Continuum Sensitivity Analysis using Boundary Velocity Formulation for Shape Derivatives

Mandar D. Kulkarni

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Aerospace Engineering

Robert A. Canfield, Chair
Mayuresh J. Patil, Co-Chair
Rakesh K. Kapania
Seongim S. Choi
Edward J. Alyanak

August 2, 2016
Blacksburg, Virginia

Keywords: continuum sensitivity, shape derivatives, shape optimization, aeroelasticity,

fluid-structure interaction

Continuum Sensitivity Analysis using Boundary Velocity Formulation for Shape Derivatives

Mandar D. Kulkarni

ACADEMIC ABSTRACT

The method of Continuum Sensitivity Analysis (CSA) with Spatial Gradient Reconstruction (SGR) is presented for calculating the sensitivity of fluid, structural, and coupled fluid-structure (aeroelastic) response with respect to shape design parameters. One of the novelties of this work is the derivation of local CSA with SGR for obtaining flow derivatives using finite volume formulation and its nonintrusive implementation (i.e. without accessing the analysis source code). Examples of a NACA0012 airfoil and a lid-driven cavity highlight the effect of the accuracy of the sensitivity boundary conditions on the flow derivatives. It is shown that the spatial gradients of flow velocities, calculated using SGR, contribute significantly to the sensitivity transpiration boundary condition and affect the accuracy of flow derivatives. The effect of using an inconsistent flow solution and Jacobian matrix during the nonintrusive sensitivity analysis is also studied.

Another novel contribution is derivation of a hybrid adjoint formulation of CSA, which enables efficient calculation of design derivatives of a few performance functions with respect to many design variables. This method is demonstrated with applications to 1-D, 2-D and 3-D structural problems. The hybrid adjoint CSA method computes the same values for shape derivatives as direct CSA. Therefore accuracy and convergence properties are the same as for the direct local CSA.

Finally, we demonstrate implementation of CSA for computing aeroelastic response shape derivatives. We derive the sensitivity equations for the structural and fluid systems, identify the sources of the coupling between the structural and fluid derivatives, and implement CSA nonintrusively to obtain the aeroelastic response derivatives. Particularly for the example of a flexible airfoil, the interface that separates the fluid and structural domains is chosen to be flexible. This leads to coupling terms in the sensitivity analysis which are highlighted. The integration of the geometric sensitivity with the aeroelastic response for obtaining shape derivatives using CSA is demonstrated.

Continuum Sensitivity Analysis using Boundary Velocity Formulation for Shape Derivatives

Mandar D. Kulkarni

GENERAL AUDIENCE ABSTRACT

Many natural and man-made systems exhibit behavior which is a combination of the structural elastic response, such as bending or twisting, and aerodynamic or fluid response, such as pressure; for example, flow of blood in arteries, flapping of a bird's wings, fluttering of a flag, and flight of a hot-air balloon. Such a coupled fluid-structure response is defined as aeroelastic response. Flight of an aircraft through turbulent weather is another example of an aeroelastic response. In this work, a novel method is proposed for calculating the sensitivity of an aircraft's aeroelastic response to changes in the shape of the aircraft. These sensitivities are numbers that indicate how sensitive the aircraft's responses are to changes in the shape of the aircraft. Such sensitivities are essential for aircraft design.

The method presented in this work is called Continuum Sensitivity Analysis (CSA). The main goal is to accurately and efficiently calculate the sensitivities which are used by optimization tools to compute the best aircraft shape that suits the customers needs. The key advantages of CSA, as compared to the other methods, are that it is more efficient and it can be used effectively with commercially available (nonintrusive) tools. A unique contribution is that the proposed method can be used to calculate sensitivities with respect to a few or many shape design variables, without much effort.

Integration of structural and fluid sensitivities is carried out first by applying CSA individually for structural and fluid systems, followed by connecting these together to obtain the coupled aeroelastic sensitivity. We present the first application of local formulation of CSA for nonintrusive implementation of high-fidelity aeroelastic sensitivities. The following challenging tasks are tackled in this research: (a) deriving the sensitivity equations and boundary conditions, (b) developing and linking computer codes written in different languages (C++, MATLAB, FORTRAN) for solving these equations, and (c) implementing CSA using commercially available tools such as NASTRAN, FLUENT, and SU2. CSA can improve the design process of complex aircraft and spacecraft. Owing to its modularity, CSA is also applicable to multidisciplinary areas such as biomedical, automotive, ocean engineering, space science, etc.

Dedicated to my better half, Kshitija and our son, Nilay.

Acknowledgements

I have been the most blessed graduate student with the best advisors, Dr. Robert Canfield and Dr. Mayuresh Patil. I started my PhD work in 2011 with Dr. Patil and very quickly realized that there are so many things to learn from him, such as his wit and humour, thinking out-of-the-box, apart from exceptional technical (and specifically debugging) skills. I will always remember his advise that whenever you are giving a presentation, make sure that the content is colour-coded, clear and simple such that all audience will learn at least something (and that a circle does not change to an oval due to aspect ratio issues $\smile$). After a year, I got an opportunity to start work with Dr. Canfield. So, the termination of one of the funding sources came as a blessing in disguise because I got a chance to work with both of them – two of the smartest people in the AOE department. Over the next four years, I have learnt from him to be thorough, meticulous, and at the same time be very courteous and caring. I have always been surprised of how both of them motivated me through positive feedback, especially during times when I was at a performance (local) minimum! I can only aim to be as down to earth and encouraging as my advisors. I cannot thank them enough for their support and guidance in so many aspects of a graduate student life.

My office companions and colleagues: David Cross, Anthony Ricciardi, Shaobin Liu, Will Walker, Jeff Ouellette, Ryan Seifert, Rob Grimm, Eric Stewart, Eddie Hale, Ben Names, Nick Albertson and David Sandler, have been tremendously helpful and made my office a fun place to work. They were always around when I needed to bounce off my ideas. I enjoyed my discussions with other friends in the department, Wrik Mallik, Ashok Kancharla, and Himanshu Shukla. I thank Nathan Love, Michael Barclift, Qingzhao Wang, Troy Bergin, and Ben Leonard for supporting my research in various ways.

The material in this dissertation is based on research sponsored by Air Force Research Laboratory (AFRL) under agreement number FA8650-09-2-3938 and by Air Force Office of

Scientific Research (AFOSR) under agreement number FA9550-16-1-0125DEF. I gratefully acknowledge the support of AFRL Senior Aerospace Engineers Dr. Raymond Kolonay, Dr. Philip Beran and Dr. Edward Alyanak, AFOSR Program Officer Dr. Jean-Luc Cambier, and director of the Collaborative Center for Multidisciplinary Sciences Dr. Rakesh Kapania.

I am very fortunate to have my better half, Kshitija Deshpande, alongside me during this journey to my PhD. There was no better motivation to start graduate school at Virginia Tech, than Kshitija being here earlier. Thinking back, I cannot imagine having completed my PhD without the strong, persistent inspiration and support that I got from Kshitija. Her perfectionism, great people-skills and habit of thinking much ahead of time are just some of the things that I have been trying to learn. Kalyani Nagaraj, Shubhangi Deshpande, Shvetha Soundararajan, Aniruddha Choudhary have been a part of our family and I will always miss going to their place for a quick cup of *chaha* (Indian tea) and long interesting conversations. John Harris, the families of Karuna and Keyur Joshi, Karen and Robert Howe were our best friends and always open to our knocking at their doors for any help or advise. In 2015, I got the biggest boost to successfully complete my PhD, when our son Nilay was born. My final year of PhD was definitely the most challenging and at the same time most rewarding. Nilay has taught me three things that made it possible for me to get through some of the most difficult problems of PhD research, (a) there is no thing such as fear, (b) always get up if you fall, and (c) be forgiving. My mother (*Aai*), father (*Appa*), sister's family- Kavita (*Tai*), Suhas (*Dada*), little Aarya and Aaradhya, and mother-in-law (*Aai*), father-in-law (*Baba*), and sister-in-law, Shreya have been very supportive and encouraging throughout my PhD studies. I cannot miss mentioning that *Baba's* stay with us for the past couple of months was a key in helping me finish up my PhD at Virginia Tech.

Finally, this acknowledgement would be incomplete without the mention of encouragement from all my friends, extended family and well-wishers both in the US and India. I am glad to have you all in my life!

Attribution

Dr. Robert A. Canfield is a co-author for the manuscripts of Chapters 2, 3, and 4. His contributions include strategic advice and review of the articles for technical accuracy, completeness, and grammatical correctness.

Dr. Mayuresh J. Patil is also a co-author for the manuscripts of Chapters 2, 3, and 4. His contributions include strategic advice and review of the articles for technical accuracy, completeness, and grammatical correctness.

Dr. David M. Cross is a co-author for the AIAA Technical Note that is included as Section 3.4 in Chapters 3. He contributions include guidance for understanding parts of the sensitivity boundary conditions of the axial bar problem and valuable inputs regarding the technical accuracy of the hybrid adjoint method.

Dr. Qingzhao Wang contributed to the NACA0012 example presented in Chapter 2 by providing simulation results from FLUENT.

Contents

1 Introduction **1**

1.1 Motivation 3

1.2 Research Objectives 8

1.3 Dissertation Organization 9

2 Nonintrusive Continuum Sensitivity Analysis for Fluid Applications **10**

2.1 Introduction 11

2.2 Local Continuum Shape Sensitivity Formulation 16

2.2.1 Governing Equations 16

2.2.2 Differentiation of Continuous Equations 17

2.2.3 Discretization of the Flow and Sensitivity Equations 20

2.2.4 On Accurate Boundary Conditions for CSA 21

2.2.5 Nonintrusive Implementation 22

2.2.6 Requirements of a Flow Analysis Code for Nonintrusive CSA 25

2.3 Sensitivity of Flow in a Lid-driven Cavity 26

2.3.1 Flow Analysis 26

2.3.2 Sensitivity Analysis 32

2.3.3 Flow Derivatives 34

2.3.4 Grid Convergence Study 38

2.4 Sensitivity of Flow Over NACA0012 Airfoil 39

2.4.1 Flow Analysis . . . 39
2.4.2 Sensitivity Analysis . . . 43
2.4.3 Discretizing CSE Boundary Conditions . . . 50
2.4.4 Results . . . 53
2.4.4.1 Geometry and Grids . . . 53
2.4.4.2 Flow Analysis . . . 55
2.4.4.3 Spatial Gradient of Velocity and the CSE Transpiration Boundary Condition . . . 60
2.4.4.4 Total Derivatives . . . 70
2.5 Conclusion . . . 72

3 Nonintrusive Continuum Sensitivity Analysis for Structural Applications: Direct and Adjoint Formulations 77

3.1 Introduction . . . 78
3.2 Local Continuum Sensitivity Analysis . . . 79
3.2.1 Continuous Domain Equations and Boundary Conditions . . . 79
3.2.2 Differentiation of the Continuous Equations . . . 80
3.3 Direct and Hybrid Adjoint Formulations . . . 83
3.3.1 Direct Formulation of Continuum Sensitivity Analysis . . . 83
3.3.2 Adjoint Formulation of Continuum Sensitivity Analysis . . . 86
3.3.3 Approximation of Spatial Gradients . . . 88
3.3.4 Application of Continuum Sensitivity Analysis to Two- and Three-Dimensional Structures . . . 91
3.3.5 Summary and Highlights of the Hybrid Adjoint Method . . . 92
3.4 Axial Bar . . . 93
3.4.1 Problem Description . . . 93
3.4.1.1 Design Velocity . . . 94
3.4.1.2 Distributed Axial Loading . . . 96

3.4.2 Analytical Sensitivity by Differentiating Exact Solution 97
3.4.3 Analytic Solution Using Direct Local CSA 98
3.4.3.1 Parameterization 1 98
3.4.3.2 Parameterization 2 101
3.4.4 Solution Using Hybrid Adjoint Formulation 102
3.5 Cantilever Beam with Solid Elements (Direct Formulation) 105
3.5.1 Problem Description 105
3.5.2 Primary Analysis 108
3.5.3 Continuum Sensitivity Analysis 111
3.5.4 Grid Convergence Study 122
3.6 Cantilever Beam with Solid Elements (Hybrid Adjoint Formulation) 132
3.6.1 Problem description 132
3.6.2 Result using finite difference method 133
3.6.3 Results obtained from direct formulation of CSA 134
3.6.4 Results obtained from hybrid adjoint formulation of CSA 135
3.7 Conclusion 139

4 Nonintrusive Continuum Sensitivity Analysis for Aeroelastic Shape Derivatives 140

4.1 Introduction 140
4.2 Local Continuum Shape Sensitivity Formulation for Aeroelastic System . . . 143
4.2.1 Governing Equations 143
4.2.2 Differentiation of Continuous Equations 145
4.2.3 Discretization of the Differentiated Equations 147
4.2.4 Two-dimensional Coupled Fluid-Structure Analysis 149
4.2.4.1 Structural Response 150
4.2.4.2 Fluid Response 151
4.2.4.3 Coupling of the Fluid and Structural Responses 153

4.2.4.4 Results . . . 154
4.2.5 Two-dimensional Coupled Fluid-Structure Sensitivity Analysis . . . 160
4.2.5.1 Structural Response Sensitivity Equations . . . 160
4.2.5.2 Fluid Response Sensitivity Equations . . . 161
4.2.5.3 Coupling of the Fluid and Structural Derivatives . . . 163
4.2.5.4 Results . . . 164
4.3 Conclusion . . . 166

5 Conclusions and Future Work **167**
5.1 Conclusions . . . 167
5.1.1 CSA for Fluid Applications . . . 167
5.1.2 Hybrid Adjoint CSA Formulation . . . 169
5.1.3 CSA for Aeroelasticity Applications . . . 170
5.2 Code Development . . . 171
5.3 Publications and Presentations . . . 172
5.4 Recommendations for Future Work . . . 172

Bibliography **174**

A Sensitivity of Flow in a Convergent-Divergent Nozzle **186**
A.1 Flow Analysis . . . 186
A.2 Derivatives of the Flow in a quasi-1-D Convergent Divergent Nozzle . . . 189
A.3 Results . . . 193

B Reduced Order Model for Unsteady Aerodynamics of Flapping Wing Micro Air Vehicle in Hover **196**
B.1 Introduction and motivation . . . 196
B.2 Unsteady Vortex Lattice Method . . . 200
B.2.1 No-penetration boundary condition . . . 202

B.2.2 Calculation of Γ_{b_i} and Γ_{w_1} 204
B.2.3 Calculation of Lift and Drag 205
B.3 Quasi-steady calculation 207
B.4 Proposed reduced order method 209
B.4.1 Momentum Disc Theory (MDT) calculation 209
B.4.2 Calculation using Reduced Wake and Momentum Disc Theory (RW-MDT) 211
B.5 Error metric 213
B.5.1 Average error metric (non-dimensionalized with RMS quasi-steady error) 213
B.5.2 Instantaneous error metric (non-dimensionalized with RMS quasi-steady error) 214
B.6 Results 214
B.6.1 Improvement of quasi-steady result using Momentum Disc Theory . . 215
B.6.2 Results for retaining only a fraction of the wake (RW calculation) . . 217
B.6.3 Results using the proposed reduced order scheme (RW-MDT calculation) 219
B.7 Uncertainty in Lift 223
B.8 Summary and conclusion 226

List of Figures

1.1 Sensitivity Analysis taxonomy 4

1.2 A representative N^2 diagram for Efficient Supersonic Air Vehicle (Allison et al., 2012). 6

2.1 Domain, Ω, with boundary Γ. 17

2.2 Illustration of SGR patches. 23

2.3 Flowchart for direct formulation of Continuum Sensitivity Analysis 23

2.4 Streamlines and horizontal velocities in the lid-driven cavity 28

2.5 Vertical velocities in the lid-driven cavity 28

2.6 Pressure in the lid-driven cavity 29

2.7 Manufactured solution for horizontal velocity. 31

2.8 Manufactured solution for vertical velocity. 32

2.9 Manufactured solution for pressure. 32

2.10 Manufactured solution for local derivative of horizontal velocity with respect to width of the cavity. 35

2.11 Manufactured solution for local derivative of vertical velocity with respect to width of the cavity. 35

2.12 Manufactured solution for local derivative of pressure with respect to width of the cavity. 36

2.13 Discretization error (DE) in the local derivative of horizontal velocity, $\epsilon_{u'}$, for the case of 9 layer SGR with fifth-order Taylor series, plotted for the finest mesh with 65×65 grid points. 36
2.14 Discretization error (DE) in the local derivative of vertical velocity, $\epsilon_{v'}$, for the case of 9 layer SGR with fifth-order Taylor series, plotted for the finest mesh with 65×65 grid points. 37
2.15 Discretization error (DE) in the local derivative of vertical velocity, $\epsilon_{p'}$, for the case of 9 layer SGR with fifth-order Taylor series, plotted for the finest mesh with 65×65 grid points. 37
2.16 Grid convergence study for the local derivatives of lid-driven cavity flow using a manufactured solution: Discretization error norms and rate of convergence are plotted against mesh refinement parameter h; Finest mesh is 65 X 65 nodes ($h = 1$) and coarsest mesh is 17 X 17 nodes ($h = 4$). 39
2.17 Domain, boundaries and unstructured mesh for flow over an airfoil. The inset figure shows the entire flow field with airfoil at the center. 41
2.18 Grid and control volume for vertex-centered finite volume scheme (Palacios et al., 2013). 42
2.19 Change in shape of NACA 0012 airfoil with design variable b. Note that the axes are scaled to magnify the shape perturbation. 44
2.20 Nonintrusive process when Jacobian matrix can be output from flow solver such as SU2. 48
2.21 Nonintrusive process when Jacobian matrix cannot be output from flow solver such as Fluent. 49
2.22 Design velocity, unit normal and derivative of unit normal on the airfoil surface. 54
2.23 Close-up of airfoil 512×512 O-mesh and details near the trailing edge (Vassberg and Jameson, 2010). 55

