \\ \mathbf{F} &= [F_x \;\; F_y \;\; F_z]^T \end{aligned} \tag{2.3}$$

Strictly speaking, these equations do not simulate the behaviour of a network of springs because there are no interactions between the x and y coordinates. The decoupling means that grid movement in one coordinate direction will not influence the displacement in the other, as would be in the case for a network of springs. Burg [5] proposed a formula for a two-dimensional unstructured mesh which relates the forces generated at each vertex to the displacements in both coordinate directions. However, he noted that this method often fails for complicated geometries and large deformations due to formation of invalid elements.

At equilibrium, the nodal forces, $\mathbf{F}_i$, in Eq. 2.2 are equal to zero. This means solving the system of equilibrium in each of the coordinate directions, for each nodal displacement, $\boldsymbol{\delta}_i$, in the domain until the displacement in the current time step is within a defined tolerance of the displacement in the previous time step. The linear system is given by:

$$\mathbf{F}_{x,y,z} = \mathbf{K}\boldsymbol{\delta}_{x,y,z} \tag{2.4}$$

In the system of equilibrium shown in Eq. 2.4, $\mathbf{K}$ is a $[N_I \times N_I]$ matrix and $\boldsymbol{\delta}_{x,y,z}$ and $\mathbf{F}_{x,y,z}$ are $[N_I \times 1]$ column vectors for each of the coordinate directions indicated by the subscripts x, y, z, where N_I is the number of interior nodes. The elements of $\mathbf{F}_{x,y,z}$ are zero unless they are connected to a boundary node with a prescribed displacement. $\mathbf{K}$ is a sparse, symmetric matrix and is stored in a compressed row format. The system of equilibrium equations, Eq. 2.4, is solved using the conjugate-gradient method with incomplete lower-upper (ILU) preconditioning. Convergence is reached when a tolerance below 10^{-7} is obtained.

The lineal model provides a stiffening measure to prevent the collision of two nodes,

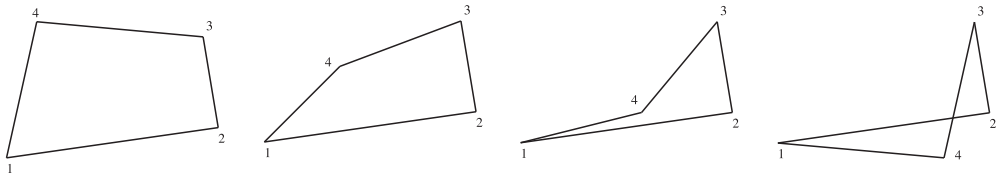


Figure 2.1: Negative area produced by the spring analogy method

since by design, the stiffness is inversely proportional to the length of the connecting edge, $k_{ij} \propto 1/l_{ij}$. This method works well for small amplitude displacements, but for a more complex problem, it often results in invalid elements due to the crossing of a vertex across an edge that faces it, [8, 2] as depicted in Fig. 2.1.

Chapter 3

Linear Elasticity Model

3.1 Equations of Linear Elasticity

The displacement of the interior nodes in a linearly elastic body in the spatial domain, Ω, bounded by Γ, are governed by the equilibrium equation:

$$\mathbf{D}^T\boldsymbol{\sigma} + \mathbf{f} = 0 \qquad \text{on} \qquad \Omega \tag{3.1}$$

subject to the displacement boundary condition, also known as the Dirichlet boundary condition

$$\mathbf{u} = \hat{\mathbf{u}} \qquad \text{on} \qquad \Gamma \tag{3.2}$$

where $\mathbf{D}$ is the symmetric gradient operator, $\boldsymbol{\sigma}$ is the stress vector consisting of normal, $\sigma_{x,y,z}$, and shear, $\tau_{xy,yz,xz}$, stresses; $\mathbf{f}$ is the vector of external forces; $\mathbf{u}$ is the vector of element displacement and $\hat{\mathbf{u}}$ is the vector of prescribed displacement on the boundary. In three-dimensions, the components in the equilibrium equation are expressed as:

$$\mathbf{D} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix}, \qquad \boldsymbol{\sigma} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{bmatrix}, \qquad \mathbf{f} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \tag{3.3}$$

The constitutive equations connect the stress and strain field in the elastic body.

These equations model the behaviour of the materials as continuum media. Generally they are partial differential equations or even integro-differential equations in space and time [30, 13]. However, for linear elasticity, the relation becomes algebraic, linear and homogenous, following Hooke's law; the stress $\boldsymbol{\sigma}$, is simply the product of the elasticity matrix, $\mathbf{C}$, and the strain vector, $\boldsymbol{\varepsilon}$:

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon} \tag{3.4}$$

Note that $\mathbf{C}$ is a symmetric $[6 \times 6]$ matrix with 36 components that are not all independent. For plane strain, isotropic materials, they can be expressed as a function of only two independent material constants, Young's modulus E and Poisson's ratio ν.

$$\mathbf{C} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5-\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5-\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5-\nu \end{bmatrix} \tag{3.5}$$

Young's modulus is also known as modulus of elasticity and elastic modulus. It is a measure of the stiffness of the material, whereas Poisson's ratio is a ratio of transverse compressive strain to longitudinal tensile strain, and is a measure of how much deformation in one principal direction affects the deformation in the other principal direction.

The strain, $\boldsymbol{\varepsilon}$, is related to the nodal displacement, $\mathbf{u}$, through the kinematic relation:

$$\boldsymbol{\varepsilon} = \mathbf{D}\mathbf{u} \tag{3.6}$$

where $\mathbf{u}$ is the displacement vector. The full matrix form of Eq. 3.6 is

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \\ \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3.7}$$

3.2 Finite Element Formulation

A typical finite element is defined by a prescribed number of boundary nodes, nb. The variation of displacement within the element, $\mathbf{u_e}$, can be approximated by a "Rayleigh-Ritz" or shape function, $\mathcal{N}$, such that

$$\mathbf{u_e} = \mathcal{N}_e \mathbf{u}_e = [N_1 N_2 ... N_{nb}] \cdot [u_1 u_2 ... u_{nb}]^T \tag{3.8}$$

The element shape function $\mathcal{N}_i$ is determined using Lagrange's interpolation formula [15]. The commonly used shape functions used for quadrilaterals and hexahedrals are the bilinear and trilinear functions, respectively, both with the following properties:

- They vary linearly for lines parallel to the coordinate axes, and non-linearly (due to the quadratic terms) along other lines.
- They are zero along element boundaries which do not contain the nodal point to which they belong.
- Two element shape functions belonging to the same nodal point are identical along the common element boundary. In other words, the approximated function is continuous across the element boundary.

The element compatibility requires the transformation of elements from the global Cartesian coordinate system to a local coordinate system. In two-dimensions, the region is described by the $\zeta\eta$-coordinate system bounded by the lines $\zeta = \pm 1$ and $\eta = \pm 1$, whereas in three dimensions, the region is a cube, with additional boundaries intersecting the third coordinate at $\mu = \pm 1$, as shown in Fig. 3.1.

In the natural coordinate system, the interpolation functions for the quadrilateral take the form:

$$\mathcal{N}_i(\zeta, \eta) = \frac{1}{4}(1 + \zeta\zeta_i)(1 + \eta\eta_i) \tag{3.9}$$

where the subscript i denodes the corner node number of the element (see Fig. 3.1); ζ_i and η_i are the local coordinates of the i^{th} corner node. Note that by convention, the nodes are numbered counter-clockwise to ensure consistency and positivity of element *Jacobian* and volume calculations. In the case of the hexahedron, the nodes are numbered counter-clockwise on a face when observed from the opposite face. The shape function of the hexahedron is:

$$\mathcal{N}_i = \frac{1}{8}(1 + \zeta\zeta_i)(1 + \eta\eta_i)(1 + \mu\mu_i) \tag{3.10}$$

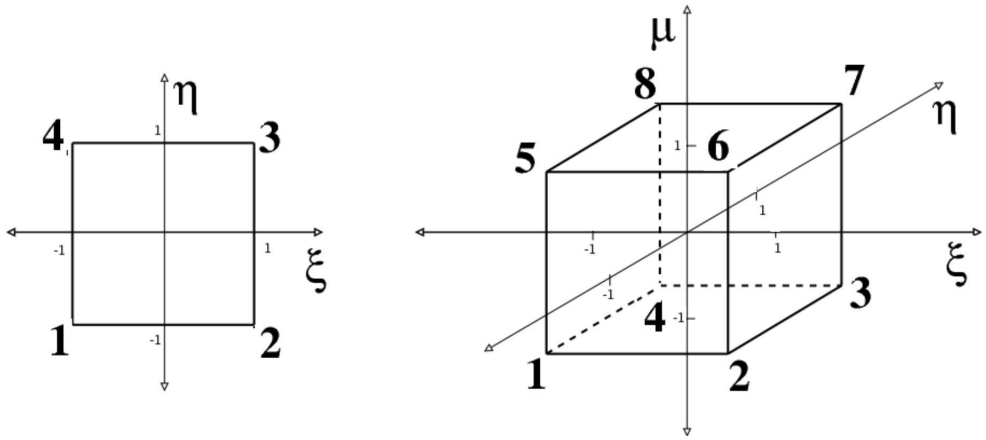


Figure 3.1: Quadrilateral (left) and hexahedral (right) elements in local, isoparametric coordinate system

where μ_i is the local coordinate of the i^{th} corner node in the third dimension. The elemental strain vector can be expressed in terms of the nodal displacements as:

$$\boldsymbol{\varepsilon}_e = \mathbf{B}_e \mathbf{u}_e \tag{3.11}$$

where $\mathbf{B}_e$ contains the appropriate derivatives of the shape functions multiplied by the *Jacobian* of the transformation. For example, the strain vector of a quadrilateral in two-dimensions is determined as follows:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{bmatrix}$$

$$\boldsymbol{\varepsilon} = \begin{bmatrix} J_{11}^{\star} & J_{12}^{\star} & 0 & 0 \\ 0 & 0 & J_{21}^{\star} & J_{22}^{\star} \\ J_{21}^{\star} & J_{22}^{\star} & J_{11}^{\star} & J_{12}^{\star} \end{bmatrix} \begin{bmatrix} \frac{\partial N_1}{\partial \zeta} & 0 & \frac{\partial N_2}{\partial \zeta} & 0 & \frac{\partial N_3}{\partial \zeta} & 0 & \frac{\partial N_4}{\partial \zeta} & 0 \\ \frac{\partial N_1}{\partial \zeta} & 0 & \frac{\partial N_2}{\partial \zeta} & 0 & \frac{\partial N_3}{\partial \zeta} & 0 & \frac{\partial N_4}{\partial \zeta} & 0 \\ 0 & \frac{\partial N_1}{\partial \eta} & 0 & \frac{\partial N_2}{\partial \eta} & 0 & \frac{\partial N_3}{\partial \eta} & 0 & \frac{\partial N_4}{\partial \eta} \\ 0 & \frac{\partial N_1}{\partial \eta} & 0 & \frac{\partial N_2}{\partial \eta} & 0 & \frac{\partial N_3}{\partial \eta} & 0 & \frac{\partial N_4}{\partial \eta} \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{bmatrix}$$

Here, the $J^{\star}$'s are the components of the inverse of the *Jacobian* matrix with subscripts

indicating the matrix indices. The *Jacobian* for a quadrilateral is a $[2\times2]$ matrix obtained by multiplying the shape function by the nodal coordinates of the element.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial N_1}{\partial \zeta} & \frac{\partial N_2}{\partial \zeta} & \frac{\partial N_3}{\partial \zeta} & \frac{\partial N_4}{\partial \zeta} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}$$

The stresses in the element, $\boldsymbol{\sigma}_e$, that correspond to the strains $\boldsymbol{\varepsilon}_e$ is:

$$\boldsymbol{\sigma}_e = \mathbf{C}_e \boldsymbol{\varepsilon}_e \tag{3.12}$$

where $\mathbf{C}_e$ is the appropriate (2-d plane stress, 2-d plane strain, or 3-d) constitutive elasticity matrix (Eq. 3.5, for example).