2.24 Flow solution over NACA0012 airfoil with 1024×1024 O-mesh obtained using SU2 and Fluent. 56
2.25 Mach line contours for flow over NACA0012 airfoil with 1024×1024 O-mesh obtained using SU2, $M_{min} = 0.0061794$, $M_{max} = 0.674184$, $\Delta M = 0.02$. . . . 57
2.26 Lift convergence obtained with different solvers. 58
2.27 Drag convergence obtained with different solvers. 59
2.28 Effect of functional error on rate of convergence. Here, (L) represents data for C_L, (D) represents data for C_D, "w/ err" represents erroneous values and "w/ O(2)" represents expected values with second-order rate of convergence. 61
2.29 Horizontal velocity contours of flow over NACA0012 airfoil for 1024×1024 O-mesh obtained using SU2. 62
2.30 Vertical velocity contours of flow over NACA0012 airfoil for 1024×1024 O-mesh obtained using SU2. 63
2.31 Horizontal velocity contours, near leading edge, of flow over NACA0012 airfoil for 1024×1024 O-mesh obtained using SU2. 63
2.32 Vertical velocity contours, near leading edge, of flow over NACA0012 airfoil for 1024×1024 O-mesh obtained using SU2. 64
2.33 Spatial gradients of velocity on the airfoil obtained using 2-layer ($L = 2$) SGR for 1024×1024 O-mesh. 65
2.34 Convective terms C_u and C_v appearing in the CSE transpiration BC, calculated using using 2-layer ($L = 2$) SGR, for the 1024×1024 O-mesh. 67
2.35 Convergence of spatial gradient $\partial u/\partial y$ at two locations using one, two and three layers for SGR; one near leading edge (left) and one near mid-chord (right). 68
2.36 Convergence of spatial gradient $\partial v/\partial y$ at two locations using one, two and three layers for SGR; one near leading edge (left) and one near mid-chord (right). 69

2.37 Transpiration velocity on the airfoil calculated using using 2-layer ($L = 2$) SGR with first-order Taylor series, for the 1024×1024 O-mesh. 70
2.38 Convergence of norm of the transpiration velocity, for 1, 2, and 3 layer SGR. 71
2.39 Total derivatives of density, velocity and pressure on the airfoil surface obtained using CSA and SGR with two layers and first-order Taylor series. "CSA (SU2)" and "CSA (FLUENT)" represent CSA derivatives with SU2 and FLUENT used for primary analysis, respectively. "FD (FLUENT)" and "FD (SU2)" represent finite difference derivatives calculated using FLUENT and SU2, respectively. 73
2.40 Convergence of lift derivative for 1, 2 and 3 layer SGR. 74
2.41 Convergence of drag derivative for 1, 2 and 3 layer SGR. 75

3.1 Domain, Ω, with boundary Γ. 80
3.2 Flowchart for implementation of the hybrid adjoint Continuum Sensitivity Analysis 89
3.3 Elastic bar with axial load 94
3.4 Different parametric representations of the elastic bar: (a): Parameterization 1: Material points move to the right, (b): Parameterization 2: Material points move to the left. 95
3.5 Types of linearly varying loads 96
3.6 Exact and finite element solutions to the axial bar problem. 98
3.7 Non-dimensional local derivative of the axial bar displacement with respect to its length. 104
3.8 Non-dimensional convective term for parameterization 1 and parameterization 2. 104
3.9 Discrete adjoint variable vectors. 105
3.10 Non-dimensional total derivatives of the axial bar displacement with respect to its length. 106

3.11 Cantilever beam modelled with solid elements and subjected to surface traction. 107
3.12 Design velocities, $\mathcal{V}_z$ for each design variable of the solid cantilever beam. The design velocities $\mathcal{V}_x$ and $\mathcal{V}_y$ are zero for all design variables. 109
3.13 Axial displacement of the beam of the solid cantilever beam on the center line, $y = 1$, $z = 2$. The symbol e indicates total number of elements. 112
3.14 Transverse z-displacement of the solid cantilever beam on the center line, $y = 1$, $z = 2$. The symbol e indicates total number of elements. 113
3.15 Axial displacement of the solid cantilever beam on the face $x = 0$ (where essential boundary conditions are applied). The symbol e indicates total number of elements. 114
3.16 Transverse z-displacement of the solid cantilever beam on the face $x = 0$ (where essential boundary conditions are applied). The symbol e indicates total number of elements. 115
3.17 Normal stress σ_x distribution in the solid cantilever beam for $e = 640$ elements. 116
3.18 Shear stress τ_{zx} distribution in the solid cantilever beam for $e = 640$ elements. The filled marker indicates the analytic values from the Airy stresses. 116
3.19 Normal stress τ_{yz} distribution in the solid cantilever beam for $e = 640$ elements. Note that the values of stress τ_{yz} are negligible. 117
3.20 Normal stress σ_z distribution in the solid cantilever beam for $e = 640$ elements. Note that the values of σ_z are one order of magnitude smaller than τ_{zx} and two orders of magnitude smaller than σ_x. 118
3.21 Distribution of the spatial gradient of the shear stress, $\frac{\partial \tau_{zx}}{\partial z}$, in the solid cantilever beam with $e = 640$ elements, calculated using SGR with second-order Taylor series and four patch layers. The filled markers indicate the analytic values from the Airy stresses. 121

3.22 Total derivatives of z-transverse displacement of the solid cantilever beam modeled with $e = 5120$ elements. "CSA-SGR BC" results were calculated using SGR with second-order Taylor series and four layer patches. 122

3.23 Total derivatives of normal stress $\dot{\sigma}_x$ of the solid cantilever beam modeled with $e = 5120$ elements, plotted near the root, on the line $x = 2$, $y = 1$. "CSA-SGR BC" results were calculated using SGR with second-order Taylor series and four layer patches. 123

3.24 Total derivatives of shear stress $\dot{\tau}_{zx}$ of the solid cantilever beam modelled with $e = 5120$ elements, plotted on the line $x = L/2$, $y = 1$. "CSA-SGR BC" results were calculated using SGR with second-order Taylor series and four layer patches. 124

3.25 Convergence of the transverse z-displacement at the tip of the solid cantilever beam, $x = L$, $y = 1$, $z = 2$. NC stands for number of cells or number of elements. 126

3.26 Convergence of the normal stress σ_x near the root of the solid cantilever beam, at the point $x = 2$, $y = 1$, $z = 4$. NC stands for number of cells or number of elements. 127

3.27 Convergence of the shear stress τ_{zx} near the central portion of the solid cantilever beam, at the point $x = L/2$, $y = 1$, $z = 2$. NC stands for number of cells or number of elements. 128

3.28 Convergence of the derivative transverse z-displacement with respect to the design variable b_1 at the tip of the solid cantilever beam, $x = L$, $y = 1$, $z = 2$. CSE BCs were calculated using SGR with second-order Taylor series and four layer patches. 129

3.29 Convergence of the derivative of normal stress $\dot{\sigma}_x$ with respect to the design variable b_1 near the root of the solid cantilever beam, at the point $x = 2$, $y = 1$, $z = 4$. CSE BCs were calculated using SGR with second-order Taylor series and four layer patches. 130

3.30 Convergence of the derivative of shear stress $\dot{\tau}_{zx}$ with respect to the design variable b_1 near the central portion of the solid cantilever beam, at the point $x = L/2$, $y = 1$, $z = 2$. CSE BCs were calculated using SGR with second-order Taylor series and four layer patches. 131

3.31 Design velocities for the two parameterizations of the cantilever beam. . . . 132

3.32 Step size study to determine the best step size: 2-Norm of the total derivative $\dot{w}$ on the center line of the beam, $y = 1$, $z = 2$, is plotted. 133

3.33 Total derivative of transverse displacement on the center line of the beam, $y = 1$, $z = 2$, obtained using finite difference method. 134

3.34 Nondimensional local derivative of the transverse displacement of cantilever beam on the center line, with respect to its length, calculated using direct CSA formulation. 136

3.35 Nondimensional total derivative of the transverse displacement of cantilever beam on the center line, with respect to its length, calculated using direct CSA formulation. 136

3.36 Nondimensional local derivative of the transverse displacement of cantilever beam on the center line, with respect to its length, calculated using hybrid adjoint CSA formulation. 138

3.37 Nondimensional total derivative of the transverse displacement of cantilever beam on the center line, with respect to its length, calculated using hybrid adjoint CSA formulation. 138

4.1 Domain, Ω, with boundary Γ. 144

4.2 Flexible airfoil in uniform flow. 149

4.3 Close-up of O-mesh around the flexible airfoil with 64×64 cells (Vassberg and Jameson, 2010). 156
4.4 Structural mesh for the flexible airfoil with 138 triangular finite elements. . . 156
4.5 Difference in successive values of C_L for weak coupling iterations. The successive difference for the last six iterations is exactly zero, hence not shown on the log scale of the Y axis in the plot. 157
4.6 Difference in successive values of C_D for weak coupling iterations. The successive difference for the last nine iterations is exactly zero, hence not shown on the log scale of the Y axis in the plot. 157
4.7 Aeroelastic response of the flexible airfoil. 158
4.8 Airfoil boundary movement during weak coupling iterations. 159
4.9 Vector plot to illustrate structural deformation of the flexible fluid-structure interface at steady-state. The deformation is with respect to the undeformed NACA 0012 shape. Vector lengths indicate magnitude of displacements and are scaled for better visualization. 159
4.10 Derivatives of aeroelastic response of the flexible airfoil obtained using finite difference method. 165

A.1 A convergent divergent nozzle (Anderson, 1995) 186
A.2 Density, Mach number and pressure in a convergent divergent nozzle. Solid lines indicate analytical solution, whereas the circles indicate numerical (Euler) solution. 193
A.3 Step size convergence of Euler solution for the convergent-divergent nozzle. . 194
A.4 Derivatives of the nozzle flow. 195
A.5 Rate of convergence of derivatives of the nozzle flow, with Roe's flux difference scheme. 195
A.6 Rate of convergence of derivatives of the nozzle flow, with Van Leer's flux vector splitting scheme. 195

B.1 Airfoil flapping motion 200
B.2 Comparison of inflow obtained from MDT with inflow from total unsteady calculation 210
B.3 Improvement of quasi-steady lift using MDT 215
B.4 Effect of considering MDT inflow in a quasi-steady calculation 216
B.5 Effect of retaining a fraction of the shed wake 217
B.6 Effect of retaining a very small fraction of the shed wake 218
B.7 Effect of retaining a fraction of the shed wake and also using MDT inflow . . 219
B.8 Effect of retaining only a fraction of the shed vortices, judged from the value of e_1 at the end of 10 oscillations 220
B.9 Parametric study; effect of changing rotation amplitude, α_0 221
B.10 Parametric study; effect of changing semi-chord, b 221
B.11 Parametric study, effect of changing number of bound vortices, n_b 222
B.12 Parametric study, effect of changing number of time steps per oscillation, nt_{po} 222
B.13 Lift generated in the 10th oscillation, starting at ten different location on the stroke plane. The calculation is done according to the UVLM and all vortices are retained. 223
B.14 Comparison of wake shed by airfoil at the end of 15 oscillations, starting at different locations 224
B.15 Lift generated in the 10th oscillation, starting at ten different location on the stroke plane. 'TU' = Total Unsteady calculation, 'QS' = Quasi-steady calculation, 'MDT' = calculation with Momentum Disc Theory inflow, 'RW' = Reduced Wake calculation with retention of 1 oscillation, 'RW-MDT' = Reduced Wake calculation with retention of 1 oscillation and MDT inflow . 225

List of Tables

2.1 Details of 2-D flow examples 15

2.2 Survey of CFD solvers for nonintrusive CSA. 27

2.3 Details of the flow analysis 55

3.1 Details of the cantilever beam finite element model for the coarsest grid. . . 107

3.2 Natural BC applied on the faces of the 3-D (solid) cantilever beam. 111

4.1 Details of the aeroelasticity analysis 155

Chapter 1

Introduction

Aircraft are used for a variety of applications such as for transport, military, experimentation, advertisement, agriculture, and so on. In general, aircraft may be propelled by human power, gas, battery or solar power; aircraft sizes may range from a few milimeters to a couple of hundred feet; and they can have fixed, rotary or flapping wings. With so many parameters, design of an aircraft is one of the most challenging engineering tasks. Aircraft may cost from a couple of hundred dollars, for a model aircraft, to billions of dollars for a commercial aircraft. As a reference, the cost of a Boeing 747-8 aircraft in 2016 was approximately $360 million (The-Boeing-Company, 2016). Most of commercial aircraft design are naturally geared towards making a safe and efficient aircraft at the lowest cost. Military aircraft have design requirements such as high speed, high maneuverability and long endurance. New configurations such as sensorcraft, joined-wing and blended wing body are being designed to meet these ever-growing requirements. However, this gives rise to the need for tools and computational models that can simulate the response of such new configurations. Good design tools and models should be able to predict the performance as well as the different failure modes of the novel aircraft within sufficient accuracy.

Consider the NASA Helios prototype, which was flown in 2003 to test the newly designed High Altitude Long-Endurance (HALE) type aircraft. The Helios prototype took off well

but encountered turbulence about 30 minutes later which resulted in its crash. The accident investigation report published by NASA in 2004 (Noll et al., 2004) highlights that a root cause for this failure was "Lack of adequate analysis methods [which] led to an inaccurate risk assessment of the effects of configuration changes leading to an inappropriate decision to fly an aircraft configuration highly sensitive to disturbances." It is also stated that "the board determined that the mishap resulted from the inability to predict, using available analysis methods, the aircraft's increased sensitivity to atmospheric disturbances such as turbulence, following vehicle configuration changes required for the long-duration flight demonstration." From this it is clear that tools are required for not only predicting the behavior, i.e. response of the configuration, but also for predicting the sensitivity of the response to different configurational changes. The latter is a key motivation for research presented in this dissertation.

Design derivatives, also sometimes referred to as sensitivities, describe how a particular response would change with respect to a design variable. Aircraft design variables can be broadly classified as sizing (also known as value), shape, and topology variables. Sizing variables change the size of a member, such as thickness of ribs or spars, cross sectional areas of components, while shape variables change the shape of the physical domain of the aircraft, such as airfoil shape, wing span, wing sweep angle, fuselage cross section, empennage configuration and so on. Topology variables determine the optimal layout of material and connectivity inside a design domain. Topology optimization is a separate topic in itself and is not considered in the current work. Realistically, there can be hundreds and thousands of such design variables, which finally define and can possibly alter the characteristics of an aircraft. To understand the complete problem and conduct design optimization over the complete design space requires the calculation of sensitivity of all performance and constraint metrics with respect to all the design variables. This gives an idea of the complexity of aircraft design sensitivity analysis.

The main feature of a sizing optimization problem is that the domain of the design

model is known a priori and fixed throughout the optimization process, with the goal of obtaining optimum size of specific members in the model. On the other hand, in shape optimization, the boundary and the interior points in the domain move due to a shape design variable, which offers higher design freedom than that in sizing optimization. This movement of the material points with respect to the shape variables, defined as geometric sensitivity or design velocity, is an important factor in shape optimization. Consequently, it is more computationally challenging to estimate the sensitivity of a response to a shape design variable than to a sizing design variable. Complexity of calculations increases when the said response is a coupled response such as the fluid-structure interaction, also called as aeroelastic response. This introduces the topic of the current dissertation: "Integration of Aeroelastic and Geometric Sensitivity for Shape Optimization." Although the examples presented here are primarily related to aircraft shape derivatives, the methods can be applied to any system that exhibits a response which is sensitive to the shape of its physical domain.

1.1 Motivation

Sensitivity analysis plays an important role in gradient-based optimization techniques. In fact, convergence of a gradient-based shape optimization depends on the accuracy of gradients of the performance and constraint functions with respect to the design variables. Apart from its application in optimization, sensitivity analysis is useful in areas such as error-based grid adaptation, gradient-enhanced kriging, characterization of complex flows, fast evaluation of nearby-flows (Duvigneau and Pelletier, 2006), and uncertainty analysis. Sensitivity analysis methods can be broadly categorized as numeric methods (finite difference, complex step), analytic methods (discrete, continuum) or automatic differentiation methods, as shown in the taxonomy in Figure 1.1. With the exception of complex step method, analytic methods are generally favored over numeric methods because of their higher accuracy. Furthermore, analytic methods have the following advantages: (a) there is no need of con-

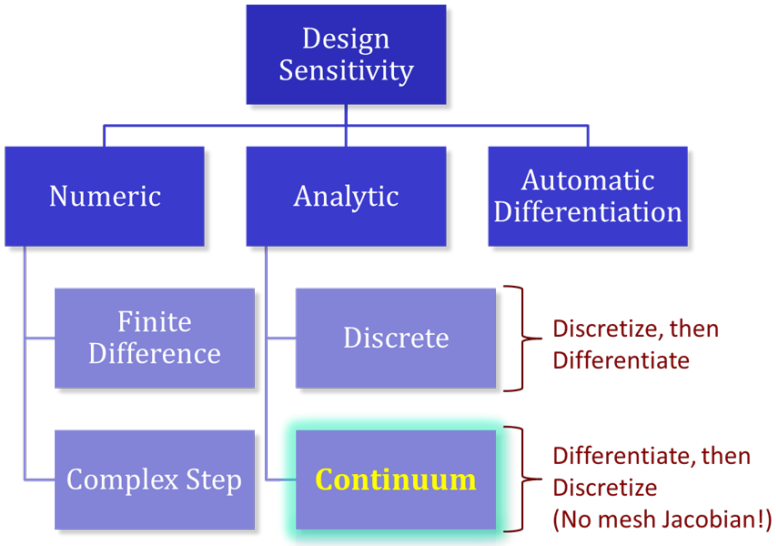

Figure 1.1: Sensitivity Analysis taxonomy

vergence study for choosing the correct step size (required for the finite difference method), and (b) there is no requirement of the analysis code to handle complex number operations (required for the complex step method). Among the analytic methods, the discrete analytic method involves discretizing the governing equations, followed by differentiation. Since differentiation occurs after discretization, calculation of the change in the mesh with respect to a shape design variable, also known as mesh sensitivity or mesh Jacobian, is required as a part of the sensitivity analysis procedure. Evaluation of mesh sensitivity often requires computationally expensive calculations. Additionally, intimate knowledge of the analysis procedure is required for implementation of the discrete analytic method because it is based on "discretize then differentiate" approach. Automatic differentiation method requires the actual source code for its implementation. Compared to these, Continuum Sensitivity Analysis (CSA) may offer a better alternative. This has motivated researchers to investigate the use of CSA for various systems.

Aircraft response calculation is an interdisciplinary process and it usually involves use of multiple tools and software programs for getting the effects of aircraft's structural, aerody-

namic, propulsion and control characteristics. Typically, these programs are run numerous times during design iterations with data being transferred back and forth. This is better explained by the N^2 diagram as shown in Figure 1.2. The data that is transferred involves not only the response values, but also the response derivatives from each software program. Commercial programs such as NASTRAN, FLUENT, ABAQUS rely mostly on numeric or discrete methods of obtaining design derivatives, which are often computationally expensive and may also be inaccurate in certain cases. However, since commercial codes can only be used as black-boxes, it is not always possible to implement analytic sensitivity calculation, if not already present, in such software for obtaining accurate derivatives. Due to this reason, an analytic sensitivity analysis procedure that can be implemented with black-box tools can be of great value in aircraft design. This motivates the nonintrusive, or black-box, implementation of CSA.

CSA has been successfully used to compute shape derivatives of structural response (Dems and Haftka, 1988; Haug et al., 1986; Arora, 1993; Choi and Kim, 2005) and fluid response (Borggaard and Burns, 1994, 1997; Stanley and Stewart, 2002; Turgeon et al., 2005; Duvigneau and Pelletier, 2006), as well as more recently, aeroelastic response (Etienne and Pelletier, 2005; Liu and Canfield, 2016; Cross and Canfield, 2015). The motivation for using CSA for shape sensitivity of aeroelastic response is twofold: (a) gradients are analytic (accurate and more efficient than finite difference) and (b) mesh sensitivity calculation is avoided (a drawback of the discrete analytic and semi-analytic shape sensitivity approaches).

In the boundary velocity formulation of the CSA method, the solution of the sensitivity equations is driven by three terms on the boundaries and on the interfaces that change shape: (1) the geometric sensitivity, (2) the spatial gradients of the response, and if applicable, (3) material derivative of the function prescribed on the boundary. The geometric sensitivity or design velocity is defined as the rate of movement of material points with respect to the shape design variable, and is either supplied by the designer, or can be obtained once the shape design variables are defined (Kulkarni et al., 2014b). The spatial gradients of

Representative ESAV N^2 Diagram

Propulsion	Flow behind inlet shocks	Flow-through panels' data	Temp., Press., Alt., M, Dimen.	Engine weight	Engine weight	Engine data in flight envelope	Thrust available for vectoring		Exhaust speed and temp.		Thrust, Altitude, Mach #, BPR, etc.
Cowl and Inlet	**Geometry**	Configuration, Mach #, Alt.	Cowl, Aft deck	Configuration	Tank and engine	Wing area	Control effector data	Configuration	Noise shielding factor	Configuration, Fuel volume	Configuration
		Aerodynamics	Skin temp., Loading	Aerodynamic Loading		Aero. data in flight envelope	Stability derivatives				
			EEWS	EEWS weight	Structural weight - EEWS						
				Structures	Structural weight - other					Structural failure	
					Weights	Aircraft weight in flight envel.	Inertial properties				Take-off gross weight
						Mission Performance				Fuel volume, Req. maneuvers	Feasibility
							Stability and Control			Added removed fuel volume	Feasibility
								RADAR Cross Section		RCS metric	
									Noise	Noise output	
										Constraints	Feasibility
Thrust, Altitude, Mach #, BPR, etc.	Configuration										**Optimization**

Figure 1.2: A representative N^2 diagram for Efficient Supersonic Air Vehicle (Allison et al., 2012).

the response, may be calculated using a technique called Spatial Gradient Reconstruction (SGR). The third term is present if a value is prescribed on the boundary or interface through a shape parameter dependent function. In that case, the material derivative of the prescribed function is required on the boundary or interface. Cross and Canfield (Cross and Canfield, 2014, 2016) demonstrated that using SGR, analytic shape sensitivity can be computed without any information about the finite element formulation, element shape functions, or how the element shapes change with changes to the mesh, when the shape of the domain changes. In other words, for the finite element method, the presented CSA method is element-agnostic and does not require the knowledge of how the mesh changes with shape design variables.

Borggaard and Burns, Duvigneau and Pelletier, and others (Borggaard and Burns, 1994, 1997; Stanley and Stewart, 2002; Turgeon et al., 2005; Duvigneau and Pelletier, 2006) have applied boundary velocity CSA for obtaining shape sensitivities of compressible, inviscid and viscous flows. They also highlight the computational advantage of boundary velocity CSA, as it involves solving a linear system of equations with the same system matrix as the primary flow analysis. As a result, computational effort is mostly in forming the new right side. Although they have stated the possibility of making use of this property for black-box sensitivity analysis, the specific steps and algorithm are missing. In the current work, boundary velocity formulation of CSA together with SGR has been used for nonintrusive implementation of CSA for flows computed using Euler and Navier-Stokes equations.

CSA has also been applied for solving aeroelastic shape derivatives (Liu and Canfield, 2011, 2016; Cross and Canfield, 2014). Liu and Canfield (Liu and Canfield, 2013a) have explained how the choice of boundary velocity against domain velocity CSA can be justified for aeroelastic sensitivity calculations. However, those studies involved flow computation based on typical section aerodynamics or potential flow theory. This involves lower-fidelity results than for cases involving viscous and compressible flow. This motivates the work on obtaining aeroelastic sensitivities using CSA, which incorporates Euler and Navier-Stokes

flow computation.

As introduced earlier, aircraft design involves hundreds of sizing and shape design variables. This means that derivatives of a few objective functions (such as lift or drag) may be required with respect to each of these design variables. In current practice, an adjoint formulation of sensitivity analysis is used for such a case. It has been shown in literature (Lozano and Ponsin, 2012; Duivesteijn et al., 2005) that the boundary conditions associated with continuous adjoint method are often difficult to formulate. This motivates the need of an adjoint formulation in which the boundary conditions are simplified. Thus, the hybrid adjoint formulation of CSA, which addresses this issue, thus introduced. The specific research objectives that are addressed in this dissertation are stated next.

1.2 Research Objectives

The main objective of the current research is to aid accurate, computationally efficient, non-intrusive (or black-box) sensitivity calculations. This will be a stepping stone towards the larger goal of "bringing multi-physics based analysis forward, into the conceptual design process," as mentioned by Alyanak and Kolonay (Alyanak, 2012; Alyanak and Kolonay, 2012). They also mention that this is important because "decisions made during [the] relatively short period of time [spent during conceptual design] have a very large impact on the life-cycle cost and performance of an aircraft." A detailed literature survey was done based on the motivation and gaps in literature presented in the previous section. This has led to the following research objectives.

1. Formulate nonintrusive CSA with SGR for computation of high-fidelity flow derivatives, and demonstrate its application using Computational Fluid Dynamics (CFD) tools that are based on finite volume discretizations.

2. Develop hybrid adjoint formulation of CSA.

3. Formulate and demonstrate nonintrusive CSA with SGR for aeroelastic response derivatives involving high-fidelity flow and structural computation.

Each of these objectives is addressed as a separate journal paper and presented as three chapters in this dissertation.

1.3 Dissertation Organization

The organization of this dissertation is as follows. Chapter 2 presents the nonintrusive formulation of CSA for computing CFD flow derivatives. Examples of flow in a quasi-one-dimensional nozzle, two-dimensional flow over an airfoil, and flow in a lid-driven cavity are used to demonstrate this approach. Chapter 3 illustrates the hybrid adjoint formulation of CSA for efficiently and nonintrusively computing design derivatives of a few performance functions with respect to many design variables. This is demonstrated with the example of an one-dimensional axial bar, a two-dimensional plate, and a three-dimensional cantilever beam. Chapter 4 extends the nonintrusive CSA formulation to compute aeroelastic shape sensitivities. This is demonstrated with the example of an airfoil supported by beam, and a flexible airfoil in uniform flow. Finally, in Chapter 5, conclusions and technical contribution of the current research are highlighted and recommendations are made for future work.

Chapter 2

Nonintrusive Continuum Sensitivity Analysis for Fluid Applications

ABSTRACT

Continuum Sensitivity Analysis (CSA) provides an analytic method of computing derivatives for structures, fluids and fluid-structure-interaction problems with respect to shape or value parameters. Its advantage of bypassing the calculation of mesh sensitivities is evident in shape optimization problems. Moreover, the Spatial Gradient Reconstruction (SGR) method makes it amenable to nonintrusive implementation. In this paper we explain the nonintrusive CSA procedure for calculating the material derivatives of one- and two-dimensional flow variables with respect to shape design parameters. The examples demonstrate application of nonintrusive CSA to flow problems involving Euler (compressible inviscid) and Navier-Stokes (incompressible viscous) equations, structured and unstructured grids, finite volume and finite difference spatial discretizations, and implicit and explicit temporal discretizations. Factors such as the accuracy of the sensitivity transpiration boundary condition, weak implementation of the boundary conditions in the finite volume framework, and use of approximate flux Jacobian matrix, all of which affect the accuracy of the total derivatives, are discussed. The sensitivity analysis is done nonintrusively using codes such as SU2, FLUENT, and an in-house code. This work establishes the use of such black-box codes, for obtaining flow sensitivities using the CSA approach.

2.1 Introduction

Sensitivity analysis plays an important role in gradient-based optimization techniques. Convergence of a shape optimization problem depends on the accuracy of gradients of the performance functions. Sensitivity analysis methods can be broadly categorized as numeric methods (finite difference, complex step), analytic methods (discrete analytic, continuum) or automatic differentiation methods. In general, analytic methods are favored over numeric methods because of their higher accuracy and lower computational cost. Furthermore, analytic methods have the following advantages: (a) there is no need of convergence study for choosing the correct step size (required for the finite difference method), and (b) there is no requirement of the analysis code to handle complex number operations (required for the complex step method). Among the analytic methods, the discrete analytic method involves discretizing the governing equations followed by differentiation. Hence, for shape design variables it suffers from the disadvantage of calculating mesh sensitivity. Additionally, intimate knowledge of the analysis procedure is required for implementation of the discrete analytic method. Automatic differentiation requires the actual source code for its implementation. Continuum Sensitivity Analysis (CSA) offers a better alternative.

The motivation for using CSA for shape sensitivity is twofold: (a) gradients are analytic (accurate and more efficient than finite difference) and (b) mesh sensitivity calculation is avoided (a drawback of the discrete analytic shape sensitivity approach). CSA can be derived in two forms (Liu and Canfield, 2016), the local form, also known as the boundary velocity sensitivity equation method (Choi and Kim, 2005) or Eulerian sensitivity equation method (Pelletier et al., 2008), and the total form, also known as the domain velocity sensitivity equation method (Choi and Kim, 2005) or Lagrangian sensitivity equation method (Charlot et al., 2012). The local form was known to give poor flow shape derivative results as compared to total form CSA, due to challenges arising from accurate evaluation of spatial gradients on the boundaries that change shape (Duvigneau and Pelletier, 2006). Charlot et al. (2012) mention that although these gradients can be calculated using reconstruction

techniques (Duvigneau and Pelletier, 2006), the local CSA method becomes expensive in 2-D and impractical in 3-D. While this may be true if the spatial gradients are to be calculated at all points in the domain, typically local CSA requires spatial gradients to be calculated only on the interfaces or boundaries that change shape. Cross and Canfield (2014) demonstrated that using the Spatial Gradient Reconstruction (SGR) technique, which was inspired from the work of Duvigneau and Pelletier (2006), accurate and efficient spatial gradients can be evaluated, thus resolving one of the challenges that was preventing researchers to use local CSA. In the current work, the same SGR technique has been used to evaluate spatial gradients of flow variables on shape variable dependent boundaries.

Most the previous work in the area of Computational Fluid Dynamics (CFD) flow sensitivity that used local CSA, was done with the finite element (Duvigneau and Pelletier, 2006) or finite difference discretizations (Borggaard and Burns, 1997, 1994). The focus of current work is to apply CSA with SGR to solve for fluid flow derivatives calculated using Euler and Navier-Stokes equations with finite volume discretization, for which a very limited number of studies are available. Gobal et al. (2015) used CSA with an immersed boundary finite volume scheme to solve for the derivatives of solution to the linear thermal diffusion equation. They used the finite difference method and the Heaviside functions to compute the spatial derivatives terms that appear in the sensitivity equations. However, since the boundary conditions for their problems do not depend on the design variables, the advantage of using their approach for computing flow derivatives on shape dependent boundaries is unclear. Challenges arise when finite volume discretization is used because boundary conditions are typically imposed with a weak or integral approach (Hirsch, 1990; Palacios et al., 2013). Villa et al. (2012) mention that the weak approach is used by finite volume-based CFD solvers because it allows imposition of a suitable boundary flux, rather than a boundary state as in the strong approach, which is preferable for setting up the boundary fluxes for the finite volume method. Specifically, Villa (2009) studied the drawbacks of weak imposition of the no-penetration boundary condition on 1D hyperbolic equation solutions using the

finite volume approach and stated that the difference between the computed physical values and the actual values of velocity could be significant. This can be particularly detrimental for solution of sensitivity equations, because it adds a source of error in the imposition of boundary conditions, apart from the error in the spatial gradients and the discretization error. We discuss this issue in detail and highlight areas where improvement in using SGR can be made.

Finally, we discuss another contribution of the current work, namely the nonintrusive, or black-box, implementation of CSA. The discrete analytic method, the complex step method and the automatic differentiation method all need modification to the source code for calculating design derivatives. The operator-overloading approach to performing automatic differentiation is the most convenient but its implementation is typically 10-35 times slower than the original algorithm (Gay, 2005). Newer advancements on operator-overloading have made the automatic differentiation process faster, still the computational cost is typically 4 times, or more depending on the application, of the original algorithm (Hogan, 2014; Albring et al., 2016). On the other hand, although the finite difference method can be used with black-box codes, it is not preferred because of dependence on the finite difference step-size. There can be a considerable savings in computational efforts and programming time if shape derivatives are computed with the same code that was used for flow analysis. This would also be useful in a collaborative environment that requires the use of the same code by multiple groups for calculating derivatives with respect to different shape design variables.

The work that most closely resembles a nonintrusive formulation of local CSA for finite volume discretization is by Godfrey and Cliff (2001). Their approach was to obtain the flow solution from a CFD solver and use it to solve the sensitivity equations separately while constructing exact Jacobians. This idea of solving for the shape derivatives separately (external to the flow solver) was a key inspiration of the current work. However, the current work differs from the work by Godfrey and Cliff (2001) in several ways. First, Godfrey and Cliff (2001) use the finite difference method to calculate the spatial derivatives in the sens-

itivity equations. While this may be possible for the case of a structured mesh, which they have used, the finite difference method cannot be used for unstructured meshes (Becker and Ashcroft, 2014; Duvigneau and Pelletier, 2006). Second, Godfrey and Cliff (2001) studied inconsistent turbulent modeling for flow and sensitivity equations. Specifically, they studied the interdependence of the flow and sensitivity solutions when the sensitivity equation was derived from a turbulence model different from the one that provides the discrete flow data. Out of the three cases studied, they report that their sensitivity analysis method provides best results when the turbulent modeling is consistent in the flow solver and sensitivity solver, with the exception of the case when the Baldwin-Lomax flow solution and eddy viscosity are incorporated into the Spalart-Allmaras sensitivity formulation. Thirdly, Godfrey and Cliff (2001) solve the linear sensitivity equations using an iterative approximate linear solver. Finally, Godfrey and Cliff (2001) report results for a single mesh and compared the CSA results with finite difference derivatives. In another closely related work, Borggaard and Burns (1994, 1997) discuss how the continuum sensitivity equations (CSEs) can be solved by minimum modifications to a flow analysis code and state that the same grid and computational scheme can be used to approximate the sensitivities. However, the implementation details for a finite volume based CFD solver are not addressed.

In the current work we demonstrate a nonintrusive implementation (without modifying the "black-box" analysis source code) of CSA for analyzing fluid systems with a focus on the use of commonly used CFD codes that use finite volume discretization for solving the flow variables and their shape derivatives nonintrusively. Particularly, we are interested in finding out how the use of a Jacobian matrix that is inconsistent with the Jacobian matrix used for the flow solution, affects the sensitivity results. Owing to the large size of the Jacobian matrix, the capability of exporting or storing the Jacobain matrix is not available with many flow solvers. For our nonintrusive approach, the Jacobian has to be either exported from the CFD solver (if this capability is available) or constructed outside of the solver. The exact Jacobian for Euler equations is known (Hirsch, 1990) and hence it can be constructed

Table 2.1: Details of 2-D flow examples

Flow	Example	Eqs.	Time discretization	Space discretization	Software	Code verification
Incompressible viscous	Flow in a lid-driven cavity	Navier-Stokes	Explicit	Finite difference	In-house (Matlab)	Grid convergence (MMS)
Compressible inviscid	Flow through convergent-divergent nozzle	Euler	Euler implicit	Finite volume	In-house (Matlab)	Exact solution available
Compressible inviscid	Flow over NACA0012 airfoil	Euler	Euler implicit	Finite volume	SU^2 (C++)	Grid convergence

outside of the flow solver, however it may not be consistent with the matrix used for the flow solution. We studied this aspect of the nonintrusive CSA approach. Once the Jacobain matrix was obtained, the linear sensitivity equations were solved using LU factorization with partial pivoting.