The stiffness matrix for each element can be determined using the strain-energy method - itself derived from the equilibrium equation through application of the calculus of variations - which states that:

$$\text{Work done} = \text{stored strain energy} = \frac{1}{2}\int_V \boldsymbol{\varepsilon}_e^T \boldsymbol{\sigma}_e \, dV = \frac{1}{2}\mathbf{u}_e^T \mathbf{K}_e \mathbf{u}_e \tag{3.13}$$

where

$$\mathbf{K}_e = \int_V \mathbf{B}_e^T \mathbf{C}_e \mathbf{B}_e \, dV \tag{3.14}$$

is the element stiffness matrix, and $dV = dx\,dy\,dz$ is the differential volume. In natural coordinates $dV = |\mathbf{J}|\, d\zeta\, d\eta\, d\mu$, where $|\mathbf{J}|$ is the determinant of the *Jacobian* matrix.

The total strain energy, $\mathbf{U}$, stored in all the elements is the sum of all the elemental strain energies:

$$\mathbf{U} = \sum_e \frac{1}{2}\mathbf{u}_e^T \mathbf{K}_e \mathbf{u}_e = \frac{1}{2}\mathbf{u}^T \mathbf{K} \mathbf{u} \tag{3.15}$$

where $\mathbf{K} = \sum_e \mathbf{K}_e$ is the global stiffness matrix, and $\mathbf{u} = [u_1\, v_1\, w_1\, u_2\, v_2 \ldots \ldots w_{NI}]^T$ is the global nodal displacement vector that contains the total degrees of freedom of the system. The total potential energy functional, Π, can be written as the difference between the total strain energy, which is the only stored energy in an elastic body, and the total external work $\mathbf{u}^T\mathbf{F}$:

$$\Pi = \frac{1}{2}\mathbf{u}^T \mathbf{K} \mathbf{u} - \mathbf{u}^T \mathbf{F} \tag{3.16}$$

where $\mathbf{F} = [F_{x_1}\, F_{y_1}\, F_{z_1}\, F_{x_2}\, F_{y_2} \ldots .. F_{z_{NI}}]^T$

Minimizing the potential energy functional with respect to the nodal displacement components in $\mathbf{u}$ yields,

$$\mathbf{Ku} = \mathbf{F} \tag{3.17}$$

Eq. 3.17 defines the pseudo-elastostatic problem of mesh deformation. Since the boundary displacements are given, the nodal force vector $\mathbf{F}$ is mostly zero except at the deformed boundaries, where it is equal to the stiffness of the node times its displacement. In this formulation, elements with high stiffness will be displaced as a solid body. Thus, a variable stiffness distribution is used to control the distortion of the mesh. Elements are assigned material properties which are inversely proportional to their respective volumes. In addition, a stiffening mechanism that varies with the distortion measures is applied to enhance control of the geometric distortions of the elements, including aspect ratio, taper and skew.

3.3 Distortion Measure Strategy

The element distortion measure for volumetric, angular and curvature distortions at the ith step in the incremental procedure in Section 3.5, $\tilde{\Phi}_e^i$, is defined as the ratio of the element shape quality to its reference value:

$$\tilde{\Phi}_e^i = \frac{\hat{\Phi}_e^i}{\hat{\Phi}_e^0} \tag{3.18}$$

$\hat{\Phi}_e^i$ is calculated differently depending on the type of the element. Details are given in the following sub-sections. These formulations were suggested by Bar-Yoseph *et al.* [2], and have been demonstrated to work well for large deformation problems.

3.3.1 Quadrilateral element

For a quadrilateral element, the element distortion measure is defined in terms of the norm of the element aspect ratios of its sub-triangular elements:

$$\hat{\Phi}_Q^i = \left[\sum_{i=1}^{4} \left(\hat{\Phi}_\Delta^i \right)^P \right]^{1/P} \tag{3.19}$$

Δ_i is the sub triangular element inside the quadrilateral shown in Fig. 3.1. $\Delta_1 = \Delta_{1,2,4}$, $\Delta_2 = \Delta_{1,2,3}$, $\Delta_3 = \Delta_{2,3,4}$, $\Delta_4 = \Delta_{1,3,4}$, and P indicates the degree of the norm desired. For all cases, a degree of 2 was used. The element aspect ratio of a triangular element is defined as:

$$\hat{\Phi}^i_\Delta = \frac{R_\Delta}{r_\Delta} \tag{3.20}$$

where R_Δ is the radius of the smallest circumscribed circle, and r_Δ is the radius of the largest inscribed circle:

$$\frac{R_\Delta}{r_\Delta} = \frac{s_\Delta l_1 l_2 l_3}{4A^2_\Delta} \tag{3.21}$$

where $A_\Delta = \sqrt{s_\Delta \left(s_\Delta - l_1\right)\left(s_\Delta - l_2\right)\left(s_\Delta - l_3\right)}$ is the area of the triangle Δ, $s_\Delta = 0.5\left(l_1 + l_2 + l_3\right)$ is the semi-perimeter, and $l_{1,2,3}$ are the lengths of the sides of the triangle.

3.3.2 Hexahedral element

A similar element subdivision is done on the hexahedra. The hexahedron is subdivided into eight non-overlapping tetrahedral sub-elements, T_i, and $i = 1, 2, \ldots 8$, where $T_1 = T_{1,3,6,2}$, $T_2 = T_{8,3,7,6}$, $T_3 = T_{4,7,8,5}$, $T_4 = T_{1,8,5,6}$, $T_5 = T_{1,4,5,2}$, $T_6 = T_{6,5,7,2}$, $T_7 = T_{4,3,7,2}$, $T_8 = T_{1,4,8,3}$ (Fig. 3.1). Note that the nodal numbering of the tetrahedra follows the same counter-clockwise convention discussed earlier. In this case, the numbering on a face of a tetrahedron is counter-clockwise looking from the last node. The element distortion measure is defined as the norm, to a degree P, of the element distortion measure of its sub-tetrahedra

$$\hat{\Phi}^i_H = \left[\sum_{i=1}^{8} \left(\hat{\Phi}_{T_i}\right)^P\right]^{1/P} \tag{3.22}$$

The element distortion measure of a tetrahedron is

$$\hat{\Phi}^i_T = \frac{R_T}{r_T} \approx \frac{l_{max}}{r_T} \tag{3.23}$$

where l_{max} is the maximum edge length, R_T is the radius of the smallest circumscribed sphere and r_T is the radius of the largest inscribed sphere, given by

$$r_T = \frac{3V}{S} \tag{3.24}$$

where V is the volume, and S is the surface area of the tetrahedron:

$$S = \sum_{i=1}^{4} \sqrt{s_i \left(s_i - l_{i,1}\right) \left(s_i - l_{i,2}\right) \left(s_i - l_{i,3}\right)} \tag{3.25}$$

and

$$s_i = 0.5 \left(l_{i_1} + l_{i_2} + l_{i_3}\right) \tag{3.26}$$

where $l_{i_{1,2,3}}$ are the lengths of the three edges on the ith face.

3.4 Element Stiffness Matrix Evaluation

The integration in Eq. 3.14 is done numerically using Gauss quadrature. According to the Weierstrass theorem on polynomial approximation, any function $f(x)dx$ can be approximated within a given interval (a, b) by a polynomial of sufficiently high degree [15]. Thus, the value of the definite integral $\int_a^b f(x)dx$ can be approximated by appropriate weighted sums of particular values of the function f evaluated at specified points x_i

$$\int_a^b f(x)dx = \sum_{i=1}^{p} w_i f(x_i) \tag{3.27}$$

where p is the number of Gauss or sampling points and w_i are the weights. Since a polynomial of degree m is uniquely defined by $m + 1$ constants, if $m = 2p - 1$, the constants x_i and w_i can be determined to a desired precision so that the integration is exact. This principle is applied to the integration of Eq. 3.14. Assuming that the stress-strain matrix $\mathbf{C}$ is constant over the element,

$$\mathbf{K}_e = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \mathbf{B}^T\mathbf{C}\mathbf{B}|\mathbf{J}|d\zeta\, d\eta\, d\mu = \sum_{i=1}^{p_\eta}\sum_{j=1}^{p_\zeta}\sum_{k=1}^{p_\mu} w_i w_j w_k \mathbf{C}_{ijk}^T \mathbf{E}\mathbf{B}_{ijk}|\mathbf{J}|_{ijk} \tag{3.28}$$

where $p_\eta = p_\zeta = p_\mu = 2$ are the number of Gauss or sampling points in the ζ, η and μ directions, respectively, and $w_i = 1.00$ are their respective weights, while $\mathbf{B}_{ijk}$ and $|\mathbf{J}|_{ijk}$ are the strain-displacement matrix and *Jacobian* determinant evaluated at the Gauss point (ζ_i, η_i, μ_i). The values of p_i and w_i are determined using Legendre polynomials [15, 14, 33, 34]. The locations of the Gauss points are listed in Table 3.1.

Table 3.1: Locations of Gauss points for an 8-node hexahedron

Node	ζ_i	η_i	μ_i
1	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$
2	$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$
3	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$
4	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$
5	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$
6	$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$
7	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$
8	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$

3.4.1 Fast Element Stiffness Computation

Consider the matrix multiplication required in Eq. 3.28. Both the strain-displacement matrix $\mathbf{B}_{ijk}$ and the constitutive matrix $\mathbf{C}$ are sparse matrices whose structures can be exploited such that greater computational efficiency is achieved [15]. Note that $\mathbf{C}$ and $\mathbf{B}_{ijk}$ have the form:

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & 0 & 0 & 0 \\ c_{21} & c_{22} & c_{23} & 0 & 0 & 0 \\ c_{31} & c_{32} & c_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{66} \end{bmatrix} \qquad \mathbf{B}_{ijk} = \begin{bmatrix} b_{11} & 0 & 0 \\ 0 & b_{22} & 0 \\ 0 & 0 & b_{33} \\ b_{41} & b_{42} & 0 \\ 0 & b_{52} & b_{53} \\ b_{61} & 0 & b_{63} \end{bmatrix}$$

$$\mathbf{CB}_{ijk} = \begin{bmatrix} c_{11}b_{11} & c_{12}b_{22} & c_{13}b_{33} \\ c_{21}b_{11} & c_{22}b_{22} & c_{23}b_{33} \\ c_{31}b_{11} & c_{32}b_{22} & c_{33}b_{33} \\ c_{44}b_{41} & c_{44}b_{42} & 0 \\ 0 & c_{55}b_{52} & c_{55}b_{53} \\ c_{66}b_{61} & 0 & c_{66}b_{66} \end{bmatrix} \tag{3.29}$$

A straight forward implementation of matrix multiplication would require 108 (18×6) multiplication and 90 (15×6) addition operations to get the resultant matrix. However, by considering the zeros in the $\mathbf{C}$ and $\mathbf{B}_{ijk}$ matrices, the product can be calculated with

only 15 multiplications. This procedure, which involves direct assembly of the elements, leads to a total saving of over 86% in computation. It is thus adopted in the computation of the product term $\mathbf{B}_{ijk}^T C \mathbf{B}_{ijk}$ in the element stiffness matrix.

3.5 Stiffening Algorithm

For problems involving large displacements, it is necessary to stiffen the elements that have undergone some distortions, making them more rigid so that the deformation can propagate further into the far-field mesh, the more elastic part of the domain. To accomplish this, the displacement is divided into smaller increments. The nodal positions are updated at each increment and the most current size of the element is used to calculate the modulus of elasticity in the material matrix. The Young's modulus of an element is calculated as:

$$E^i = V^i \left(\tilde{\Phi}^i \right)^\gamma \tag{3.30}$$

where V^i is the element volume (in two-dimensions it would be element area) at the current incremental step, i, $\tilde{\Phi}^i$ is its normalized measure of quality, determined from Eq. 3.18, and γ is the stiffening power. γ is a positive factor that can be set high or low depending on the element type and distortion mode. This formulation allows the modulus of elasticity to vary as a power-law function of the normalized measure of element quality. It has two advantages:

1. Only relative distortion will cause the material to stiffen.

2. The element quality can be monitored during the incremental procedure. If an element approaches a tangled state, the stiffness will increase exponentially, thus preventing further deformation.