Details of the proposed nonintrusive CSA approach are given in Section 2.2. CSA was applied to two types of steady flows: (a) incompressible viscous flow illustrated with the example of a lid-driven cavity, and (b) compressible inviscid flow, illustrated with the examples of 1-D flow through a convergent divergent nozzle and 2-D flow over NACA0012 airfoil. The details about the flow analysis for these examples is given in Table 2.1. A manufactured solution was used to verify the CSA code for the lid-driven cavity example as described in Section 2.3. The NACA0012 example is described in Section 2.4. The convergent divergent nozzle example was used to demonstrate CSA for value parameters and is described in Appendix A.

2.2 Local Continuum Shape Sensitivity Formulation

2.2.1 Governing Equations

The partial differential equations governing fluid flow can be written compactly as

$$\boldsymbol{R}\left(\boldsymbol{u}, t; b\right) = \boldsymbol{\mathcal{A}}\left(\boldsymbol{u}, L\left(\boldsymbol{u}\right)\right) - \boldsymbol{f}\left(\boldsymbol{x}, t; b\right) = 0 \quad \text{on } \Omega, \tag{2.1}$$

with the corresponding boundary conditions (BCs)

$$\boldsymbol{\mathcal{B}}\left(\boldsymbol{u}, L\left(\boldsymbol{u}\right)\right) = \boldsymbol{g}\left(\boldsymbol{x}, t; b\right) \quad \text{on } \Gamma, \tag{2.2}$$

where the vector of dependent (state) variables $\boldsymbol{u}(\boldsymbol{x}, t; b)$ are functions of the spatial and temporal independent coordinates, $\boldsymbol{x}$ and t, respectively, and depend implicitly on design variable b. The domain and boundary in Cartesian space are shown in Figure 2.1. The linear differential operator L has terms such as $\left\{\frac{\partial}{\partial t}, \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial y^2}, \ldots\right\}$ that appear in the governing equations or boundary conditions. $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ are algebraic or integral operators acting on $\boldsymbol{u}$ and $L\left(\boldsymbol{u}\right)$. In general the differential operators in the Equations (2.1) and (2.2) can be nonlinear. The distributed body force applied on the system is given by $\boldsymbol{f}$ in (2.1). The general BCs in (2.2) can be either Dirichlet (essential or geometric) such as a prescribed value $\boldsymbol{\mathcal{B}}_e\left(\boldsymbol{u}\right) \equiv \boldsymbol{u}|_{\Gamma_e} = \boldsymbol{g}_e \equiv \bar{\boldsymbol{u}}$ on the boundary Γ_e, or may involve a differential operator for Neumann (nonessential or natural) BCs such that $\boldsymbol{\mathcal{B}}_n\left(L\left(\boldsymbol{u}\right)\right) = \boldsymbol{g}_n$ on the boundary Γ_n. Since the current work involves steady-state flow and sensitivity analysis, the time term t in these equations represents pseudo-time.

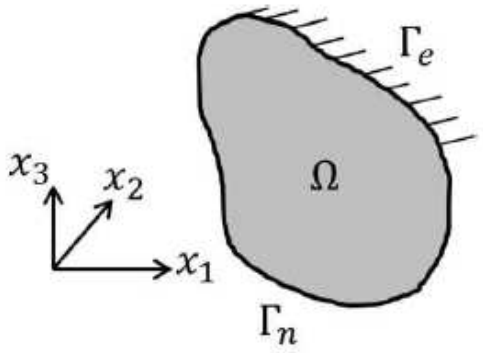

Figure 2.1: Domain, Ω, with boundary Γ.

2.2.2 Differentiation of Continuous Equations

The focus of the current work is to obtain design derivatives of flow variables at points in the domain with respect to a limited number of design variables. Hence, we use the direct formulation of CSA as explained next. However, an adjoint formulation of CSA (Kulkarni et al., 2016) is also available, which allows one to calculate derivatives of a limited number of performance measures with respect to many design variables using the same CSA boundary conditions presented here. Consider the problem of obtaining the derivative of the steady-state response $\boldsymbol{u}(\boldsymbol{x}; b)$ with respect to design parameter b at all points in the domain. The response depends on the spatial variable $\boldsymbol{x}$ and it may have an explicit or implicit dependence on the shape variable b, as indicated by the semicolon. Since steady response derivatives are of interest, the time variable t is suppressed in the further discussion. The boundary velocity (local) formulation of CSA produces CSEs that are posed in terms of the local derivatives of the response, $\boldsymbol{u'} = \partial \boldsymbol{u}/\partial b$. Hence, solution of the CSEs yields the local derivative. The total or material derivative $\dot{\boldsymbol{u}} = D\boldsymbol{u}/Db$ is then obtained by adding the convective term to the local derivative.

$$\frac{D\boldsymbol{u}}{Db} = \frac{\partial \boldsymbol{u}}{\partial b} + \sum_{i=1}^{3} \frac{\partial \boldsymbol{u}}{\partial x_i}\frac{\partial x_i}{\partial b} \quad \Longleftrightarrow \quad \dot{\boldsymbol{u}} = \boldsymbol{u'} + \boldsymbol{\nabla}_{\boldsymbol{x}}(\boldsymbol{u}) \cdot \boldsymbol{\mathcal{V}} \tag{2.3}$$

The convective term consists of the spatial gradients of the response $\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u} = \partial\boldsymbol{u}/\partial\boldsymbol{x}$, and the geometric sensitivity or design velocity $\boldsymbol{\mathcal{V}}(\boldsymbol{x}) = \partial\boldsymbol{x}/\partial b$, which depends on the parametrization of the computational domain. For value design parameters, the convective term goes to zero, since the design velocity is zero, and so the material derivative is same as the local derivative. However, for shape design variables, the design velocity is not zero and hence there is a need to calculate the convective term wherever necessary.

CSA is based on the philosophy of "differentiate and then discretize" and involves differentiating Equations (2.1) and (2.2) with respect to b, followed by discretization and solution of the resulting discretized system. Based on the type of differentiation, CSA is categorized as either local form CSA or total form CSA (Liu and Canfield, 2016). The local form CSA involves partial differentiation of Equation (2.1), while the total form CSA involves total differentiation. Due to the advantages of the local form CSA over the total form CSA (Cross and Canfield, 2014), the current work focuses on the local form CSA. The CSEs are obtained by partial differentiation of Equation (2.1) as

$$\frac{\partial \boldsymbol{R}}{\partial b} = \frac{\partial \boldsymbol{\mathcal{A}}(\boldsymbol{u}, L(\boldsymbol{u}))}{\partial \boldsymbol{u}}\boldsymbol{u'} + \frac{\partial \boldsymbol{\mathcal{A}}(\boldsymbol{u}, L(\boldsymbol{u}))}{\partial L}L(\boldsymbol{u'}) - \frac{\partial \boldsymbol{f}(\boldsymbol{x}, t; b)}{\partial b} = 0. \tag{2.4}$$

Since the material boundary changes due to a change in the shape design parameter, the boundary conditions for the CSEs are obtained by total or material differentiation of the original boundary conditions and then moving the convective terms to the right side

$$\frac{\partial \boldsymbol{\mathcal{B}}}{\partial u}\boldsymbol{u'} + \frac{\partial \boldsymbol{\mathcal{B}}}{\partial L}L(\boldsymbol{u'}) = \dot{g}(\boldsymbol{x}, t; b) - \boldsymbol{\mathcal{V}}(\boldsymbol{x}) \cdot \left(\frac{\partial \boldsymbol{\mathcal{B}}}{\partial u}\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u} + \boldsymbol{\nabla}_{\boldsymbol{x}}(\boldsymbol{\mathcal{B}}(L(\boldsymbol{u})))\right), \tag{2.5}$$

where $\dot{g}(\boldsymbol{x}, t; b)$ is the material derivative of the prescribed boundary condition, typically zero for Dirichlet boundary conditions. Nevertheless, even when the the boundary condition (2.2) is homogeneous $\left(\boldsymbol{u}|_{\Gamma_e} = 0\right)$, the CSE boundary condition (2.5) is in general non-homogeneous due to the convective term: $\boldsymbol{u'}|_{\Gamma_e} = \dot{\boldsymbol{g}}_e - \boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u} \cdot \boldsymbol{\mathcal{V}}(\boldsymbol{x})$, even for $\dot{\boldsymbol{g}}_e = 0$. The commutation of derivatives on the left side of Eq. (2.5) is possible when the derivatives are local. The

CSEs (2.4) with the boundary conditions (2.5) form a linear system of equations in terms of sensitivity variable $\boldsymbol{u'}$, which can be solved by the same or different numerical method used for solving the analysis problem.

Eqs. (2.4) and (2.5) may be restated as

$$\frac{\partial \boldsymbol{R}}{\partial b} = \boldsymbol{\mathcal{A}}_b \left(\boldsymbol{u}, L\left(\boldsymbol{u'}\right)\right) - \boldsymbol{f'}\left(\boldsymbol{x}, t; b\right) = 0 \quad \text{on } \Omega, \tag{2.6}$$

with the corresponding sensitivity BCs

$$\boldsymbol{\mathcal{B}}_b \left(\boldsymbol{u}, L\left(\boldsymbol{u'}\right)\right) = \boldsymbol{g}_b \left(\boldsymbol{x}, t; b\right) \quad \text{on } \Gamma, \tag{2.7}$$

where $\boldsymbol{g}_b$ is the right side of Eq. (2.5). The similarity of Eqs. (2.6) and (2.7) to Eqs. (2.1) and (2.2) motivates the same solution method for each set of equations with the same mesh for the discretized form. For linear governing Eqs. (2.1), $\boldsymbol{\mathcal{A}}_b = \boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}_b = \boldsymbol{\mathcal{B}}$. For nonlinear governing equations, the solution $\boldsymbol{u}$ is obtained from the analysis solution of Eq. (2.1) for use in Eqs. (2.6) and (2.7). Similar to the flow analysis, equations the time term t, suppressed in these equations, represents pseudo-time, since the current work involves steady-state sensitivity analysis.

In the direct formulation of CSA, Eqs. (2.6) and (2.7) are solved to obtain the local derivatives $\boldsymbol{u'}$ in the domain Ω. This may be followed by adding the convective term, as shown in (2.3), at the locations of interest in the domain, to obtain the total derivative of the response variable $\dot{\boldsymbol{u}}$ at those locations. In typical aerodynamic shape optimization applications, it is of interest to obtain the derivative of a performance metric such as

$$\psi = \int_{\bar{\Gamma}} \sigma \boldsymbol{u} \, d\bar{\Gamma} \tag{2.8}$$

which is based on weighted surface integral of the response $\boldsymbol{u}$. Here σ are the weights and $\bar{\Gamma} \in \Gamma$. Assuming that the weights do not depend on the shape design variable, the total

derivative of the performance measure is obtained using the values of $\dot{\boldsymbol{u}}$ as

$$\dot{\psi} = \frac{D\psi}{Db} = \int_{\bar{\Gamma}} \sigma \dot{\boldsymbol{u}} \, d\bar{\Gamma} + \int_{\bar{\Gamma}} \sigma u \, \dot{d\bar{\Gamma}}, \tag{2.9}$$

where $\dot{d\bar{\Gamma}}$ denotes the infinitesimal change in the boundary $\bar{\Gamma}$ due to the shape design variable.

2.2.3 Discretization of the Flow and Sensitivity Equations

Until this point, the continuous governing equations were differentiated to obtain the CSEs. Thus, there is no approximation involved in deriving the CSE system (2.6–2.7). Next, consider a discretization at mesh level h of the flow equations (2.1) and a Newton-Raphson implicit scheme that results in a coupled linear system of equations

$$\left[T\left(\{\boldsymbol{u}\}_h^n\right)\right]\{\Delta \boldsymbol{u}\}_h^n = \{\boldsymbol{R}\}_h^n, \tag{2.10}$$

which has to be solved at each iteration n for the updates to the flow variables $\{\Delta \boldsymbol{u}\}_h^n$. This update is used to get the values of flow variables at the next iteration $\{\boldsymbol{u}\}_h^{n+1} = \{\boldsymbol{u}\}_h^n + \{\Delta \boldsymbol{u}\}_h^n$. Here $\left[T\left(\{\boldsymbol{u}\}_h^n\right)\right]$ is the tangent matrix and $\{\boldsymbol{R}\}_h^n$is the residual of the flow equations at time step n and discretization h. Similarly, the CSEs can be discretized to get a coupled linear system of equations

$$\left[T_{\text{CSE}}\left(\{\boldsymbol{u}\}_h\right)\right]\{\Delta \boldsymbol{u'}\}_h \;=\; \{\boldsymbol{R}_{\text{CSE}}\}_h. \tag{2.11}$$

As a consequence of the CSEs being linear in the sensitivity variables, the tangent matrix $\left[T_{\text{CSE}}\left(\{\boldsymbol{u}\}_h\right)\right]$ in the discretized CSEs (2.11) is independent of the sensitivity variables $\boldsymbol{u'}$ and only depends on the flow variables $\boldsymbol{u}$. Borggaard and Burns (Borggaard and Burns, 1994, 1997), Wickert (Wickert et al., 2010) and Liu and Canfield (Liu and Canfield, 2013b) showed that if the same discretization used for the analysis is used to discretize the CSEs,

then

$$[T_{\mathrm{CSE}}\left(\{\boldsymbol{u}\}_h\right)] = \left[T\left(\{\boldsymbol{u}\}_h^N\right)\right],$$

where N is the last iteration step of the flow solver once steady-state convergence is achieved. Also, since the CSEs are linear in the sensitivity variables, the local shape derivatives are obtained by just a single (one-shot) solution of the linear system (2.11) for $\{\boldsymbol{u'}\}_h$, assuming $\{\boldsymbol{u'}\}_h = \{\Delta\boldsymbol{u'}\}_h$ with zero initial guess $\left(\{\boldsymbol{u'}\}_h^0 = 0\right)$ without the loss of generality. Once the local derivatives are computed by solving (2.11), the total derivatives can be obtained by adding the convective term according to the discrete form of (2.3) and the derivative of the performance measure can be obtained by the discrete form of (2.9).

2.2.4 On Accurate Boundary Conditions for CSA

Although the boundary conditions (2.5) do not appear explicitly in the discretized equations (2.11), they are included in the residual vector $\{\boldsymbol{R}_{\mathrm{CSE}}\}_h$. In fact, the solution of the local shape design derivatives $\{\boldsymbol{u'}\}$ are mainly driven by CSE boundary conditions. Hence, it is important to accurately evaluate these while forming the CSE residual vector $\{\boldsymbol{R}_{\mathrm{CSE}}\}_h$. Duvigneau and Pelletier (2006) have emphasized on the effect of CSE boundary conditions on the accuracy of the local shape design derivatives. Taking a close look at Eq. (2.5), it can be seen that once the design variables are defined, all terms on the right side are known except for $\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u}$ in the convective term, which involves spatial gradients of the flow variables. Improvising on the l-patch method by Duvigneau and Pelletier, which itself was based on the super-convergent patch recovery method by Zienkiewicz and Zhu (1992), Cross and Canfield (2014) proposed the Spatial Gradient Reconstruction (SGR) method to approximate the spatial gradient term appearing in the CSE boundary condition (2.5) for structures. SGR is used in the current research to approximate the spatial gradients $\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u}$ appearing in the CSE boundary conditions. Since SGR does not require information of the discretization method used, such as finite element or finite volume or finite difference, it can be used as a post-

processing step, following the analysis solution, to calculate the required spatial gradients. This makes CSA amenable for nonintrusive implementation.

An SGR patch constitutes a collection of points which are used to get the spatial gradients at an expansion point. Examples of SGR patches are illustrated in Figure 2.2. A Taylor series approximation is used to approximate the spatial gradients of the quantity of interest, such as velocity, with a least-squares approach. As an example, for a 4-layer, third-order SGR, the following third-order Taylor series expansion is used to solve the least-squares problem.

$$\begin{aligned} \phi(x+\Delta x,\, y+\Delta y) &= \phi(x,\, y,\, z) + \phi_{,x}\Delta x + \phi_{,y}\Delta y \\ &+ \frac{1}{2}\phi_{,xx}\left(\Delta x\right)^2 + \frac{1}{2}\phi_{,yy}\left(\Delta y\right)^2 \\ &+ \phi_{,xy}\Delta x\Delta y + \phi_{,yz}\Delta y\Delta z, \end{aligned} \tag{2.12}$$

where ϕ could be a scalar such as a component of velocity or flux and $\phi_{,x}$, $\phi_{,y}$, $\phi_{,z}$ are the spatial gradients that are computed. Accuracy of the spatial gradients obtained from SGR can be improved by modifying parameters such as the patch-size and the order of Taylor series. Cross and Canfield (2016) discuss the numerical behavior, accuracy, and convergence of design derivatives when SGR is used to approximate the CSE boundary conditions and they also give guidelines in selecting SGR parameters to achieve better results.

2.2.5 Nonintrusive Implementation

The procedure for nonintrusive implementation of the direct formulation of CSA is illustrated by the flowchart given in Figure 2.3 and is explained next.

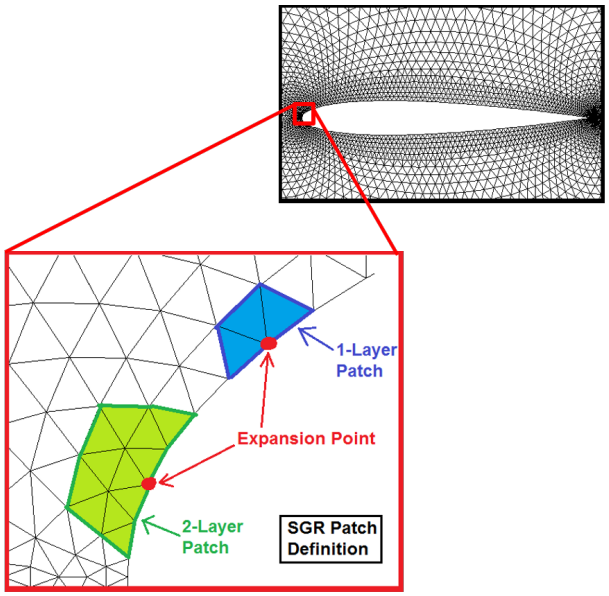

Figure 2.2: Illustration of SGR patches.