Chapter 4

Integration

The final step is to integrate the mesh movement algorithms into the aerodynamic shape optimization software *Optima2D*, *OptimaMB* and *Optima3D* for two- and three-dimensional problems. Integration with the two-dimensional software involves interfacing with their respective *regrid* subroutines. This subroutine generates the deformed airfoil coordinates, which are used by the grid perturbation algorithms to perturb the mesh. The new set of nodal coordinates are then passed back to the *regrid* subroutine. Perturbation is done with respect to the initial mesh to ensure uniqueness of the mesh sensitivity derivatives in the gradient calculations. A similar process is performed in three-dimensions. The difference is that the deformed surface is supplied by the geometry manipulation software developed by Fudge [9, 10] and new coordinates of the interior volume nodes are passed back to *Optima3D* for flow solution and gradient calculations. Although the principle is the same as in two-dimensions, the complexities in connectivities and orientation of blocks and boundaries and in matrix calculations are greatly increased in three-dimensions.

4.1 Aerodynamic Optimization Procedure

The aerodynamic shape optimization problem begins with a geometry whose shape can be controlled by a set of parameters called design variables. These geometric design variables are usually a subset of B-spline control points used to define the airfoil shape [21] or the B-spline control net defining the wing-body configuration of an airplane [9]. Additional design variables such as angle of attack and angle of twist can be specified to fully describe the design space. It is important to include a sufficient number of design

variables from which an optimal design can be obtained. The goal of the optimization procedure is to obtain a final set of design variables such that the objectives (e.g. lift-constrained drag minimization and maximization of lift-to-drag ratio in one or more flow conditions) are satisfied. It is formulated as a minimization problem of an objective function that is dependent on the design and flow variables $\mathcal{X}$ and Q, respectively:

$$\min_{\mathcal{X}} \ \mathcal{J}(\mathcal{X}, Q) \tag{4.1}$$

In reality, it is usually necessary to place some physical constraints into the design due to manufacturing limitations. For example, an airfoil must have a minimum thickness (for fuel storage considerations), or a wing a maximum span (for structural integrity considerations). These can be applied to Eq. 4.1 in terms of constraint equations:

$$C_j(\mathcal{X}, Q) \leq 0 \qquad j = 1, \ldots, N_\mathrm{c} \tag{4.2}$$

where N_c denotes the number of constraint equations. The constraint equations are also used to prevent the occurrence of invalid geometry, such as the cross-over of airfoil surfaces [21]. In *Optima2D* and *OptimaMB*, the flow variables, Q, are obtained by solving the compressible thin-layer Navier-Stokes equations with the one-equation Spalart-Allmaras turbulence model, while *Optima3D* uses the inviscid Euler equations [24].

In the gradient-based approach, the minimum of the objective function in Eq. 4.1 is determined by calculating the derivative of the objective function. In the *Optima* codes, this is done using the discrete adjoint method:

$$\mathcal{G} = \frac{d\mathcal{J}}{d\mathcal{X}} = \frac{\partial \mathcal{J}}{\partial \mathcal{X}} - \Psi^T \frac{\partial \mathcal{R}}{\partial \mathcal{X}} \tag{4.3}$$

where Ψ represents the adjoint variables which can be solved by the adjoint equation, Eq. 4.4, using the generalized minimal residual (GMRES):

$$\frac{\partial \mathcal{R}^T}{\partial Q} \Psi = \frac{\partial \mathcal{J}^T}{\partial Q} \tag{4.4}$$

and $\mathcal{R}$ is the residual of the discretized flow equation. The objective function sensitivity, $\frac{\partial \mathcal{R}}{\partial \mathcal{X}}$, and the residual sensitivity, $\frac{\partial \mathcal{J}}{\partial \mathcal{X}}$, are evaluated using centered differences. The adjoint formulation allows the calculation of the gradient of the objective function independent of the number of design variables. The evaluation of grid sensitivities is included in the

residual sensitivity calculation since

$$\frac{\partial \mathcal{R}}{\partial \mathcal{X}} = \frac{\partial \mathcal{R}}{\partial N}\frac{\partial N}{\partial \mathcal{X}} \tag{4.5}$$

where $\frac{\partial N}{\partial \mathcal{X}}$ is the grid sensitivity derivative, the sensitivity of the grid node location, N, with respect to the design variable, $\mathcal{X}$. To ensure differentiability of the grid node locations with respect to the design parameters, the deformed grid is always obtained from the original grid. Conforming the grid to one set of design variables and back to the original set will generate the original grid, and uniqueness of the function is preserved.

4.2 Algebraic Grid Perturbation Method

Both *Optima2D* and *OptimaMB* [21] use an algebraic grid perturbation method to adjust the interior grid nodes from their original positions to conform to the new configuration. This algorithm preserves the location of the outer boundary, while the positions of the interior nodes are determined by

$$y_i^{new} = y_i^{old} + \Delta y \left(1 - \tilde{d}_i\right) \qquad i = 1, \ldots, i_{max-1} \tag{4.6}$$

where Δy represents the airfoil shape change, $\tilde{d}_i$ is the normalized arclength distance with respect to the grid-line distance from the airfoil to the far-field boundary, which is defined as:

$$\begin{gathered} \tilde{d}_1 = 0 \\ \tilde{d}_i = \textstyle\sum_{i=2}^{i} d_{ij} / \sum_{i=2}^{i_{max}} d_{ij} \qquad i = 2, \ldots, i_{max-1} \end{gathered} \tag{4.7}$$

where d_{ij} is the length of the grid line from node $j = i - 1$ to node i. An analogous formulation holds for the x-coordinate direction. This algorithm is used for C-topology grids in *Optima2D*, while a slightly modified version is used for the H-topology grids in *OptimaMB*. The modification is necessary to improve orthogonality of the grid lines near the deforming boundary as well as block interfaces. New grid node locations are determined by

$$y_i^{new} = y_i^{old} + \frac{\Delta y}{2}\left[1 + \cos\left(\pi \tilde{d}_i\right)\right] \qquad i = 1, \ldots, i_{max-1} \tag{4.8}$$

The improved algebraic method used by Nemec [21] (Eq. 4.8) operates only on blocks that have direct contact with the deforming airfoil boundary. In this respect, the deformation

of the grid remains local. Thus, the grid quality can be significantly compromised for large displacements. This problem is addressed in the new grid movement algorithms in that all deformations are allowed to propagate across block interfaces into the far-field region.

Chapter 5

Results

5.1 2-Dimensional Test Cases

Several large deflection problems are presented to compare the quality of the meshes generated by the two new mesh perturbation algorithms.

5.1.1 Single-Element Airfoils

In the first case, a NACA 0012 airfoil is given a rotation of 20° about the 50% chord location. The mesh consists of 9045 nodes, as shown in Fig. 5.1, with a closer view of the near-field region and leading and trailing edge of the airfoil in Fig. 5.2. An invalid mesh is obtained using the spring analogy method. At the tip of the airfoil, where there is a high degree of curvature, the scheme cannot maintain adequate element quality, resulting in grid cross-over. With the linear elasticity method, a stiffening power of 15 is used to minimize the distortion of the elements at the trailing edge, and the maximum displacement is obtained through 50 small, equal increments, with total a run-time of 410 seconds. In this case, a Poisson's ratio of -0.8 is used to prevent the crunching up of the elements around the trailing edge due to the large displacement. The element quality in the near-field region is maintained. These cells moved as a solid body, and the displacement is propagated into the far-field region. The maximum cell aspect ratio in the grid and minimum cell area improved slightly from 25659.9 and 1.46829×10^{-9} to 25638.2 and 1.47025×10^{-9}, respectively.

The second case is a deformation of an RAE 2822 airfoil whose mesh consists of $25,875$ nodes, shown in Fig. 5.3. The airfoil is given a shape transformation and a rigid-

body translation over a distance of 20% chord in the x-direction and 10% chord in the y-direction. The results are shown in Fig. 5.4. Again, the spring analogy yielded an invalid mesh, with some grid folding occurring at the trailing edge region. A check on the Jacobian of the elements also revealed some negative values due to the presence of the inverted elements. The mesh resulting from the linear elasticity method, however, maintained a high degree of element quality. The maximum cell aspect ratio has only increased slightly from 25599.2 to 25857.2, while the minimum area has decreased from 1.41169×10^{-9} to 1.40412×10^{-9}. A series of 30 increments is applied to the geometry shown in Fig. 5.3 with a total of 363 seconds of run-time. In this case, Poisson's ratio is set to -0.2, and a stiffening power of 15 is used.

5.1.2 Multi-Element Airfoil

The third case involves repositioning of the flap of a multi-element airfoil, the NLR 7301, with a mesh consisting of $30,600$ nodes, as shown in Fig. 5.5. The mesh obtained using the spring analogy method showed acceptable grid quality (Fig. 5.6). The minimum cell area decreased only slightly from 3.94549×10^{-9} to 3.93192×10^{-9}, while the maximum element aspect ratio increased from 29100.5 to 29108.1. Using only 3 increments, the final displacement was obtained in 97 seconds. With the linear elasticity method, the final position was achieved after 30 increments with a total run-time of 6362 seconds. Although the mesh quality obtained using this method is slightly better than that from the spring analogy method, with minimum cell area and maximum element aspect ratio of 3.94488×10^{-9} and 29100.6, respectively, the time taken is significantly greater. The resultant mesh using this method has grid lines in the stream-wise direction, just above the leading edge of the flap, with relatively high degree of curvature. This does not compromise the quality of the elements because it occurs over the region where the mesh density in the radial direction is also high. In fact, the overall result is a smoother mesh compare to the one obtained using the spring analogy method. The Poisson's ratio and stiffening power are 0 and 8, respectively.

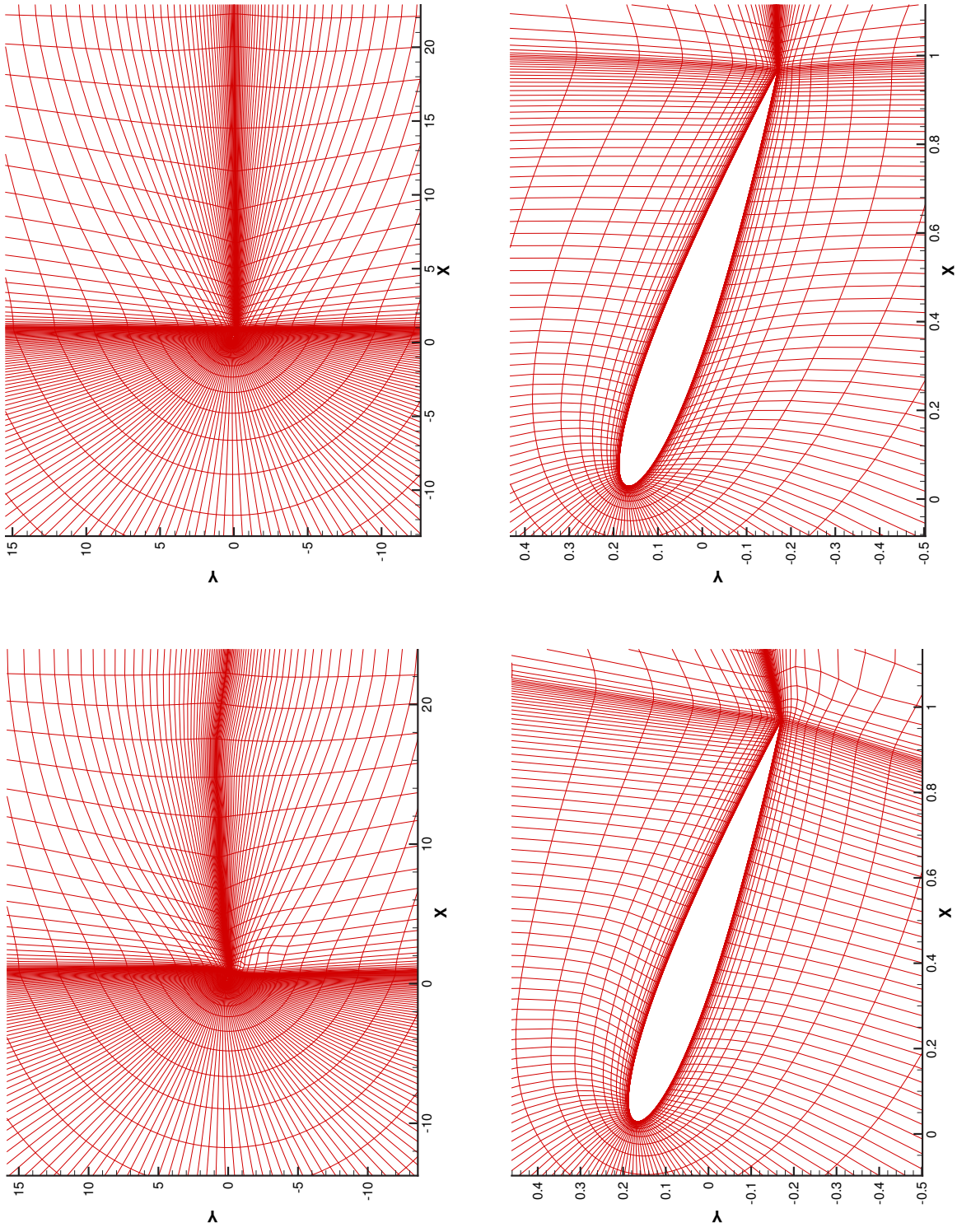


Figure 5.1: Far- and near-field view of NACA 0012 mesh after applying a rotation of 20° pitch-up at 50% chord using the linear elasticity method (left column) and the spring analogy method (right column)

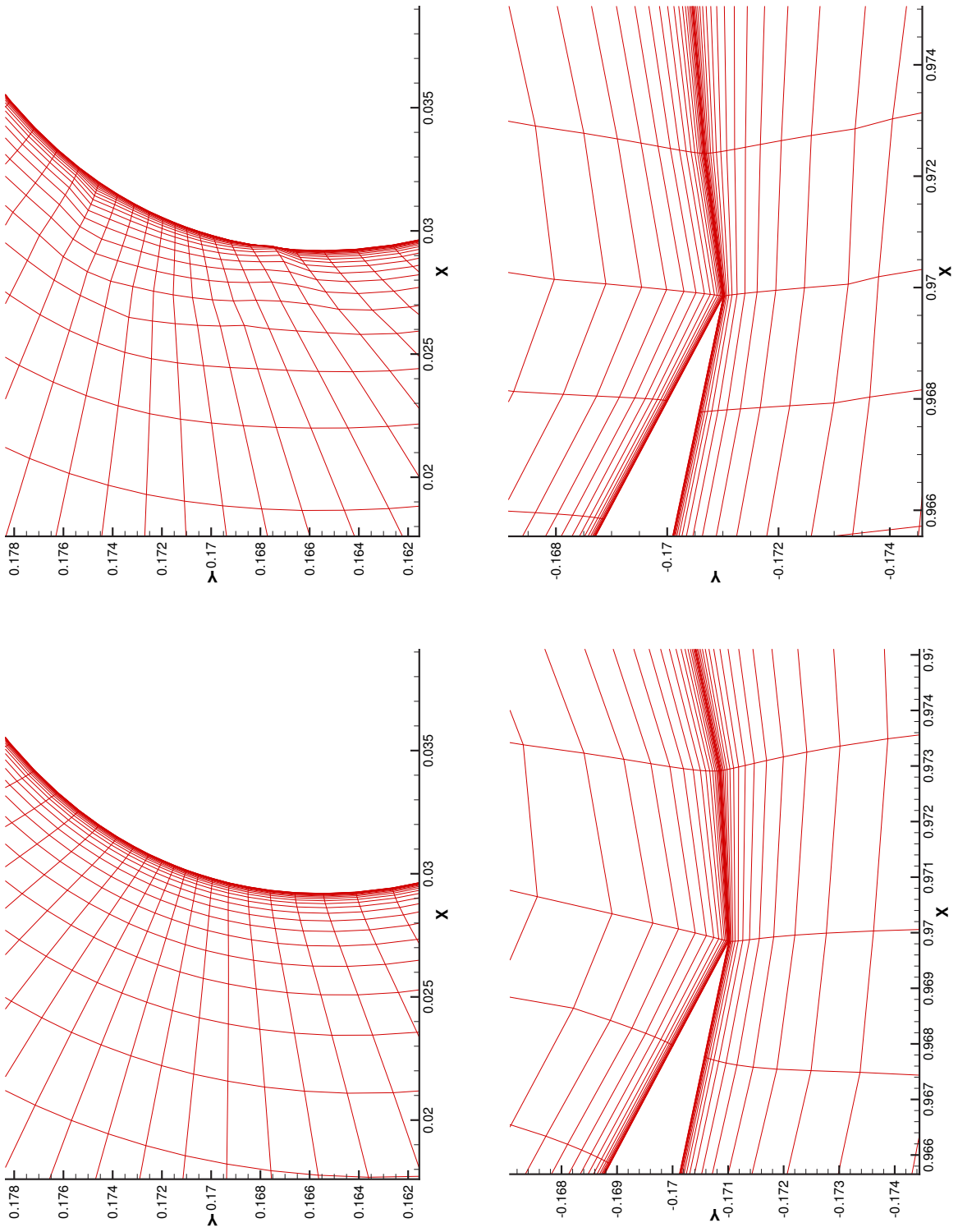
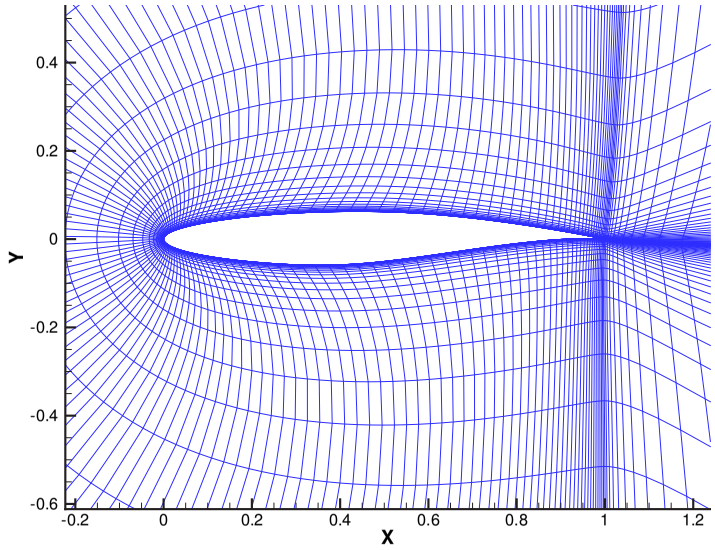
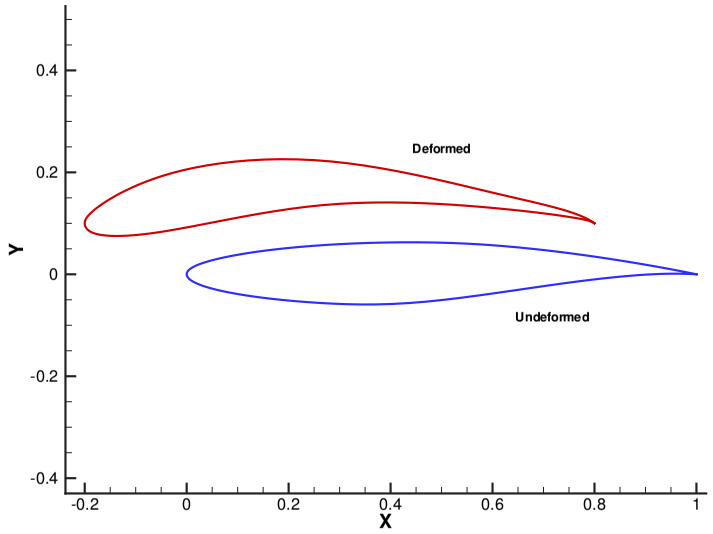


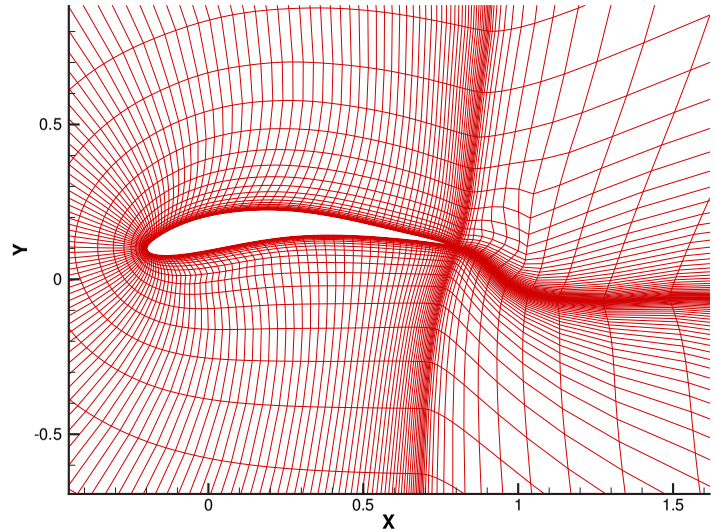
Figure 5.2: Close-up view of the deformed mesh of the NACA 0012 airfoil at 20° pitch-up angle at 50% chord, showing the leading and trailing edge, using the linear elasticity method (left column) and the spring analogy method (right column)