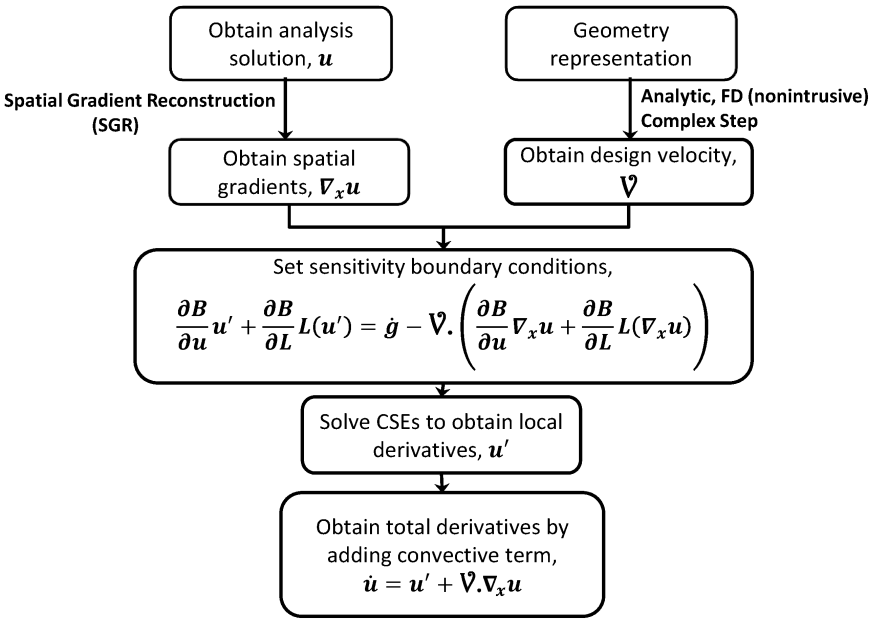

Figure 2.3: Flowchart for direct formulation of Continuum Sensitivity Analysis

First, the flow analysis solution $\boldsymbol{u}$ is obtained by running the analysis code (for example SU2 or Fluent). Then, the SGR method is employed to approximate the spatial gradients $\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u}$ that appear in the sensitivity essential BCs. This is shown in the left branch of the flowchart. Another term required for assembling the CSE boundary conditions (2.5) is the design velocity $\boldsymbol{\mathcal{V}}(\boldsymbol{x})$. This term can be obtained nonintrusively by using either the analytic equations of the parametric geometry, or by using finite difference approximation. Alternatively, the design velocity could also be calculated using a complex step approximation (Kulkarni et al., 2014b). This is shown in the right branch of the flowchart. With this, the CSEs boundary conditions (2.5) are defined. Next, the linear CSEs (2.11) have to be solved. This can be done using either of the following two approaches:

1. Solve the CSEs using the same code that solves the flow equations. In this case the flow code should be able to accommodate non-homogeneous boundary conditions such as (2.5). Also, the flow code should use the steady-state tangent matrix $\left[T\left(\{\boldsymbol{u}\}_h^N\right)\right]$ to solve the CSEs. Since the CSEs are linear, there is no need to iterate. The flow solver can be stopped after the first iteration, when it is used to solve the CSEs. Although not necessary, pseudo-time step iterations can also be used to achieve steady-state convergence.

2. Solve the CSEs externally. In this case, the tangent matrix $\left[T\left(\{\boldsymbol{u}\}_h^N\right)\right]$ has to be exported or regenerated and the right side of the CSEs used to solve the CSEs external to the flow solver.

Solution of CSEs results in the local derivatives $\boldsymbol{u'}$. In the final step, the convective term is added to these local derivatives to obtain total or material derivatives $\dot{\boldsymbol{u}}$. The next sub-section explains the requirements of a flow analysis code so that it can be used for nonintrusive implementation of CSA, based on the procedure mentioned above.

2.2.6 Requirements of a Flow Analysis Code for Nonintrusive CSA

The advantages of CSA over other design sensitivity analysis methods, such as finite difference, complex step approximation, discrete analytic method, and semi-analytic method, are that it can yield more accurate derivatives, it does not suffer from requirement to choose a step size, and that it does not require calculation of mesh sensitivity. The latter advantage is mentioned particularly by Borggaard and Burns (1997). Cross and Canfield (2012) have stated the requirements of an analysis code for it to be used as a "black-box" or nonintrusive tool for CSA of a structural system. Extending these, the following can be stated as the requirements of a flow solver so that it can be used for nonintrusive application of CSA.

1. Output response $\boldsymbol{u}$: The solver should output the state variables such as fluid pressure, velocities and temperature. This corresponds to the step of obtaining "Analysis Solution" in the flowchart of Figure 2.3. Although this seems like a trivial requirement, it is stated here because some solvers may have option of only plotting the flow variables and not accessing them, which may not be enough, since the flow variables have to be post-processed to obtain the spatial gradients $\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u}$. As explained in the previous sub-section, these spatial gradients are required on the boundary with nonzero design velocity to form the CSE boundary conditions, and in the final step to calculate the convective term $(\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u} \cdot \boldsymbol{\mathcal{V}})$ at the locations where the total derivative $\dot{\boldsymbol{u}}$ are needed.

2. For solving CSEs internally (using the same flow analysis code) there are the following two requirements. First, since the boundary conditions for CSEs are typically non-homogeneous unlike the homogeneous boundary conditions for the original analysis, to solve the CSEs using the same analysis code, it is required that the code be able to handle the non-homogeneous boundary conditions (2.5). Second, the flow solver should be able to regenerate the tangent matrix $\left[T\left(\{u\}_h^N\right)\right]$ from the steady-state solution and further use it to solve the CSEs.

3. For solving the CSEs externally, the flow solver should be able to output the tangent matrix $\left[T\left(\{u\}_h^N\right)\right]$. For CFD, the tangent matrix is the full flow Jacobian matrix. In implicit CFD solvers, for steady-state solutions, a psedo-time term may be added to the elements on the diagonal of the Jacobian matrix for making it diagonally dominant. This time term should be removed before using the Jacobian matrix for solving CSEs.

If all these requirements are satisfied, a nonintrusive implementation of CSA is possible using the flow solver as a "black-box" tool. The results of a survey of some flow solvers in context of nonintrusive CSA application is shown in Table 2.2. Columns list the different flow solvers while rows list the requirements that they should satisfy for their use as non-intrusive CSA solvers. A check mark '✓' indicates that the capability exists, a cross '×' indicates that the capability does not exist while the sign '(?)' indicates that enough literature is not available to find out the about that specific requirement. Based on this survey, it was decided to use SU2 for nonintrusive implementation of compressible viscous (Euler equation) flow whereas the in-house code to be used for nonintrusive implementation of incompressible viscous flow.

2.3 Sensitivity of Flow in a Lid-driven Cavity

2.3.1 Flow Analysis

The benchmark lid-driven cavity problem (Roy and Hopkins, 2003) consists of obtaining the fluid velocities and pressure in a rectangular cavity filled with fluid, which is L_x units wide and L_y units high, when the lid of the cavity is pulled with constant velocity u_{Lid}. The steady-state flow in the cavity when the lid is pulled with the constant velocity $u_{\text{Lid}} = 1\,m/s$, is shown in Figures 2.4–2.6. Origin of the Cartesian coordinate system for the cavity is assumed to be at the left bottom corner.

Table 2.2: Survey of CFD solvers for nonintrusive CSA.

CFD Codes (→) Requirements (↓)	ZAERO	ZEUS	Incompressible N-S Solver (In-house code)	Stanford University Unstructured (SU^2)	OpenFOAM	Fun3D
Output response	√	√	√	√	√	√
Apply non-homogenous BC	X	(?)	√	(?)	(?)	(?)
Output tangent matrix	X	(?)	√	√	(?)	(?)

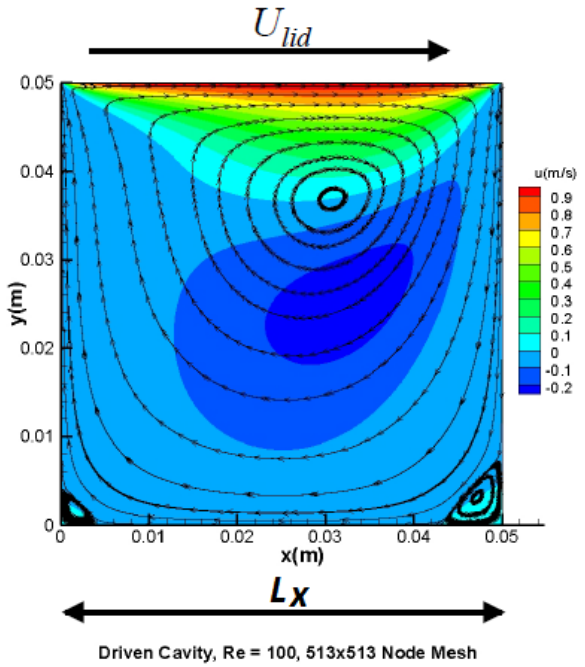

Figure 2.4: Streamlines and horizontal velocities in the lid-driven cavity

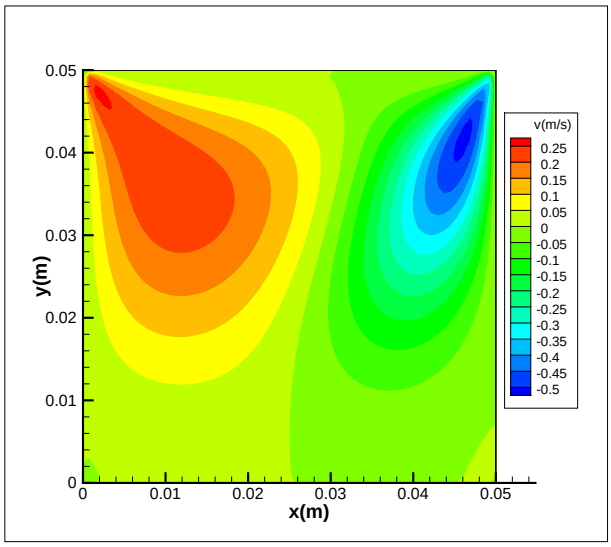

Figure 2.5: Vertical velocities in the lid-driven cavity

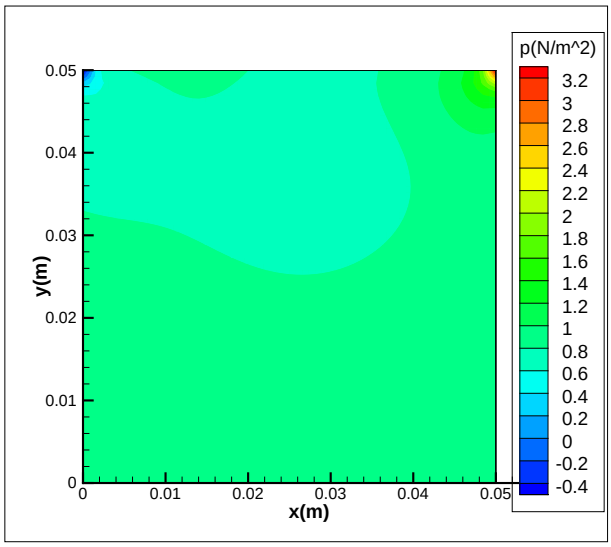

Figure 2.6: Pressure in the lid-driven cavity

This problem was solved using the following incompressible Navier-Stokes equations,

$$\frac{1}{\beta^2}\frac{\partial p}{\partial t} + \rho\frac{\partial u}{\partial x} + \rho\frac{\partial v}{\partial y} = S + f_m, \tag{2.13}$$

$$\rho\frac{\partial u}{\partial t} + \rho u\frac{\partial u}{\partial x} + \rho v\frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} - \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = f_x, \tag{2.14}$$

$$\rho\frac{\partial v}{\partial t} + \rho u\frac{\partial v}{\partial x} + \rho v\frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} - \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) = f_y, \tag{2.15}$$

where p is the pressure, ρ is the density, u and v are the velocities in the horizontal and vertical directions, S represents the artificial viscosity, μ is the dynamic viscosity, and the source terms f_m, f_x, and f_y are zero. Since the steady-state flow solution is of interest, the variable t represents pseudo-time and the associated time derivative terms with a preconditioning factor β are used to march the solution to the steady-state. The boundary conditions for this problem are the no-slip Dirichlet boundary conditions as follows.

- Velocities at the left, right and bottom wall are zero.

$$u\left(0,y\right)=0,\ v\left(0,y\right)=0 \tag{2.16}$$

$$u\left(L_x,y\right)=0,\ v\left(L_x,y\right)=0 \tag{2.17}$$

$$u\left(x,0\right)=0,\ v\left(x,0\right)=0 \tag{2.18}$$

- Vertical velocity on the top wall is zero, horizontal velocity at the top wall is the lid-driving velocity,

$$u\left(x,L_y\right)=u_{\text{Lid}},\ \ v\left(x,L_y\right)=0. \tag{2.19}$$

The boundary conditions are thus homogeneous. No conditions are imposed on the pressure at the domain boundary. Hence pressure at the four walls is extrapolated based on interior values. The time derivative preconditioning factor $(1/\beta^2)$ is used to improve the condition number of the system of equations. The artificial viscosity S is necessary to avoid odd-even coupling when central difference scheme is used for spatial discretization. Eqs. (2.13–2.15) were spatially discretized using a second-order accurate finite difference scheme. The equations were solved with a Symmetric Gauss Seidel (SGS) iterative scheme to obtain a steady-state solution using MATLAB.

Since the exact solution for the lid-driven cavity problem is not available, the following manufactured solution with sinusoidal functions was used for verification of the CSA approach, and is plotted in Figures 2.7–2.9. The source terms for the manufactured solution are calculated by substituting the manufactured solution in the Navier-Stokes equations.

$$p_{\text{mms}}\left(x,y\right) = 0.25+0.50\cos\left(\frac{1}{2}\frac{\pi x}{L_x}\right)+0.40\sin\left(\frac{1}{5}\frac{\pi y}{L_y}\right)+0.33\sin\left(\frac{2}{7}\frac{\pi xy}{L_xL_y}\right) \tag{2.20}$$

$$u_{\text{mms}}\left(x,y\right) = 0.30+0.15\sin\left(\frac{1}{3}\frac{\pi x}{L_x}\right)+0.20\cos\left(\frac{1}{4}\frac{\pi y}{L_y}\right)+0.25\sin\left(\frac{2}{5}\frac{\pi xy}{L_xL_y}\right) \tag{2.21}$$

$$v_{\text{mms}}\left(x,y\right) = 0.20+0.17\cos\left(\frac{7}{17}\frac{\pi x}{L_x}\right)+0.25\cos\left(\frac{1}{6}\frac{\pi y}{L_y}\right)+0.10\cos\left(\frac{1}{3}\frac{\pi xy}{L_xL_y}\right) \tag{2.22}$$

By substituting appropriate boundary values of x and y in Eqs. (2.20–2.22), the following Dirichlet boundary conditions for the manufactured solution are derived.

$$u\left(0,y\right)=u_{\text{mms}}(0,y), \qquad v\left(0,y\right)=v_{\text{mms}}\left(0,y\right), \tag{2.23}$$

$$u\left(L_x,y\right)=u_{\text{mms}}\left(L_x,y\right), \qquad v\left(L_x,y\right)=v_{\text{mms}}\left(L_x,y\right), \tag{2.24}$$

$$u\left(x,0\right)=u_{\text{mms}}\left(x,0\right), \qquad v\left(x,0\right)=v_{\text{mms}}\left(x,0\right), \tag{2.25}$$

$$u\left(x,L_y\right)=u_{\text{mms}}\left(x,L_y\right), \qquad v\left(x,L_y\right)=v_{\text{mms}}\left(x,L_y\right). \tag{2.26}$$

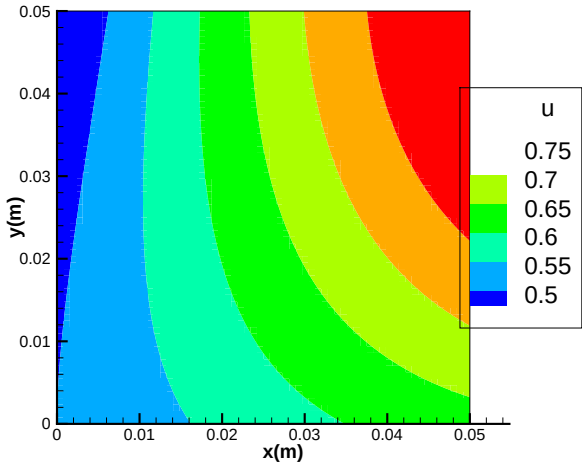

Figure 2.7: Manufactured solution for horizontal velocity.

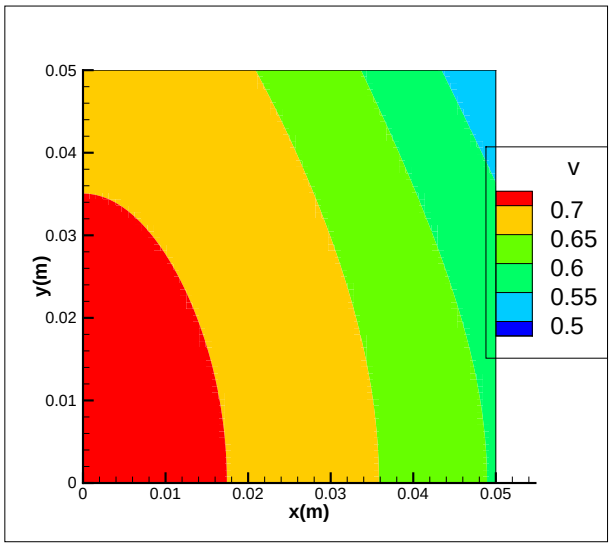

Figure 2.8: Manufactured solution for vertical velocity.