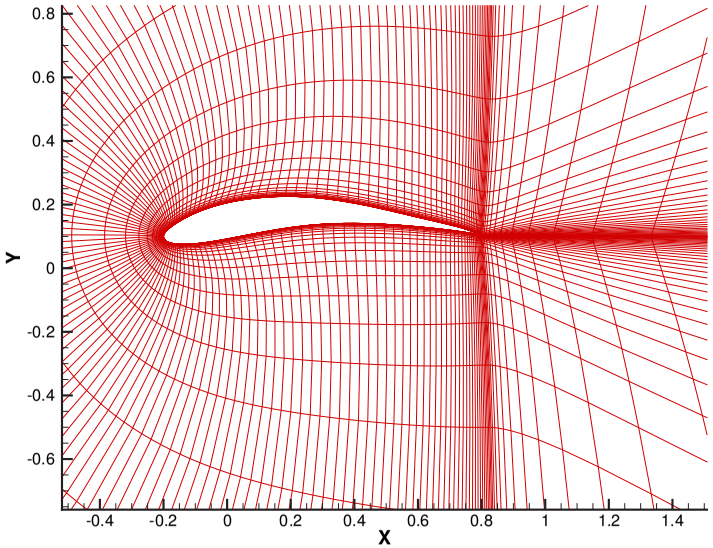
(a) RAE 2822 mesh



(b) Deformed and undeformed RAE 2822 airfoils



(c) linear elasticity



(d) spring analogy

Figure 5.3: The RAE 2822 airfoil undergoing large deformation

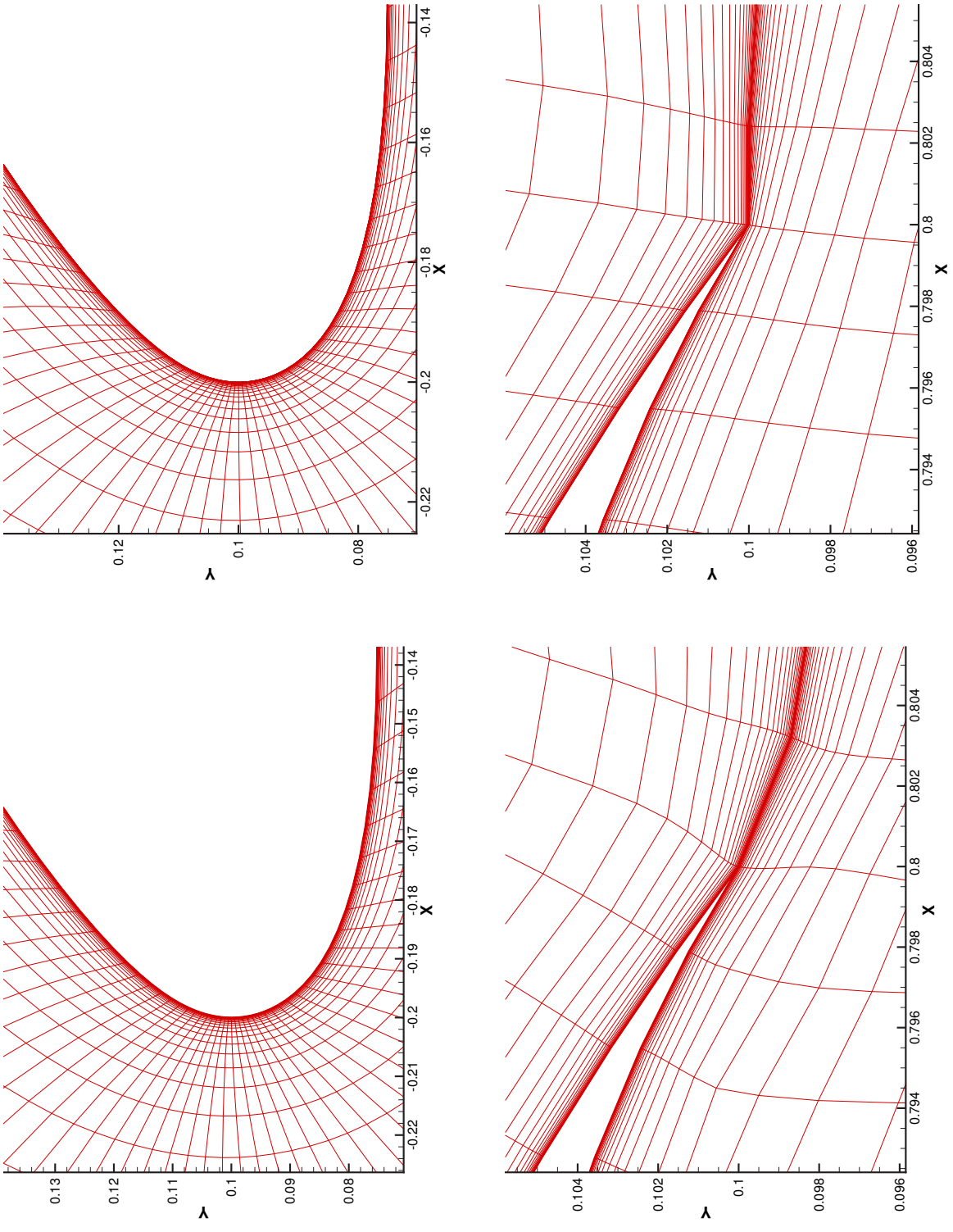
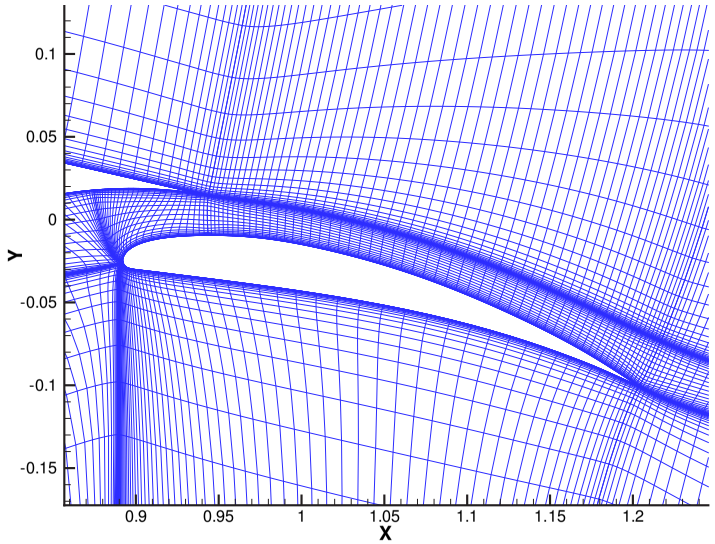
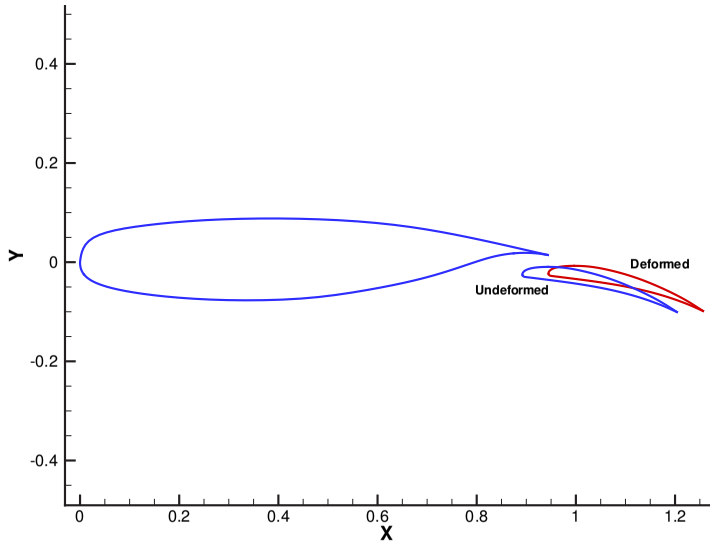


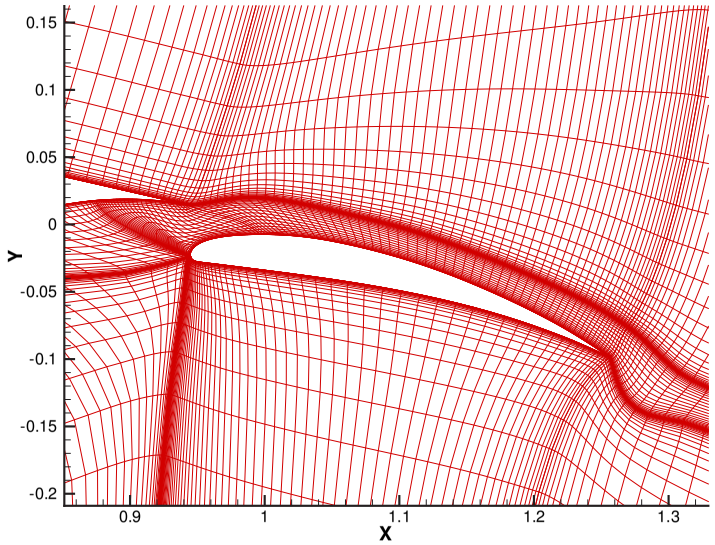
Figure 5.4: Deformed mesh of the RAE 2822 airfoil, showing the near-field region and the leading and trailing edge of the airfoil, using the linear elasticity method (left column) and the spring analogy method (right column)



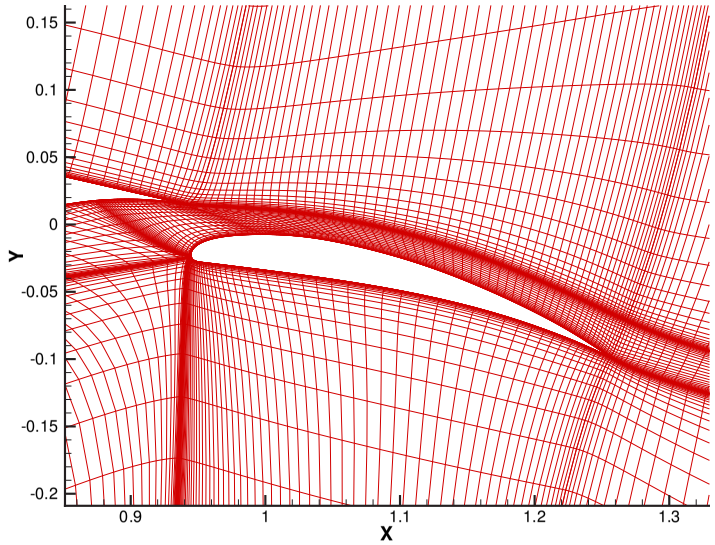
(a) Original mesh around the flap of the NLR 7301 airfoil



(b) Original and deformed flap configuration



(c) Linear elasticity method



(d) spring analogy method

Figure 5.5: NLR 7301 airfoil

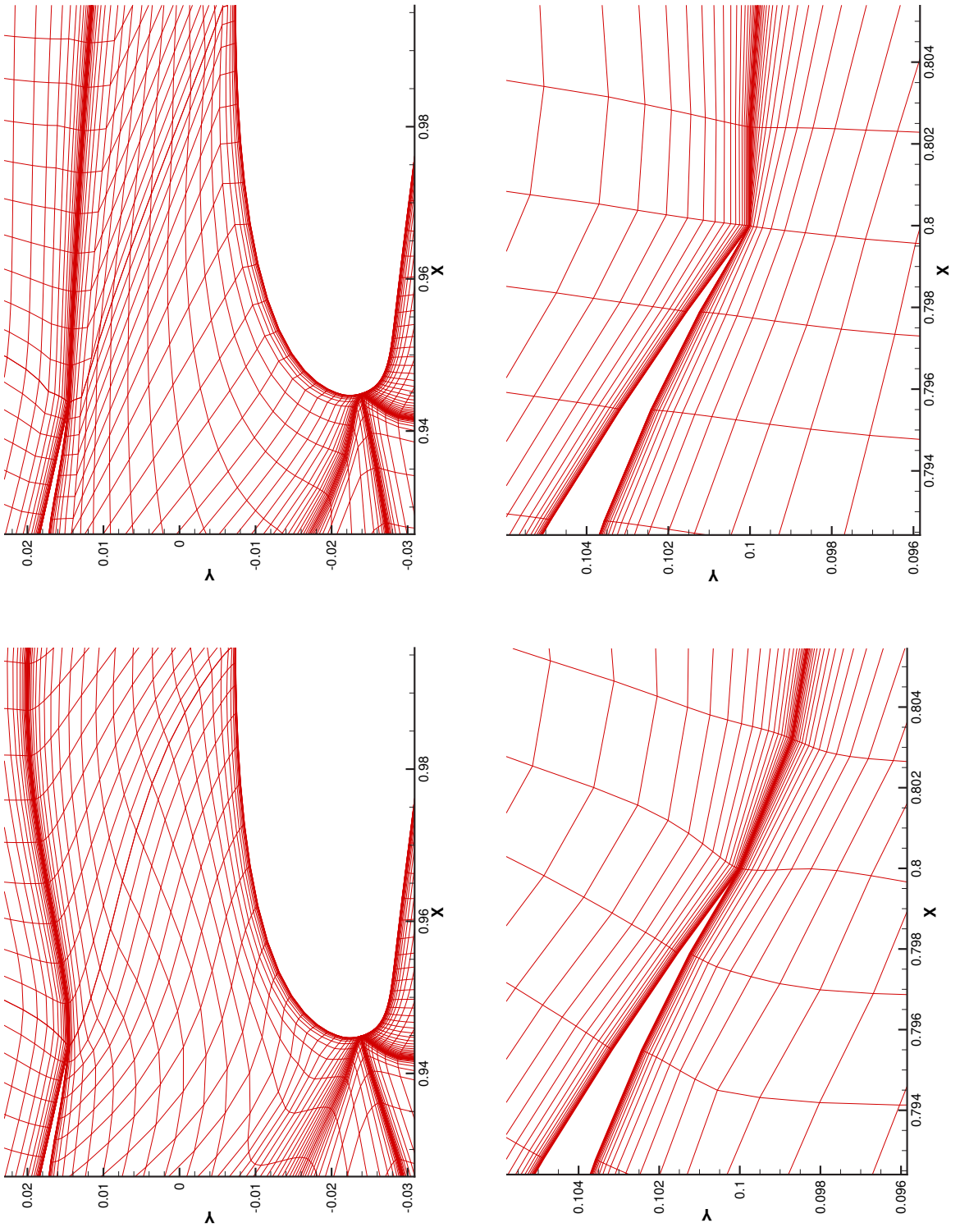


Figure 5.6: Close-up view of the leading and trailing edge of the NLR 7301 airfoil. Mesh obtained using the linear elasticity method (left) and the spring analogy method (right)