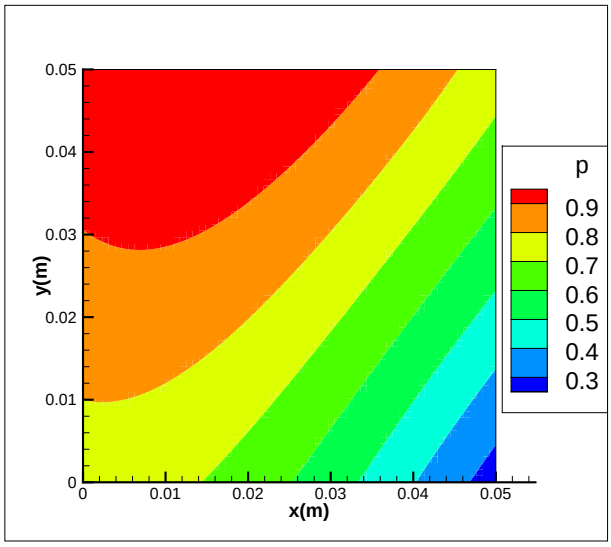

Figure 2.9: Manufactured solution for pressure.

2.3.2 Sensitivity Analysis

In context of sensitivity analysis, the problem is to obtain the derivatives of flow velocities and pressures at all locations inside the cavity, with respect to width of the cavity L_x, which is a shape variable. The CSEs are derived by differentiating the Eqs. (2.13–2.15) with respect

to the shape design variable L_x.

$$\frac{1}{\beta^2}\frac{\partial p'}{\partial t}+\rho\frac{\partial u'}{\partial x}+\rho\frac{\partial v'}{\partial y}-S=f'_m \tag{2.27}$$

$$\rho\frac{\partial u'}{\partial t}+\rho u'\frac{\partial u}{\partial x}+\rho u\frac{\partial u'}{\partial x}+\rho v'\frac{\partial u}{\partial y}+\rho v\frac{\partial u'}{\partial y}+\frac{\partial p'}{\partial x}-\mu\left(\frac{\partial^2 u'}{\partial x^2}+\frac{\partial^2 u'}{\partial y^2}\right)=f'_x \tag{2.28}$$

$$\rho\frac{\partial v'}{\partial t}+\rho u'\frac{\partial v}{\partial x}+\rho u\frac{\partial v'}{\partial x}+\rho v'\frac{\partial v}{\partial y}+\rho v\frac{\partial v'}{\partial y}+\frac{\partial p'}{\partial y}-\mu\left(\frac{\partial^2 v'}{\partial x^2}+\frac{\partial^2 v'}{\partial y^2}\right)=f'_y \tag{2.29}$$

Although the governing equations are nonlinear with respect to response variable p, u, v, the CSEs (2.27–2.29) are linear with respect to sensitivity variables p', u', v'. The CSE boundary conditions are obtained by material differentiation of the flow boundary conditions.

$$u'(0,y) = 0-\left[\mathcal{V}_x\frac{\partial u}{\partial x}+\mathcal{V}_y\frac{\partial v}{\partial y}\right]\Bigg|_{(0,y)},\ v'(0,y)=0-\left[\mathcal{V}_x\frac{\partial u}{\partial x}+\mathcal{V}_y\frac{\partial v}{\partial y}\right]\Bigg|_{(0,y)}, \tag{2.30}$$

$$u'(L_x,y) = 0-\left[\mathcal{V}_x\frac{\partial u}{\partial x}+\mathcal{V}_y\frac{\partial v}{\partial y}\right]\Bigg|_{(L_x,y)},\ v'(L_x,y)=0-\left[\mathcal{V}_x\frac{\partial u}{\partial x}+\mathcal{V}_y\frac{\partial v}{\partial y}\right]\Bigg|_{(L_x,y)}, \tag{2.31}$$

$$u'(x,0) = 0-\left[\mathcal{V}_x\frac{\partial u}{\partial x}+\mathcal{V}_y\frac{\partial v}{\partial y}\right]\Bigg|_{(x,0)},\ v'(x,0)=0-\left[\mathcal{V}_x\frac{\partial u}{\partial x}+\mathcal{V}_y\frac{\partial v}{\partial y}\right]\Bigg|_{(x,0)}, \tag{2.32}$$

$$u'(x,L_y) = 0-\left[\mathcal{V}_x\frac{\partial u}{\partial x}+\mathcal{V}_y\frac{\partial v}{\partial y}\right]\Bigg|_{(x,L_y)},\ v'(x,L_y)=0-\left[\mathcal{V}_x\frac{\partial u}{\partial x}+\mathcal{V}_y\frac{\partial v}{\partial y}\right]\Bigg|_{(x,L_y)}, \tag{2.33}$$

where $\boldsymbol{\mathcal{V}}(x,y)=\{\mathcal{V}_x(x,y),\mathcal{V}_y(x,y)\}^T$ is the design velocity. Since the design variable is L_x, the design velocity can be derived as

$$\mathcal{V}_x(x,y)=\frac{x}{L_x},\ \mathcal{V}_y(x,y)=0. \tag{2.34}$$

There are two significant issues worth noting.

1. The sensitivity boundary conditions (2.30–2.33) are non-homogeneous.

2. To compute the the boundary conditions spatial gradients and design velocities are required to be calculated at the boundaries.

As explained in Section 2.2.4, the spatial gradients of velocities appearing in Eqs. (2.30–2.33) are calculated using SGR. The advantage of using a manufactured solution is that the exact spatial gradients and derivatives are available. Thus, the true errors in the derivatives obtained from CSA can be computed to assess accuracy and rate of convergence. In theory, if the spatial gradients are accurately known and the linear CSEs are accurately solved, the resulting local derivatives will be accurate. However, this is seldom realized in practice because discretization error is introduced during solution of the differential equations. Since the exact spatial derivatives are known when using a manufactured solution, the best case scenario for CSA would be when the CSE boundary conditions are calculated using the exact spatial gradients. This is used as a limiting test case as shown in the following section.

2.3.3 Flow Derivatives

For the particular choice of manufactured solution (2.20–2.22), the total derivatives $\dot{\boldsymbol{u}} = \{\dot{u}, \dot{v}, \dot{p}\}$ are zero; however, the local derivatives and the spatial derivatives are nonzero. The manufactured solution for the local derivatives is shown in Figures 2.10–2.12. The CSA local flow derivatives were compared to the exact local derivative solutions to calculate the true discretization error

$$\epsilon_\phi = \phi_h - \bar{\phi},$$

where ϕ is a local derivatives such as u', v', p' and $\bar{\phi}$ is the corresponding exact MMS local derivative. The discretization errors in the flow local derivatives are shown in Figures 2.13–2.15. Notice that the discretization errors are at least four orders of magnitude smaller than the corresponding local derivative values.

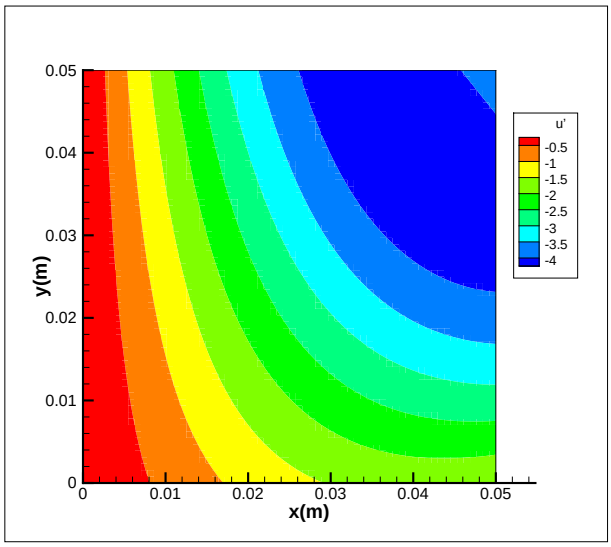

Figure 2.10: Manufactured solution for local derivative of horizontal velocity with respect to width of the cavity.

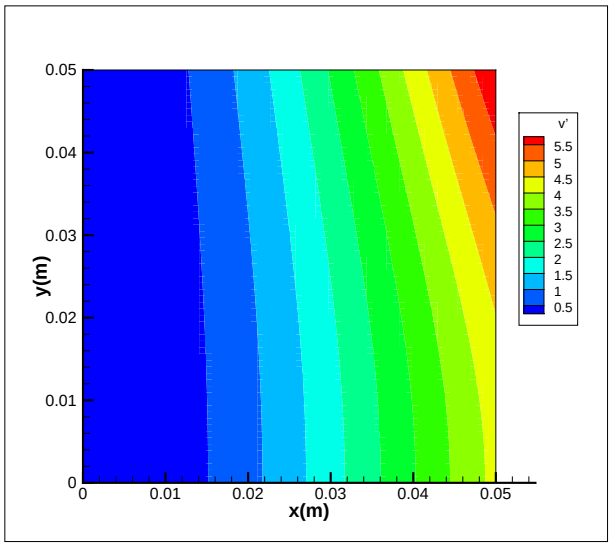

Figure 2.11: Manufactured solution for local derivative of vertical velocity with respect to width of the cavity.

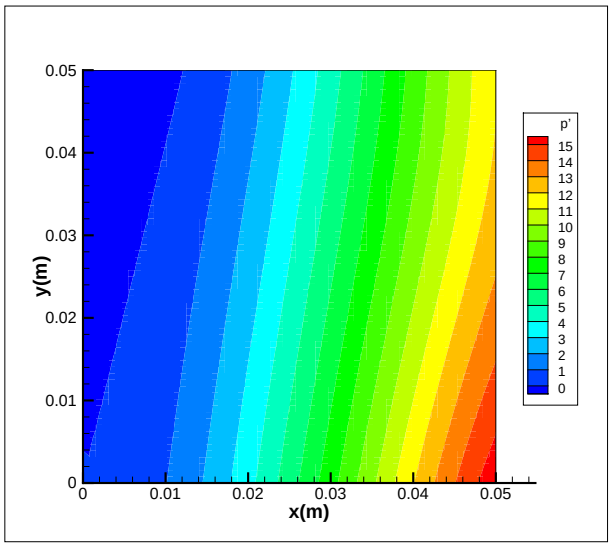

Figure 2.12: Manufactured solution for local derivative of pressure with respect to width of the cavity.

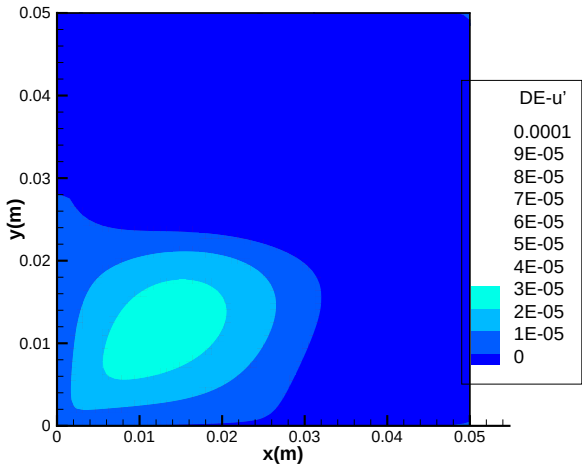

Figure 2.13: Discretization error (DE) in the local derivative of horizontal velocity, $\epsilon_{u'}$, for the case of 9 layer SGR with fifth-order Taylor series, plotted for the finest mesh with 65×65 grid points.

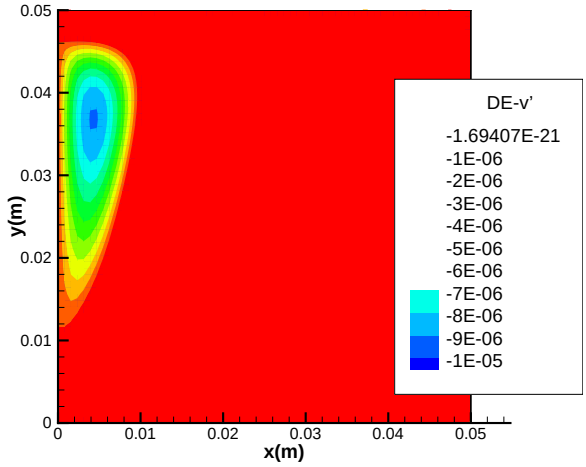

Figure 2.14: Discretization error (DE) in the local derivative of vertical velocity, $\epsilon_{v'}$, for the case of 9 layer SGR with fifth-order Taylor series, plotted for the finest mesh with 65×65 grid points.

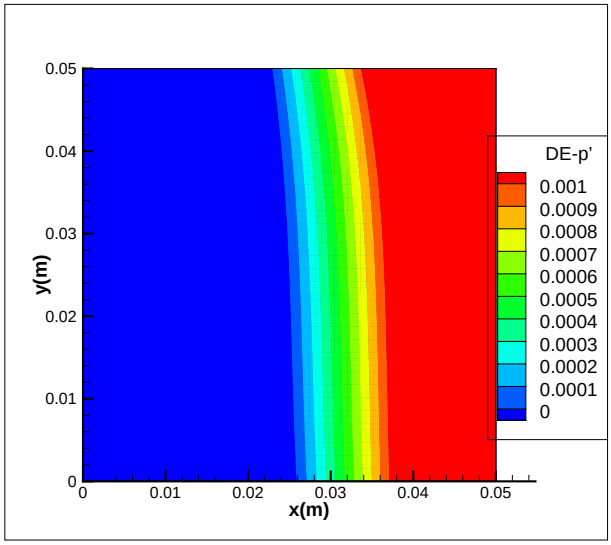

Figure 2.15: Discretization error (DE) in the local derivative of vertical velocity, $\epsilon_{p'}$, for the case of 9 layer SGR with fifth-order Taylor series, plotted for the finest mesh with 65×65 grid points.

The effect of using different number of SGR patches and order of Taylor series on the CSA derivatives was studied. A grid convergence study was done for code verification using the manufactured solution which is explained next.

2.3.4 Grid Convergence Study

Four mesh sizes were used for the grid convergence study, from a size of 17×17 nodes (mesh refinement parameter $h = 4$) to 65×65 nodes (mesh refinement parameter $h = 1$). The discretization error L_∞ norm of the discretization error in the local derivative of pressure, $||\epsilon_{p'}||_\infty$, are shown in the left part of Figure 2.16. The corresponding rate of convergence is shown in the right part of Figure 2.16. A second-order accurate finite difference scheme was used in the flow analysis. It can be seen that the second-order rate of convergence is achieved for the local derivatives as well. Typically the rate of convergence of shape derivatives obtained using a finite difference scheme would be expected to be one order less than that of the analysis variables. Hence, the finding that CSA derivative results are of the same rate of convergence as the flow variables, is a key contribution in this work.

The number of patch layers and order of Taylor series used in SGR, for approximating the CSE boundary conditions are labelled by the letters L and O in Figure 2.16, respectively. The line which is labelled "Analytic Gradients" corresponds to the limiting case when analytic spatial gradients (known from MMS) are used to construct the sensitivity BCs. Firstly, the discretization error in the local derivatives found using SGR is almost the same as the discretization error in the limiting case, which uses analytic spatial gradients. Next, the results with one layer, first-order Taylor series SGR are achieved to be first-order accurate. However, looking at the results with other layer and order combinations, we concluded that even a two-layer, second-order Taylor series SGR is enough to recover the maximum possible second-order rate of convergence. This highlights the advantages of the CSA formulation with SGR to construct the spatial gradients. The lines corresponding to the label "1+O" in in Figure 2.16 indicate that the Taylor series order may be higher than 1 at some locations in the domain. Similarly, "2+" indicates that the Taylor series order may be higher than 2 at some locations for those cases.

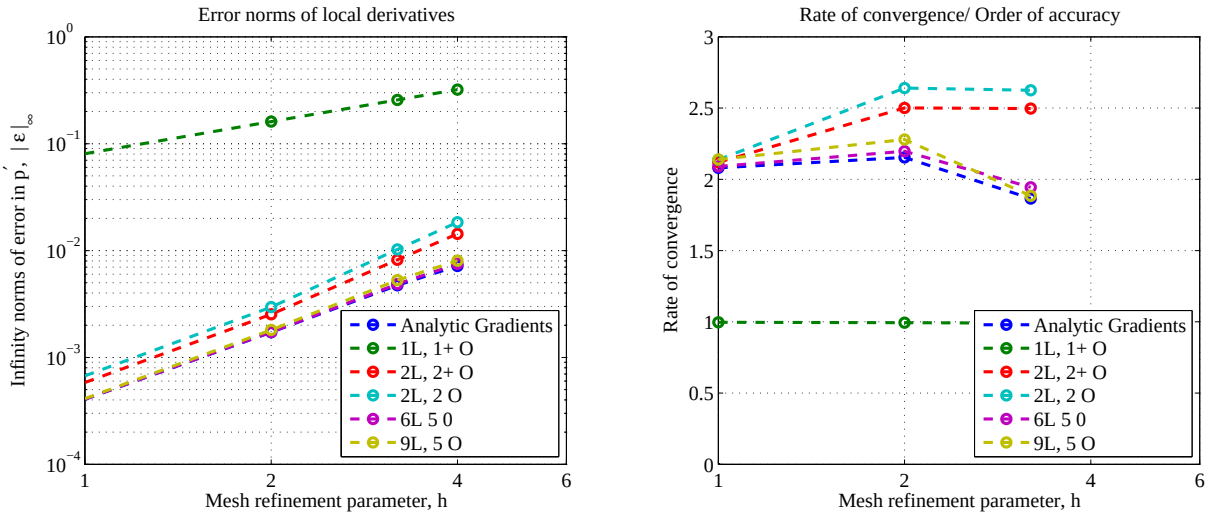

Figure 2.16: Grid convergence study for the local derivatives of lid-driven cavity flow using a manufactured solution: Discretization error norms and rate of convergence are plotted against mesh refinement parameter h; Finest mesh is 65 X 65 nodes ($h = 1$) and coarsest mesh is 17 X 17 nodes ($h = 4$).

2.4 Sensitivity of Flow Over NACA0012 Airfoil

2.4.1 Flow Analysis

The Euler flow equations in conservation form are

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} - \boldsymbol{H} = 0 \tag{2.35}$$

or

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial y} - \boldsymbol{H} = 0 \tag{2.36}$$

subject to farfield boundary condition

$$\boldsymbol{u}|_{S_F} = \boldsymbol{u}_\infty, \tag{2.37}$$

and flow tangency (or wall) boundary condition

$$(u\hat{\imath} + v\hat{\jmath}) \cdot \hat{\boldsymbol{n}}|_{S_W} = 0, \tag{2.38}$$

where $\boldsymbol{u}(x,\, y,\, t)$ is the vector of conserved variables, $\boldsymbol{F}(x,\, y,\, t)$ and $\boldsymbol{G}(x,\, y,\, t)$ are the flux vectors in the X and Y coordinate directions, $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}$ and $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{u}}$ are the respective flux Jacobian matrices, and $\boldsymbol{H}$ is the source term, which is zero. The state vector and flux vectors are

$$\boldsymbol{u} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e_t \end{Bmatrix}, \quad \boldsymbol{F} = \begin{Bmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ \rho u h_t \end{Bmatrix}, \quad \boldsymbol{G} = \begin{Bmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ \rho v h_t \end{Bmatrix}. \tag{2.39}$$

Variables ρ, p, u, v, $e_t = \left(\frac{1}{\gamma - 1}\frac{p}{\rho} + \frac{(u\hat{\imath}+v\hat{\jmath})^2}{2}\right)$, and $h_t = e_t + \frac{p}{\rho}$ denote density, pressure, horizontal velocity, vertical velocity, total energy, and total enthalpy in the domain, respectively. The pressure and density can be related to the temperature T by the equation of state $p = \rho \mathcal{R} T$, where $\mathcal{R}$ is the specific gas constant. The farfield boundary condition implies that $\boldsymbol{u}_\infty$ is the prescribed state at the farfield boundary S_F. The flow tangency boundary condition implies that the velocity vector $(u\hat{\imath} + v\hat{\jmath})$ has no component along the unit normal $\hat{\boldsymbol{n}}$ on the wall boundary S_W. The farfield and wall boundaries for flow over an airfoil are shown in Figure 2.17.

When a vertex-centered finite volume discretization is used, the system of nonlinear coupled partial differential equations (2.35) is approximated by the following semi-discrete system

$$\Omega_i \frac{\partial \{\boldsymbol{u}(t)\}_i}{\partial t} - \{\boldsymbol{R}\}_i = 0, \tag{2.40}$$

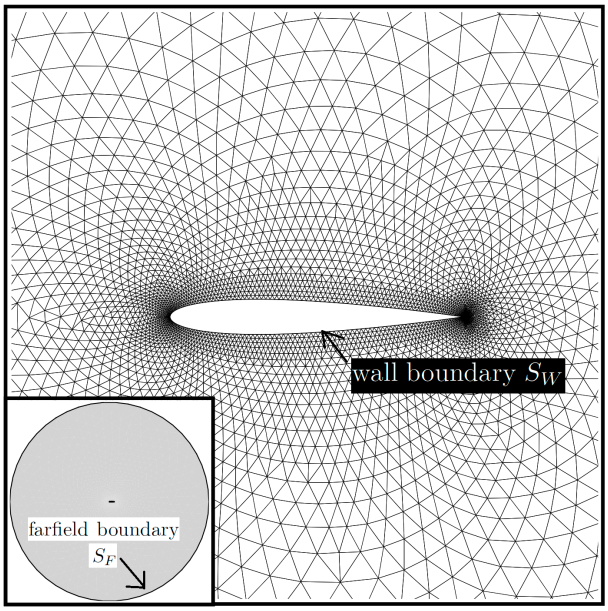

Figure 2.17: Domain, boundaries and unstructured mesh for flow over an airfoil. The inset figure shows the entire flow field with airfoil at the center.

where $\{\boldsymbol{R}\}_i$ is the residual at node i, given by

$$\{\boldsymbol{R}\}_i \equiv -\sum_{j \in \mathcal{N}(i)} \left\{\widetilde{\boldsymbol{F}}\right\}_{ij} \Delta S_{ij}. \tag{2.41}$$

The grid and associated control volume, also sometimes called a dual cell, is shown in Figure 2.18. $\left\{\widetilde{\boldsymbol{F}}\right\}_{ij}$ is the projected flux for the face of the control volume connecting nodes i and j, ΔS_{ij} is the area of that face, Ω_i is the volume of the control volume surrounding node i, and $\mathcal{N}(i)$ consists of all the nodes neighboring node i. The dual cells are constructed by joining the centroids of the triangles in the primary grid. Although these equations can be used for time-accurate solutions, the variable t in the present context represents pseudo-time and is used for the purpose of time marching to reach the steady-state.

Roe's upwind scheme can be used to approximate the projected flux as

$$\left\{\widetilde{\boldsymbol{F}}\right\}_{ij} = \frac{\left(\left\{\vec{\boldsymbol{F}}\right\}_i + \left\{\vec{\boldsymbol{F}}\right\}_j\right)}{2} \cdot \hat{n}_{ij} - \frac{1}{2}\left[P\right]\left|\left[\Lambda\right]\right|\left[P^{-1}\right]\left(\{\boldsymbol{u}\}_i - \{\boldsymbol{u}\}_j\right). \tag{2.42}$$

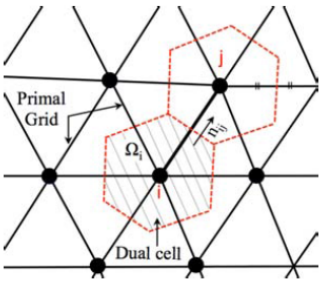

Figure 2.18: Grid and control volume for vertex-centered finite volume scheme (Palacios et al., 2013).

Here $\left\{\vec{\boldsymbol{F}}\right\}_i = \{\boldsymbol{F}\}_i \hat{\imath} + \{\boldsymbol{G}\}_i \hat{\jmath}$ is the flux tensor evaluated at node i, $[P]$ is the matrix of right eigenvectors of the flux Jacobian matrix evaluated at the $i-j$ interface constructed using the Roe-averaged variables $\bar{\rho}$, $\bar{u}$, $\bar{v}$, $\bar{h_t}$, $\bar{a}$ and projected in the $\hat{n}_{ij}$ direction, and $|[\Lambda]|$ is the matrix of modified Roe-averaged eigenvalues.

Using an implicit method to discretize time in Equation (2.40), the residual $\{\boldsymbol{R}\}_i$ is evaluated at the pseudo-time step t^{n+1}. With the Euler implicit scheme the semi-discretized system equations are written as

$$\int \frac{\partial \boldsymbol{u}}{\partial t}\, d\Omega - \{\boldsymbol{R}\}_i \approx |\Omega_i| \frac{d\boldsymbol{u}}{dt} - \{\boldsymbol{R}\}_i = 0$$

which leads to the update equation

$$I \frac{|\Omega_i^n|}{\Delta t_i^n} \{\Delta \boldsymbol{u}\}_i^n = \{\boldsymbol{R}\}_i^{n+1}$$

where $\Delta Q_i^n = Q_i^{n+1} - Q_i^n$ is the update of the state variables to be calculated at each pseudo-time step iteration until steady-state is reached. The residuals at time t^{n+1} are linearized about time t^n.

$$\{\boldsymbol{R}\}_i^{n+1} = \{\boldsymbol{R}\}_i^n + \sum_{j \in \mathcal{N}(i)} \frac{\partial \{\boldsymbol{R}\}_i^n}{\partial \{\boldsymbol{u}\}_j} \{\Delta \boldsymbol{u}\}_i^n + \mathcal{O}\left(t^2\right)$$

With this, the following linear system is solved to get the update $\{\Delta \boldsymbol{u}\}_i^n$:

$$\left(I \frac{|\Omega_i^n|}{\Delta t_i^n} \delta_{ij} \underbrace{- \sum_{j \in \mathcal{N}(i)} \frac{\partial \{\boldsymbol{R}\}_i^n}{\partial \{\boldsymbol{u}\}_j}}_{\left[T\left(\{\boldsymbol{u}\}_h^n\right)\right]} \right) \{\Delta \boldsymbol{u}\}_i^n = \{\boldsymbol{R}\}_i^n \tag{2.43}$$

where I stands for an identity matrix. Compare Eq. (2.43) with the general form of discretized Eq. (2.10). The only addition in Eq. (2.43) is the time term on the diagonal elements, which is a consequence of the pseudo-time stepping approach. In practice, these terms can be eliminated to obtain the Newton-Raphson method for iterating the nonlinear equations to a steady-state.

2.4.2 Sensitivity Analysis

Differentiating the continuous Euler equations (2.35) with respect to the shape design variable b, we get the local continuum sensitivity equations (CSEs), which have to be solved to obtain the local shape derivative of the state vector $\boldsymbol{u}' = \frac{\partial \boldsymbol{u}}{\partial b}$. The CSEs are given by

$$\frac{\partial \boldsymbol{u}'}{\partial t} + \frac{\partial \boldsymbol{F}'}{\partial x} + \frac{\partial \boldsymbol{G}'}{\partial y} = 0, \tag{2.44}$$

where the state vector $Q'(x,\, y,\, t)$ and flux vectors $F'(x,\, y,\, t)$ and $G'(x,\, y,\, t)$ are

$$\boldsymbol{u}' = \begin{Bmatrix} \rho' \\ (\rho u)' \\ (\rho v)' \\ (\rho e_t)' \end{Bmatrix}, \quad \boldsymbol{F}' = \begin{Bmatrix} \rho' u + \rho u' \\ p' + (\rho u)' u + (\rho u) u' \\ (\rho u)' v + (\rho u) v' \\ (\rho h_t)' u + (\rho h_t) u' \end{Bmatrix}, \quad \boldsymbol{G}' = \begin{Bmatrix} \rho' v + \rho v' \\ (\rho u)' v + (\rho u) v' \\ p' + (\rho v)' v + (\rho v) v' \\ (\rho h_t)' v + (\rho h_t) v' \end{Bmatrix}. \tag{2.45}$$

Note that the CSEs are linear in the sensitivity variables.

In the following example of flow over an airfoil, the design variable is chosen to be

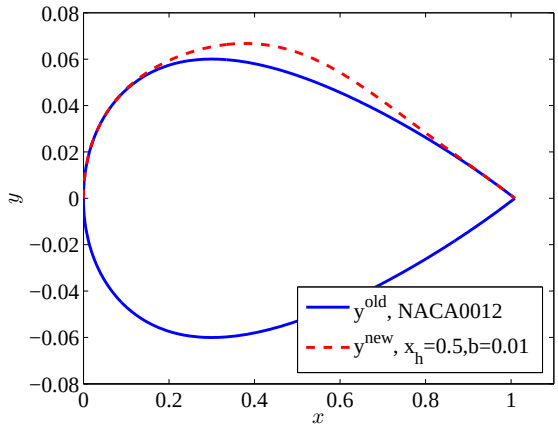

Figure 2.19: Change in shape of NACA 0012 airfoil with design variable b. Note that the axes are scaled to magnify the shape perturbation.

magnitude of the Hicks-Henne bump function (Hicks and Henne, 1978) that perturbs the top surface of the airfoil. Hence the modified top surface of the airfoil for a perturbation Δb is given by

$$y^{\text{top,new}} = y^{\text{top,old}} + \Delta b \sin^3\left(\pi\left(\frac{x}{c}\right)^e\right),\ e = \frac{\log(0.5)}{\log(x_H/c)}, \tag{2.46}$$

where c is the reference chord length of the airfoil and x_H is the location at which the bump is chosen to be located. For the current example $x_H = 0.5$ for the top surface of the airfoil. The old and new airfoil shapes for $\Delta b = 0.01$ are shown in Figure 2.19.

The farfield CSE boundary condition is

$$\boldsymbol{u'}|_{S_F} = 0 - \frac{\partial \boldsymbol{u}}{\partial x}\mathcal{V}_x\bigg|_{S_F} - \frac{\partial \boldsymbol{u}}{\partial y}\mathcal{V}_y\bigg|_{S_F} = 0. \tag{2.47}$$

The design velocity is zero on the farfield boundary S_F because the design variable only changes shape of the airfoil boundary. The CSE boundary condition at the airfoil boundary S_W

$$(u'\hat{\imath} + v'\hat{\jmath}) \cdot \hat{\boldsymbol{n}}|_{S_W} = -\left(\left\{\frac{\partial u}{\partial x}\mathcal{V}_x + \frac{\partial u}{\partial y}\mathcal{V}_y\right\}\hat{\imath} + \left\{\frac{\partial v}{\partial x}\mathcal{V}_x + \frac{\partial v}{\partial y}\mathcal{V}_y\right\}\hat{\jmath}\right) \cdot \hat{\boldsymbol{n}}\bigg|_{S_W} \tag{2.48}$$

$$- \left(u\hat{\imath} + v\hat{\jmath}\right) \cdot \dot{\hat{n}}\Big|_{S_W} ,$$

where $\boldsymbol{\mathcal{V}}(x,y) = \{\mathcal{V}_x(x,y), \mathcal{V}_y(x,y)\}^T$ is the design velocity and $\dot{\hat{n}} = \frac{D\hat{\mathbf{n}}}{Db}$ is the material derivative of the unit normal. This boundary condition is a version of the equation (2.5), obtained by using it for the flow tangency boundary condition (2.38). The CSE boundary condition (2.48) is called as transpiration boundary condition because the dependent flow variable (i.e. local derivatives of the velocity) has a nonzero component along normal direction of the wall boundary S_W. More details of this boundary condition are discussed in Section 2.4.3.

Until this point, no discretization is involved. Thus, the CSEs (2.44) with the corresponding boundary conditions (2.47–2.48) represent continuous equations that govern the local derivatives $\boldsymbol{u'}$ of the flow variables. Derivation of the CSEs completes the "differentiate" part of the CSA approach: "first differentiate, then discretize." The CSEs can be solved numerically using any discretization scheme. However, in the current work we emphasize using the same discretization that was used for the flow analysis. Moreover, this procedure can also be done nonintrusively, if the conditions given in Section 2.2.6 are met. Using the same spatial discretization scheme (vertex based finite volume) and temporal discretization scheme (Euler implicit) as given in Section 2.4.1, we get the following linear system, which is solved to get the update $\{\Delta \boldsymbol{u'}\}_i^n$:

$$\left(I\frac{|\Omega_i^n|}{\Delta t_i^n}\delta_{ij} - \underbrace{\sum_{j\in\mathcal{N}(i)} \frac{\partial\{\boldsymbol{R}_{\mathrm{CSE}}\}_i^n}{\partial\{\boldsymbol{u'}\}_j}}_{\left[T_{\mathrm{CSE}}\left(\{\boldsymbol{u}\}_h\right)\right]} \right) \{\Delta \boldsymbol{u'}\}_i^n = \{\boldsymbol{R}_{\mathrm{CSE}}\}_i^n , \tag{2.49}$$

where

$$\{\boldsymbol{R}_{\text{CSE}}\}_i = \sum_{j\in\mathcal{N}(i)} \left\{\widetilde{\boldsymbol{F}'}\right\}_{ij} \Delta S_{ij}, \tag{2.50}$$

$$\left\{\widetilde{\boldsymbol{F}'}\right\}_{ij} = \frac{\left(\left\{\vec{\boldsymbol{F}'}\right\}_i + \left\{\vec{\boldsymbol{F}'}\right\}_j\right)}{2} \cdot \hat{n}_{ij} - \frac{1}{2}\left[P\right]\left|[\Lambda]\right|\left[P^{-1}\right]\left(\{\boldsymbol{u}'\}_i - \{\boldsymbol{u}'\}_j\right), \tag{2.51}$$

$$\left\{\vec{\boldsymbol{F}'}\right\}_i = \{\boldsymbol{F}'\}_i\,\hat{\imath} + \{\boldsymbol{G}'\}_i\,\hat{\jmath}. \tag{2.52}$$

Note that the matrices $[P]$ and $|[\Lambda]|$ in Equation (2.51) are the same as for flow analysis (2.42). This is because for Roe's flux difference splitting scheme, matrices $[P]$ and $|[\Lambda]|$ are assumed to be constants for the interface at which they are calculated. Furthermore, since the CSEs are linear, the time term $I\frac{|\Omega_i^n|}{\Delta t_i^n}\delta_{ij}$ and the superscript n in (2.49) can be dropped and the right side terms in equations (2.50)–(2.52) are calculated based on an initial guess $\{\boldsymbol{u}'\}_i^0$. Thus, $\boldsymbol{u}'$ can be obtained by solving the following linear system (only once) for each design variable.

$$\left[T_{\text{CSE}}\left(\{\boldsymbol{u}\}_h\right)\right]\{\Delta\boldsymbol{u}'\}_i = \{\boldsymbol{R}_{\text{CSE}}\}_i^0, \tag{2.53}$$

where $\{\boldsymbol{R}_{\text{CSE}}\}_i^0$ is calculated using the state values for the initial guess $\{\boldsymbol{u}'\}_i^0$. Borggaard and Burns (Borggaard and Burns, 1994, 1997), Wickert (Wickert et al., 2010) and Liu and Canfield (Liu and Canfield, 2013b) showed that, if the same discretization used for the analysis is used to discretize the CSEs, then

$$\left[T_{\text{CSE}}\left(\{\boldsymbol{u}\}_h\right)\right] = \left[T\left(\{\boldsymbol{u}\}_h^N\right)\right], \tag{2.54}$$

where N is the last iteration step of the flow solver once steady-state convergence is achieved. Substituting (2.54) in (2.53), we get

$$\left[T\left(\{\boldsymbol{u}\}_h^N\right)\right]\{\Delta\boldsymbol{u}'\}_i = \{\boldsymbol{R}_{\text{CSE}}\}_i^0. \tag{2.55}$$

Comparing Equation (2.55) with Equation (2.43), we observe the following.