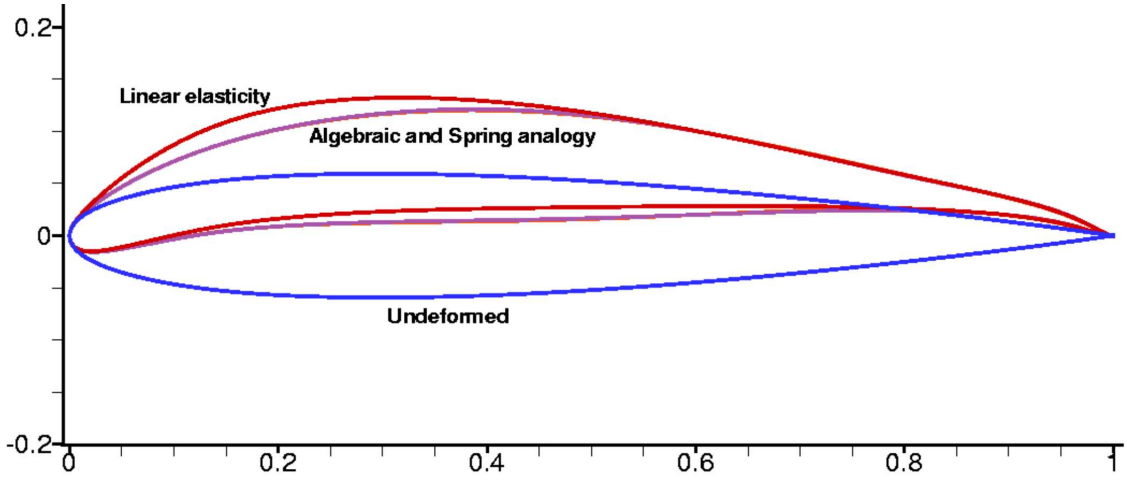


Figure 5.7: Original and optimized airfoil shapes obtained using different grid perturbation methods

5.2 Optimization Cases

Two aerodynamic shape optimization cases are presented to validate the correct implementation of the grid movement algorithms into the optimization programs: *Optima2D* for C-topology grids and *OptimaMB* for H-topology grids. These are relatively small deflection problems in that suitable mesh quality is maintained in all perturbation methods. The results obtained by the linear elasticity and spring analogy method are compared with those obtained by the algebraic method implemented by Nemec [21], see Section 4.2. The progression towards convergence as well as the optimization results are compared to ensure correct integration of the new mesh perturbation methods.

5.2.1 Maximization of Lift-to-Drag Ratio

The first case is the optimization of a NACA 0012 airfoil to maximize lift-to-drag ratio in subsonic flow conditions with a freestream Mach number of 0.25, a Reynolds number of 2.88×10^6, and an initial angle of attack of 9.0 degrees. The mesh has 201×45 nodes with 9 design variables, 8 of which are B-spline control points on the airfoil and the last one is the angle of attack. The following thickness constraints are imposed: a thickness to chord ratio of at least 4% at $x/c = 0.05$ and 0.65, 11% at $x/c = 0.35$, 2.6% at $x/c = 0.85$, 1.2% at $x/c = 0.95$, and 0.2% at $x/c = 0.99$. The optimized airfoil is shown in Fig. 5.7 and the optimization results are listed in Table 5.1.

The stiffness enhancement parameter, ϕ in Eq. 2.1, for the spring analogy method is

Table 5.1: Comparison of results for the lift-to-drag maximization problem of a NACA 0012 airfoil using different perturbation methods

Grid Source	Area[a]	Aspect Ratio[b]	c_L	c_D	α	c_L/c_D
Initial	1.46829×10^{-9}	25659.9	0.3	0.008	9.00	37.5
Algebraic	1.46641×10^{-9}	25511.9	1.228	0.02189	4.40	56.1
Linear elasticity	1.467×10^{-9}	25653.9	1.259	0.02212	4.17	56.9
Spring analogy	1.46745×10^{-9}	29955.1	1.217	0.02178	4.40	55.9

[a] minimum value
[b] maximum value

Table 5.2: Comparison of flow results for the final airfoils obtained from different perturbation methods

Grid Source	c_L	c_D	c_L/c_D
Algebraic	1.290	0.02358	54.7
Linear elasticity	1.265	0.02310	54.8
Spring analogy	1.290	0.02357	54.7

set to 5.0. Poisson's ratio and the power-law exponent in the linear elasticity method are set to 0.3 and 5.0, respectively. There is a relatively big difference between the final airfoil shape, shown in Fig. 5.7, obtained using the linear elasticity method and those obtained by the algebraic and spring analogy methods, the most pronounced differences occurring at a location about 25% chord. Comparing the grid in the near-field region, Fig. 5.8, it can be seen that the grid resulting from applying the linear elasticity grid perturbation method has better orthogonality. Consequently, better flow and optimization results are obtained, with almost 1.6% improvement in lift-to-drag ratio over the other methods.

The final airfoils obtained from each of the case discussed above are used to generate a mesh from which a flow solution is obtained. The resutls, tabulated in Table 5.2, indicate that the airfoil shape obtained using the linear elasticity method has the best lift-to-drag ratio, which supports the optimization results. However, the lift-to-drag ratio obtained in the analysis is much lower than what was predicted in the optimization with the linear elasticity method used to perturb the grid. This can be explained by examining and comparing the re-generated grid and that obtained from the optimization, shown in Fig. 5.9.

It must be noted that the same grid generation procedure used to create the initial

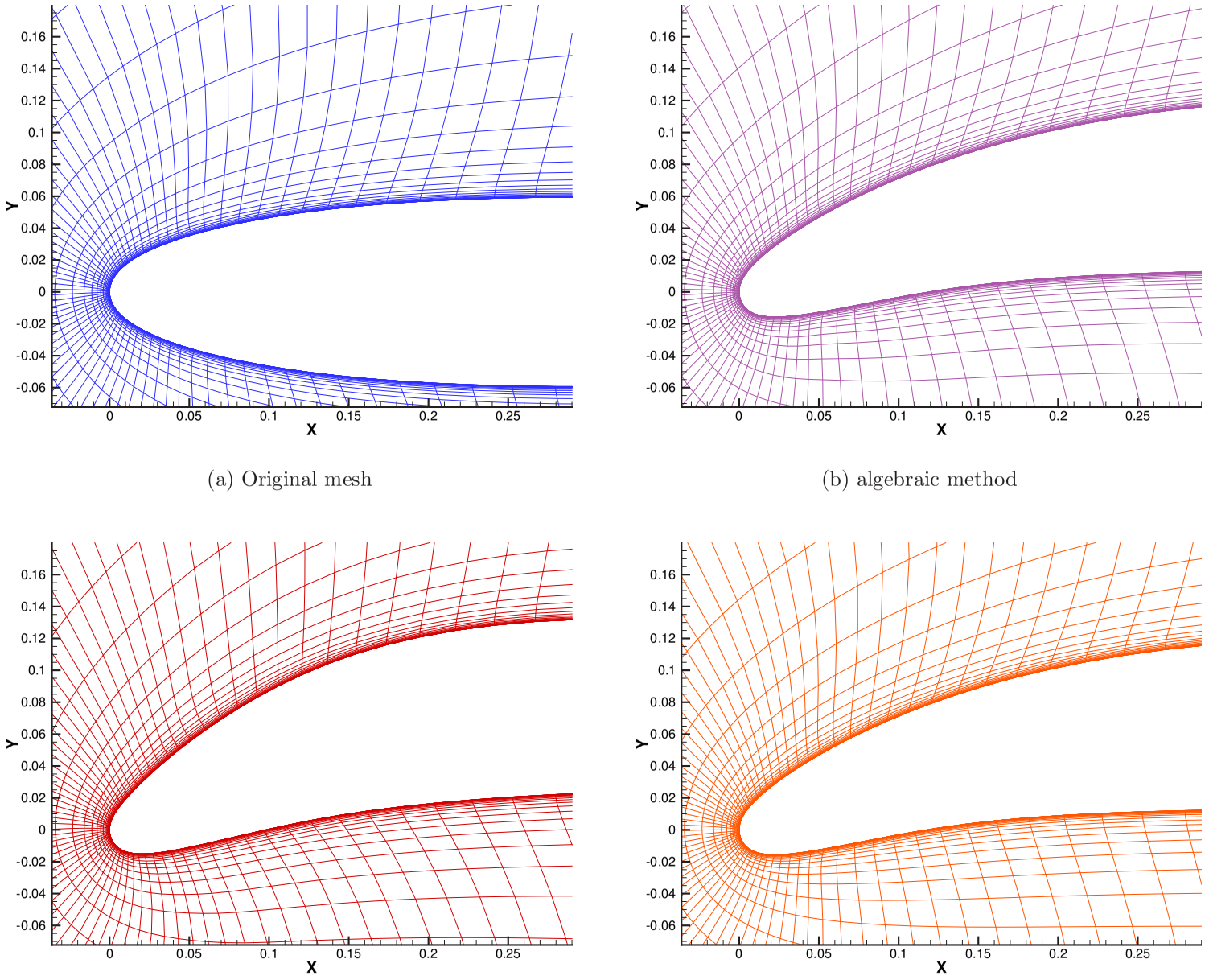
grids (in terms of number of elements, wall distance, grid node clustering, number of smoothing operations, *etc.*) is applied in the regeneration. It can be seen in Fig. 5.9 that the perturbed grid actually has better orthogonality over most of the airfoil surface than the regenerated grid. Although the regenerated grid has a slightly better minimum cell area of 1.50708×10^{-9}, the largest aspect ratio of 25766.7 is slightly worse than that of the perturbed grid.

5.2.2 Lift-Constrained Drag Minimization

The second case is the lift-constrained drag minimization problem of a NACA 0012 airfoil with only two design variables, the angle of attack and one B-spline control point at approximately 25% chord. The coefficient of lift, c_L, is fixed at 0.467, while the initial coefficient of drag, c_D, is 0.0178. The Mach number is 0.7 and Reynolds number 9.0×10^6. The mesh is a multi-block H-topology grid consisting of $29,510$ nodes, shown in Fig. 5.10. In all cases, the minimum cell area was maintained at 9.59519×10^{-12}. A comparison of optimization results is shown in Table 5.3.

In this case, the spring stiffness enhancement parameter is set to 10 in the spring analogy method, while the Poisson's ratio and power-law exponent in the linear elasticity method have values of 0 and 5, respectively. Since only two design variables were used, the major contribution to the reduction in aerodynamic drag is the angle of attack. Only a very slight change in the airfoil shape was visible. The differences in the optimization results are within 1%, with a slightly better drag value obtained using the linear elasticity method.

As was done for the previous optimization case, the optimized airfoils obtained using different grid perturbation methods are used to generate grids from which flow solutions are calculated and compared. The flow results are tabulated in Table 5.4. A similar trend is observed in the analysis results in that the regenerated grids yielded slightly lower lift, and higher drag, hence lower lift-to-drag ratio. However, they are consistent with the optimization results in predicting the lowest coefficient of drag on the optimized airfoil shape obtained using the linear elasticity grid perturbation method.



(a) Original mesh

(b) algebraic method

(c) Linear elasticity method

(d) spring analogy method

Figure 5.8: Original and deformed mesh of the NACA 0012 airfoil, showing the major differences in the deformed airfoil shape at about 25% chord

Table 5.3: Comparison of results for the lift-constrained drag minimization problem using different methods

Grid source	Maximum aspect ratio	c_D	c_D	α
Initial	45632.5	0.0178	26.2	6.00
Algebraic	45888.9	0.0137	34.2	3.19
Linear elasticity	45656.9	0.0136	34.3	3.18
Spring analogy	45039.8	0.0137	34.0	3.19

Table 5.4: Comparison of flow results for the final airfoils obtained from the lift-constrained drag minimization problem using different perturbation methods

Grid source	c_L	c_D	c_L/c_D
Algebraic	0.4306	0.01449	29.7
Linear elasticity	0.4302	0.01444	29.8
Spring analogy	0.4301	0.01455	29.6

5.3 3-Dimensional Test Cases

The first case is a deformation of the ONERA M6 wing, with a mesh consisting of $248,000$ nodes and $221,760$ hexahedral volumes. Fig. 5.11 shows the original surface mesh of the wing and the original with the deformed surface mesh superimposed. The deformed mesh has a 15° increase in leading and trailing edge sweep, a 5° increase in dihedral angle, and 15° twist. The results are listed in Table 5.5

The second case is also a deformation of an ONERA M6 wing, but with a denser mesh, consisting of $1,008,000$ nodes and $938,808$ hexahedra, shown in Fig. 5.12. The deformation includes a relative increase of 5° leading and trailing edge sweep at the root, and a 10° decrease in both dihedral angle and twist. The results are tabulated in Table 5.6.

In both cases, the stiffness enhancement parameter is set to 10 in the spring analogy method and a Poisson ratio of 0.3 was used in the linear elasticity method. The final shapes were obtained in one step, without the increments used in the previous examples. The results show that the spring analogy perturbation method yielded adequate meshes for simple deformation modes, such as the second example. For more complex deformation involving higher degrees of twist, as in the first case, the method yielded a

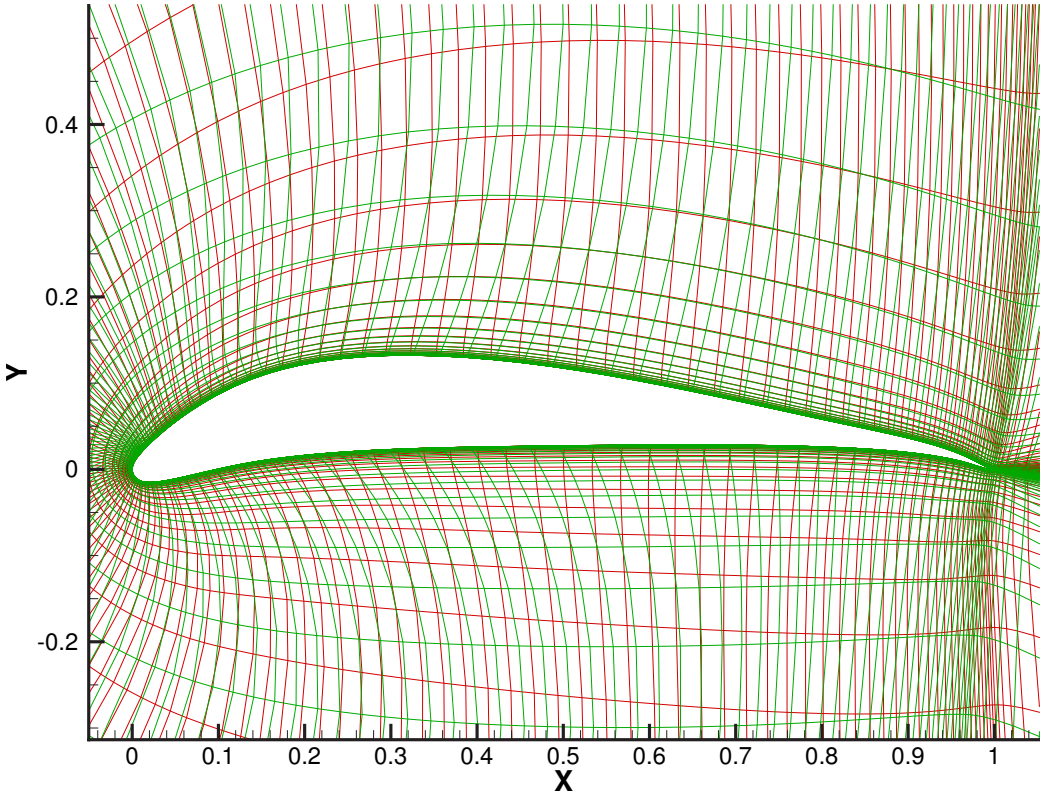


Figure 5.9: Re-generated grid (green) and perturbed grid using the linear elasticity method (red)

Table 5.5: Comparison of results for the deformation of the ONERA M6 wing (coarse mesh) using the linear elasticity and spring analogy methods

Grid source	Minimum area	Maximum aspect ratio	run-time[a]
Initial	1.08713×10^{-8}	15811.2	N/A
Linear elasticity	1.42573×10^{-8}	15899.3	1436
Spring analogy	2.02421×10^{-8}	518036	114

[a] measured in seconds

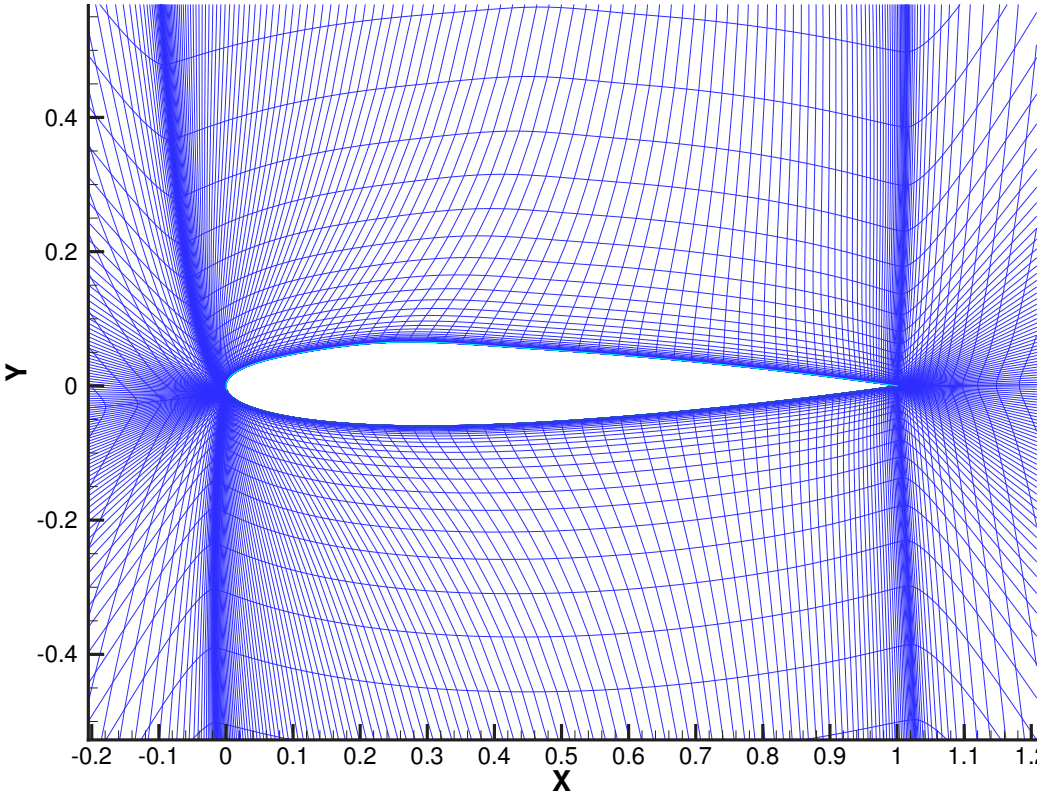
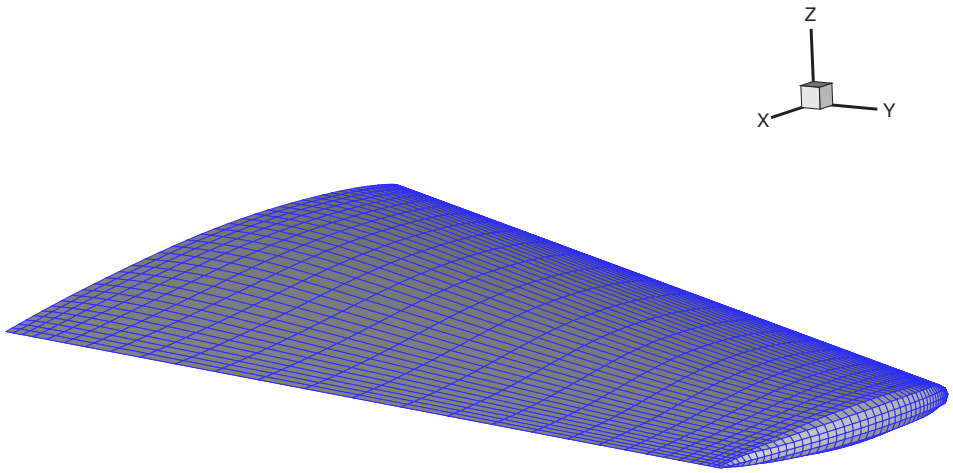


Figure 5.10: Original H-topology mesh of the NACA 0012 airfoil

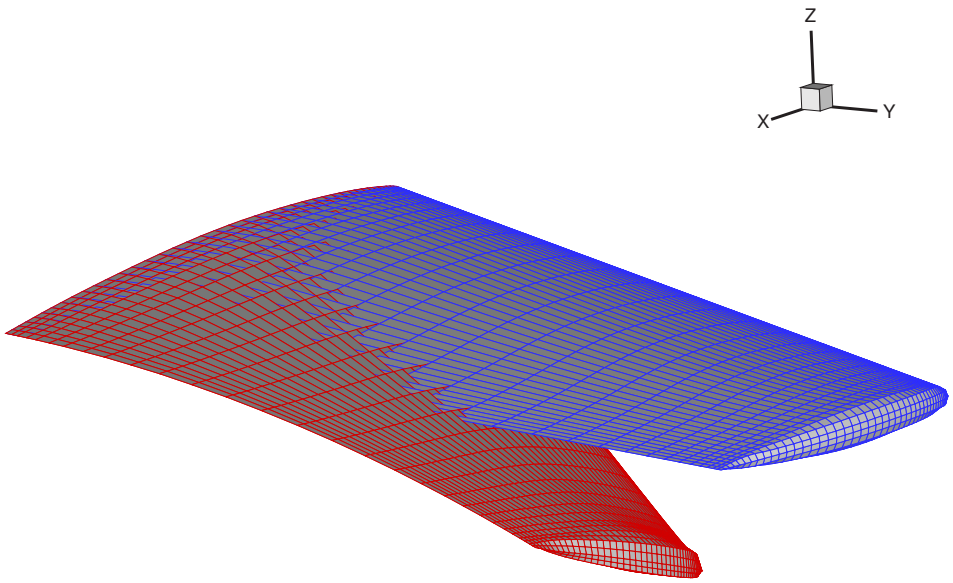
Table 5.6: Comparison of results for the deformation of the ONERA M6 wing (fine mesh) using the linear elasticity and spring analogy methods

Grid source	Minimum area	Maximum aspect ratio	run-time[a]
Initial	5.56571×10^{-9}	13419.6	N/A
Linear elasticity	5.18694×10^{-9}	13425.1	5739
Spring analogy	5.38777×10^{-9}	13481.8	667

[a] measured in seconds

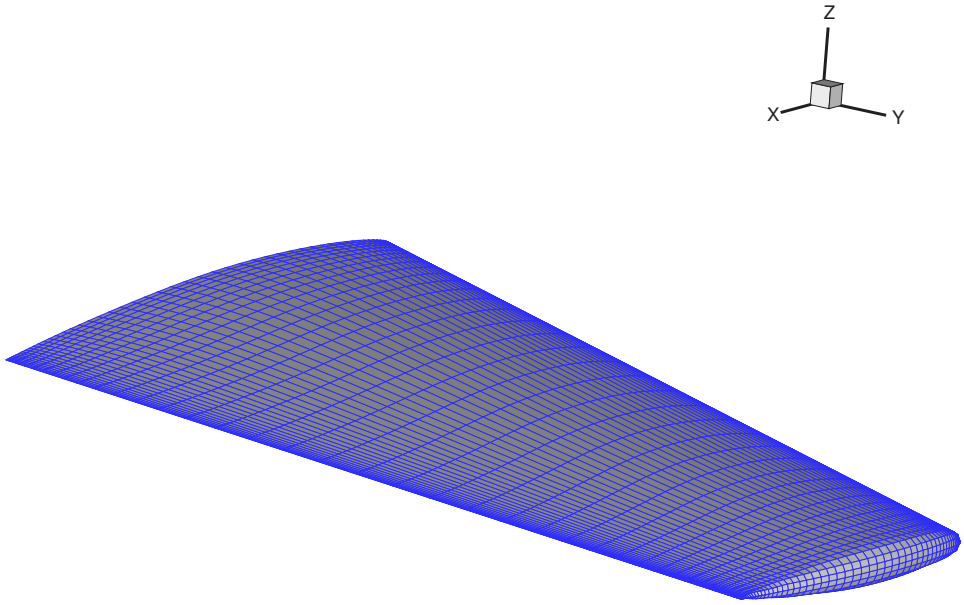


(a) original surface mesh

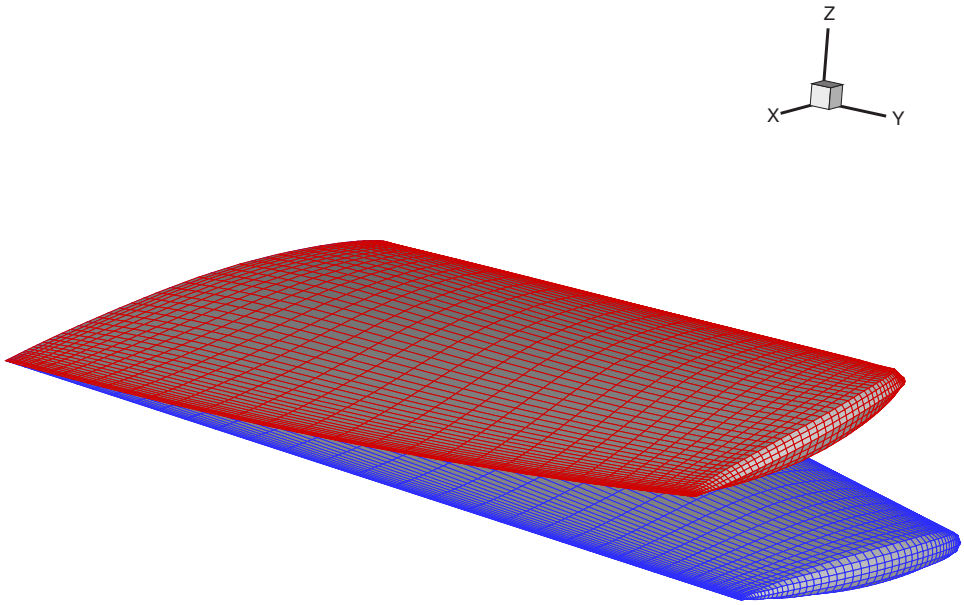


(b) deformed surface mesh superimposed on the original

Figure 5.11: ONERA M6 wing - coarse mesh



(a) original surface mesh



(b) deformed surface mesh superimposed on the original

Figure 5.12: ONERA M6 wing - fine mesh

poor quality mesh. In this case, the elements in the region near the tip of the airfoil are most distorted. The use of the linear elasticity model, on the other hand, yielded meshes with acceptable element quality in all cases, although at a computational cost of about 10 times that of the spring analogy method.

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

This research has considered the application of two mesh perturbation algorithms, the spring analogy and linear elasticity methods, which allow for automatic adjustment of interior nodes due to surface deformation within aerodynamic shape optimization software. The development of the equations describing the models was presented as was the implementation into the existing 2- and 3-dimensional shape optimization programs *Optima2D*, *OptimaMB*, and *Optima3D*. The algorithms have been applied to a number of test problems and the quality of the resulting meshes obtained from each algorithm compared. Additionally, several aerodynamic shape optimization simulations are performed to validate the correct implementation of the two grid perturbation methods. It was found that the performance of the spring analogy mesh movement algorithm is comparable to that of the algebraic method, with no significant improvement in grid quality. Overall, the linear elasticity method is the most robust tool for perturbing the grids. The use of the stiffening mechanism is essential in maintaining grid quality for more complex and large displacement problems, even though it incurs a relatively high computational cost. Due to its robustness, the linear elasticity method is ideal for use in optimization cases with large deformations, where the use of the spring analogy or the algebraic method would result in invalid meshes or line search stall due to formation of badly skewed elements.

6.2 Recommendations

The linear elasticity grid movement algorithm uses a constant value of the stiffening power (Eq. 3.30) and Poisson's ratio throughout the domain. In each of the test cases presented, these values had to be adjusted to suit the deformation mode to allow better control of element deformation. Similar adjustments had to be made for the stiffness enhancement parameter, ϕ, in Eq. 2.1 of the spring analogy method. An heuristic method should be developed to determine these values and adjust them accordingly.

Future developments should also include further optimization of the algorithm for speed. Since the global stiffness matrix of both the spring analogy and linear elasticity methods is linear, sparse and symmetric, it would be of great benefit to implement a more efficient matrix solver with solution acceleration techniques such as the Cuthill-McKee reordering scheme and an enhanced incomplete upper-lower preconditioning procedure to facilitate the solution of the algebraic equations.

Finally, the algorithms can be extended to unstructured grids to provide the optimizer greater flexibility in handling more diverse discretization domains.

References

[1] C. B. Allen, *An unsteady flow solver with algebraic grid motion for aeroelastic simulations*, ICAS 2002, International Council of the Aeronautical Sciences, Bristol, U.K., 2002. http://lu.fme.vutbr.cz/icas2002/PAPERS/R3.PDF.

[2] P. Z. Bar-Yoseph, S. Mereu, S. Chippada, and V. J. Kalro, *Automatic monitoring of element shape quality in 2-D and 3-D computational mesh dynamics*, Computational Mechanics, 27 (2001), pp. 378–395.

[3] J. T. Batina, *Unsteady Euler airfoil solutions using unstructured dynamic meshes*, AIAA Journal, 28 (1990), pp. 1381–1388.

[4] D. J. Benson, *An efficient, accurate, simple ALE method for nonlinear finite element programs*, Computer Methods in Applied Mechanics and Engineering, 72 (1989), pp. 305–350.

[5] C. O. E. Burg, *A robust unstructured grid movement strategy using three-dimensional torsional springs*, AIAA Paper 2004-2529, 34th AIAA Fluid Dynamics Conference, Portland, OR, June 2004.

[6] G. W. Burgreen and O. Baysal, *Three-dimensional aerodynamic shape optimization using discrete sensitivity analysis*, AIAA Journal, 34 (1996), pp. 1761–1770.

[7] C. Degand and C. Farhat, *A three-dimensional torsional spring analogy method for unstructured dynamic meshes*, Computers and Structures, 80 (2002), pp. 305–316.

[8] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne, *Torsional springs for two-dimensional dynamic unstructured fluid meshes*, Computer Methods in Applied Mechanics and Engineering, 163 (1998), pp. 231–245.

[9] D. Fudge, *A CAD-free and a CAD-based geometry control system for aerodynamic shape optimization*, Master's thesis, University of Toronto, 2004.

[10] D. M. Fudge, D. W. Zingg, and R. Haimes, *A CAD-free and a CAD-based geometry control system for aerodynamic shape optimization*, AIAA Paper 2005–0451, 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 2005.

[11] M. S. Gadala, M. R. Movahhedy, and J. Wang, *On the mesh motion for ALE modelling of metal forming processes*, Finite Elements in Analysis and Design, 38 (2002), pp. 435–459.

[12] S. Giuliani, *An algorithm for continuous rezoning of the hydrodynamic grid in arbitrary Lagrangian-Eulerian computer codes element programs*, Nuclear Engineering and Design, (1982), pp. 205–212.

[13] J. H. Kane, *Boundary Element Analysis in Engineering Continuum Mechanics*, Prentice Hall, Englewood Cliffs, NJ, 1994.

[14] Z. Kopal, *Numerical Analysis*, Chapman and Hall, London, 1961.

[15] C. S. Krishnamoorthy, *Finite Element Analysis: Theory and Programming*, Tata McGraw-Hill Publishing Company Limited, New Delhi, India, 2nd ed., 1994.

[16] W. K. Liu, H. Chang, J. Chen, and T. Belytschko, *Arbitrary Lagrangian-Eulerian Petrov-Galerkin finite elements for nonlinear continua*, Computer Methods in Applied Mechanics and Engineering, 68 (1988), pp. 259–310.

[17] D. G. Martineau and J. M. Georgala, *A mesh movement algorithm for high quality generalised meshes*, AIAA Paper 2004–0614, 42nd AIAA Fluid Dynamics Conference and Exhibit, Reno, NV, January 2004.

[18] S. A. Morton, R. B. Melville, and M. R. Visbal, *Accuracy and coupling issues of aeroelastic Navier-Stokes equations solutions on deforming meshes*, Journal of Aircraft, 35 (1998), pp. 798–805.

[19] M. Murayama, K. Nakahashi, and K. Matsushima, *A simple and robust unstructured dynamic mesh method with surface grid movement*, in 8th Interna-

tional Conference on Numerical Grid Generation in Computational Field Simulations, Sendai, Japan, June 2002.

[20] E. J. Neilson and W. K. Anderson, *Recent improvements in aerodynamic design optimization on unstructured meshes*, AIAA Journal, 40 (2002), pp. 1155–1163.

[21] M. Nemec, *Optimal Shape Design of Aerodynamic Configurations: A Newton-Krylov Approach*, PhD thesis, University of Toronto, 2003.

[22] M. Nemec and D. W. Zingg, *Multi-point and multi-objective aerodynamic shape optimization*, AIAA Paper 2002–5548, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Georgia, September 2002.

[23] ——, *Newton–Krylov algorithm for aerodynamic design using the Navier–Stokes equations*, AIAA Journal, 40 (2002), pp. 1146–1154.

[24] J. C. Nichols, *A three-dimensional multi-block Newton-Krylov flow solver for the Euler equations*, Master's thesis, University of Toronto, 2004.

[25] J. Reuther, J. J. Alonso, M. J. Rimlinger, and A. Jameson, *Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on distributed memory parallel computers*, Computers & Fluids, 28 (1999), pp. 675–700.

[26] J. J. Reuther, A. Jameson, J. J. Alonso, M. J. Rimlinger, and D. Saunders, *Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 1*, Journal of Aircraft, 36 (1999), pp. 51–60.

[27] J. Samareh, *Application of quaternions for mesh deformation*, NASA-2002-211646, NASA Langley Research Center, Hampton, Virginia, April 2002.

[28] K. Stein, T. Tezduyar, and R. Benney, *Mesh moving techniques for fluid-structure interactions with large displacements*, Journal of Applied Mechanics, 70 (2003), pp. 58–63.

[29] T. E. Tezduyar, M. Behr, S. Mittal, and A. A. Johnson, *Computation of unsteady incompressible flows with the finite element methods - space-time formula-*

tions, iterative strategies and massively parallel implementations, New Methods in Transient Analysis, 143 (1992), pp. 7–24.

[30] C. T. Wang, *Applied Elasticity*, McGraw-Hill Book Company Inc., New York, 1953.

[31] Z. Yang and D. J. Mavriplis, *Unstructured dynamic meshes with higher-order time integration schemes for the unsteady Navier-Stokes equations*, AIAA Paper 2005–1222, 41th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2005.

[32] D. Zeng and C. R. Ethier, *A semi-torsional spring analogy model for updating unstructured meshes*, in 9th Annual Conference of the CFD Society of Canada, vol. 1, Waterloo, ON, 2001, pp. 113–118.

[33] O. C. Zienkiewicz, *The Finite Element Method in Engineering Science*, McGraw-Hill Publishing Company Limited, London, 2nd ed., 1971.

[34] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method - Volume 1: The Basis*, Butterworth-Heinemann, Oxford, 2000.