- The Jacobian matrix $\left[T\left(\{\boldsymbol{u}\}_h^N\right)\right]$ appearing on the left side of the discrete CSEs (2.55) is same as that for the (converged) steady-state primary flow, i.e., the Jacobian at the last pseudo-time step of the primary implicit flow solver.

- Taking advantage of the linearity of the CSEs, after discretization the local sensitivities can be obtained by just a single (one-shot) solution of Equation (2.55). Hence the time term appearing in the pseudo-time stepping procedure of Equation (2.43) has been omitted in Equation (2.55).

- The right side of Equation (2.55) is the only term that has to be calculated for solving the linear system, if the Jacobian matrix $\left[T\left(\{\boldsymbol{u}\}_h^N\right)\right]$ at the steady-state flow solution can be output (or stored and reused).

The proposed nonintrusive process can be followed in the two cases, as shown in Figures 2.20 and 2.21, (a) when the Jacobian matrix can be output from the flow solver such as SU2, or (b) when the Jacobian matrix cannot be output from the flow solver such as Fluent. In case (b), SU2 is used as a tool to output the Jacobian matrix. Hence in this case, the Jacobian matrix used for CSA is not consistent with the primary analysis (which is done in Fluent), but is an approximation to the consistent Jacobian matrix. Also, since SU2 is a vertex centered code and Fluent is a cell centered code, Fluent cell centered data has to be approximated at the vertices so that it can be used in SU2 to output the Jacobian. This approximation was done using Fluent. An alternative can be to use SGR for this approximation.

The sensitivity analysis starts by assuming an initial guess for the local derivatives as $\{\boldsymbol{u'}\}_i^0 = 0$ at all nodes i. This is followed by calculating the right side $\{\boldsymbol{R}_{\text{CSE}}\}_i^0$ according to Eq. (2.50–2.52) and solving the linear system (2.55) to obtain the local derivatives as $\{\boldsymbol{u'}\}_i = \{\boldsymbol{u'}\}_i^0 + \{\Delta\boldsymbol{u'}\}_i = \{\Delta\boldsymbol{u'}\}_i$ at nodes i. Since the initial guess is zero and the sensitivity equations are linear, $\{\boldsymbol{R}_{\text{CSE}}\}_i^0$ will be zero at all nodes i which are not on the domain boundary. Clearly the local derivatives are driven by the boundary conditions (2.47–2.48). The following subsection explains how these boundary conditions can be implemented

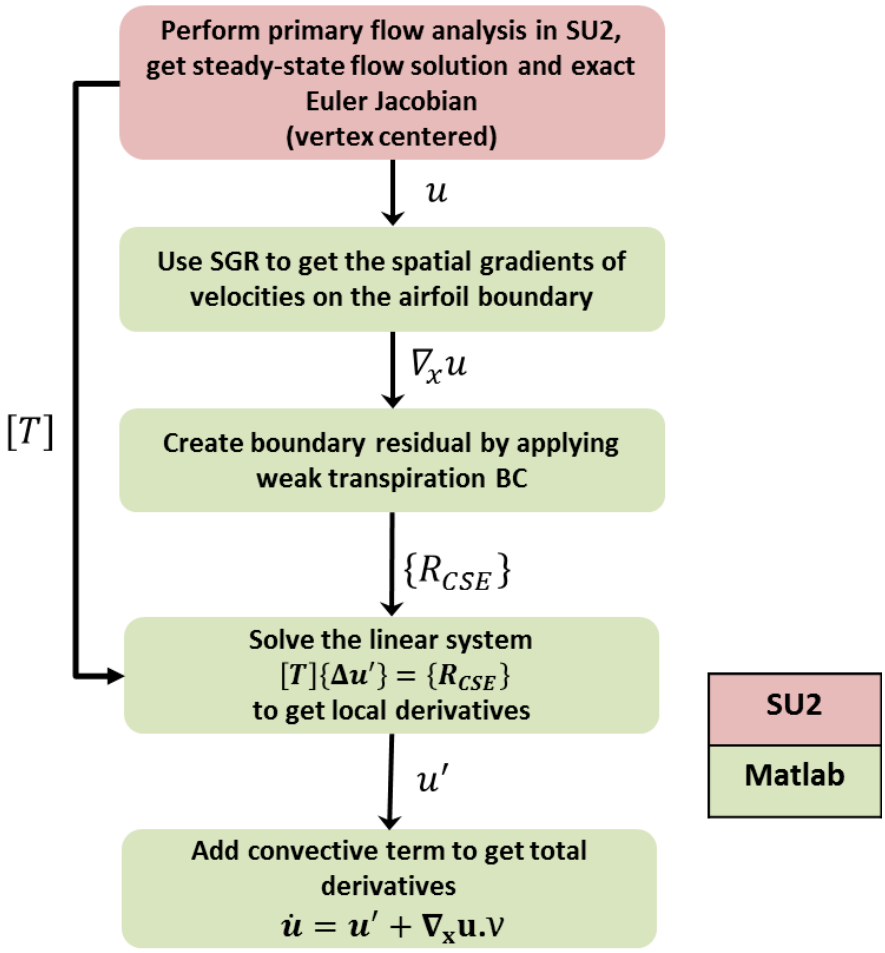

Figure 2.20: Nonintrusive process when Jacobian matrix can be output from flow solver such as SU2.

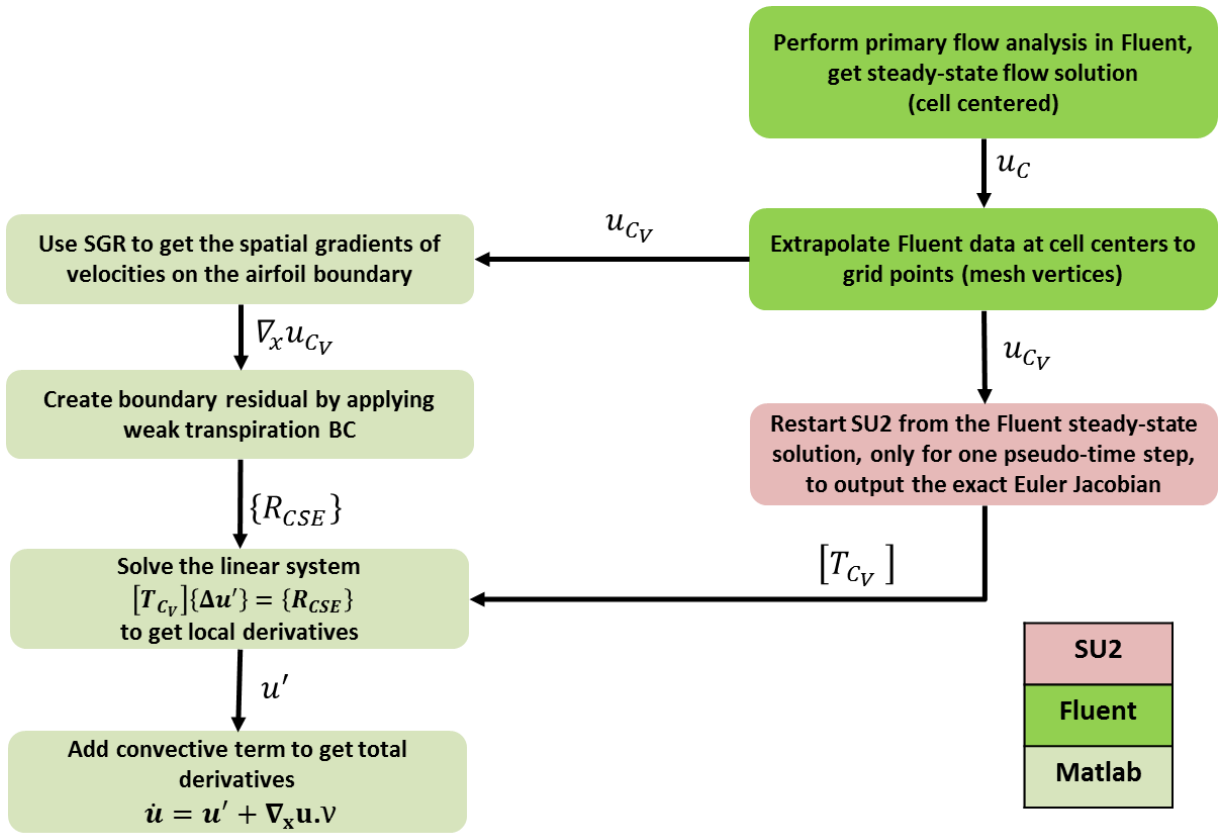

Figure 2.21: Nonintrusive process when Jacobian matrix cannot be output from flow solver such as Fluent.

in the context of nonintrusive sensitivity analysis.

2.4.3 Discretizing CSE Boundary Conditions

The CSE boundary conditions (2.47–2.48) have been written for the continuous domain. These have to discretized and implemented for the discrete system (2.55). An important contribution of the current work is the way in which these discrete boundary conditions are implemented in the finite volume formulation. Although CSE boundary conditions previously were implemented in finite difference (Borggaard and Burns, 1997) and finite element (Duvigneau and Pelletier, 2006) formulation, in this work the spatial gradients appearing in the boundary conditions are calculated using SGR and the sensitivity analysis is done nonintrusively.

The CSE farfield boundary condition (2.47) can be recognized as a form of the primary farfield boundary (2.37) where the value of the sensitivity state variables at the boundary S_F are $\boldsymbol{u'}|_\infty = 0$. Hence this boundary condition can be treated as was the primary farfield boundary condition, but with a new boundary value. The CSE transpiration boundary condition (2.48) however is non-homogeneous, unlike the homogeneous flow tangency boundary condition (2.38) in the primary analysis. Such boundary conditions are encountered in simulations involving store ejections or prescribed motion (Zhang et al., 2006; Chen and Zhang, 2013). Discretizing this boundary condition at node k on boundary S_W, we get

$$\{u'\hat{\imath} + v'\hat{\jmath}\}_k \cdot \{\hat{\boldsymbol{n}}\}_k|_{S_W} = -\left(\left(\frac{\partial u_k}{\partial x}\mathcal{V}_{x_k} + \frac{\partial u_k}{\partial y}\mathcal{V}_{y_k}\right)\hat{\imath} + \left(\frac{\partial v_k}{\partial x}\mathcal{V}_{x_k} + \frac{\partial v_k}{\partial y}\mathcal{V}_{y_k}\right)\hat{\jmath}\right) \cdot \{\hat{\boldsymbol{n}}\}_k\Big|_{S_W} - \{u\hat{\imath} + v\hat{\jmath}\}_k \cdot \left\{\hat{n}\right\}_k\Big|_{S_W}. \tag{2.56}$$

Define g_{b_k} as the right side of the above equation (2.56),

$$g_{b_k} \equiv -\left(\left(\frac{\partial u_k}{\partial x}\mathcal{V}_{x_k} + \frac{\partial u_k}{\partial y}\mathcal{V}_{y_k}\right)\hat{\imath} + \left(\frac{\partial v_k}{\partial x}\mathcal{V}_{x_k} + \frac{\partial v_k}{\partial y}\mathcal{V}_{y_k}\right)\hat{\jmath}\right) \cdot \{\hat{\boldsymbol{n}}\}_k\Big|_{S_W} \tag{2.57}$$

$$- \{u\hat{\imath} + v\hat{\jmath}\}_k \cdot \left\{\hat{\hat{n}}\right\}_k\Big|_{S_W}$$

so that the discretized boundary condition at node k can be written compactly as

$$\{u'\hat{\imath} + v'\hat{\jmath}\}_k \cdot \{\hat{\boldsymbol{n}}\}_k|_{S_W} = (u'_k n_{x_k} + v'_k n_{y_k})|_{S_W} = g_{b_k}. \tag{2.58}$$

Equation (2.56) is used to substitute g_{b_k} for the expression $(u'_k n_{x_k} + v'_k n_{y_k})$ appearing during calculation of the flux at the nodes on the boundary S_W, as shown next. Following the weak boundary condition implementation approach (Hirsch, 1990), the projected sensitivity flux at node k on boundary S_W is

$$\left\{\widetilde{\boldsymbol{F'}}\right\}_k = \left(\{\boldsymbol{F'}\}_k \hat{\imath} + \{\boldsymbol{G'}\}_k \hat{\jmath}\right) \cdot \{\hat{\boldsymbol{n}}\}_k \,, \tag{2.59}$$

where $\{\hat{\boldsymbol{n}}\}_k = (n_{x_k}\hat{\imath} + n_{y_k}\hat{\jmath})$. After substituting Equation (2.45) in (2.59) and regrouping,

$$\begin{aligned}
\left\{\widetilde{\boldsymbol{F'}}\right\}_k &= \left\{\begin{array}{c} \rho' u + \rho u' \\ p' + (\rho u)' u + (\rho u) u' \\ (\rho u)' v + (\rho u) v' \\ (\rho h_t)' u + (\rho h_t) u' \end{array}\right\}_k n_{x_k} + \left\{\begin{array}{c} \rho' v + \rho v' \\ (\rho u)' v + (\rho u) v' \\ p' + (\rho v)' v + (\rho v) v' \\ (\rho h_t)' v + (\rho h_t) v' \end{array}\right\}_k n_{y_k} \\
&= \left\{\begin{array}{c} \rho' (u n_x + v n_y) \\ p' n_x + (\rho u)' (u n_x + v n_y) \\ p' n_y + (\rho v)' (u n_x + v n_y) \\ (\rho h_t)' (u n_x + v n_y) \end{array}\right\}_k + \left\{\begin{array}{c} \rho (u' n_x + v' n_y) \\ (\rho u) (u' n_x + v' n_y) \\ (\rho v) (u' n_x + v' n_y) \\ (\rho h_t) (u' n_x + v' n_y) \end{array}\right\}_k \\
&= \left\{\begin{array}{c} \rho (g_{b_k}) \\ p' n_x + (\rho u) (g_{b_k}) \\ p' n_y + (\rho v) (g_{b_k}) \\ (\rho h_t) (g_{b_k}) \end{array}\right\}_k .
\end{aligned} \tag{2.60}$$

The subscript k in Eq. (2.60) represents the discrete values of the respective terms evaluated

at the location of node k on the boundary S_W. The term $(un_x + vn_y)$ is the component of flow velocity along the normal $\hat{\boldsymbol{n}}$. Hence, according to the flow tangency condition (2.38), at node k, $(un_x + vn_y)_k = 0$. Similarly, according to the CSE transpiration boundary condition (2.58), at node k, $(u'n_x + v'n_y)_k = g_{b_k}$. Further more, since the initial guess is $\{\boldsymbol{u'}\}_k^0 = 0$, $p'_k = 0$. Thus Eq. (2.60) simplifies to

$$\left\{\widetilde{\boldsymbol{F'}}\right\}_k = g_{b_k} \left\{ \begin{array}{c} \rho \\ \rho u \\ \rho v \\ \rho h_t \end{array} \right\}_k . \tag{2.61}$$

The value of the boundary residual at node k is then calculated as

$$\{\boldsymbol{R}_{\text{CSE}}\}_k = \left\{\widetilde{\boldsymbol{F'}}\right\}_k \Delta S_k, \tag{2.62}$$

where ΔS_k is the elemental segment length on the boundary S_W at node k.

We have considered the initial condition $\{\boldsymbol{u'}\}_i^0 = 0$ at all nodes i in the domain. Also, in this particular example, the value of design velocity is zero on the farfield boundary S_F of the domain, which results in homogeneous farfield CSE boundary condition (2.47). So, the only nonzero contribution to $\{\boldsymbol{R}_{\text{CSE}}\}_i^0$ on the right side of equation (2.55) is from $\{\boldsymbol{R}_{\text{CSE}}\}_k$, which correspond to the nodes k on the boundary S_W. It is thus clear that the CSE transpiration boundary condition (2.58) significantly affects the CSE solutions. Given this, we can list the following terms that contribute significantly to the local derivative solutions obtained by solving the CSEs.

- Spatial gradients of the velocity components on the boundary S_W contribute to g_{b_k} in equation (2.58), calculated nonintrusively from the steady-state flow solution using SGR.

- Design velocity $\boldsymbol{\mathcal{V}}(x, y) = \{\mathcal{V}_x(x, y), \mathcal{V}_y(x, y)\}^T$ on the boundary S_W contributes to g_{b_k}

in Eq. (2.58), obtained from the definition of the design variable (2.46), independent of the flow solution or mesh movement.

- Material derivative of the normal direction vector $\left\{\dot{\hat{n}}\right\}_k = \frac{D\{\hat{\boldsymbol{n}}\}_k}{Db}$ contributes to g_{b_k} in equation (2.58), obtained from the definition of the design variable (2.46), independent of the flow solution.

- Flow velocities u and v on the boundary S_W contributes to g_{b_k} in equation (2.58), obtained from the analysis solution.

- Flow density ρ and total enthalpy h_t contributes to $\left\{\widetilde{\boldsymbol{F'}}\right\}_k$ in equation (2.61), obtained from the analysis solution.

- Element lengths ΔS_k at nodes k on the boundary S_W contributes to $\{\boldsymbol{R}_{\text{CSE}}\}_k$ in equation (2.62), obtained from the geometry preprocessor used by the flow solver.

2.4.4 Results

2.4.4.1 Geometry and Grids

NACA0012 airfoil has been studied widely (Vassberg and Jameson, 2010; Anderson and Bonhaus, 1994; Jameson, 1983). In the current study we use the NACA geometry and grids used by Vassberg and Jameson (Vassberg and Jameson, 2010). The airfoil geometry is based on the NACA0012 equation

$$y(x) = \pm\frac{0.12}{0.2}\left(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4\right), \quad (2.63)$$

$$\frac{\partial y}{\partial x}(x) = \pm\frac{0.12}{0.2}\left(\frac{0.14845}{\sqrt{x}} - 0.1260 - 0.7032x + 0.8529x^2 - 0.4060x^3\right). \quad (2.64)$$

However, it is extended in chord such that the resulting sharp trailing-edge location is

$$x_{TE} = 1.0089304115.$$

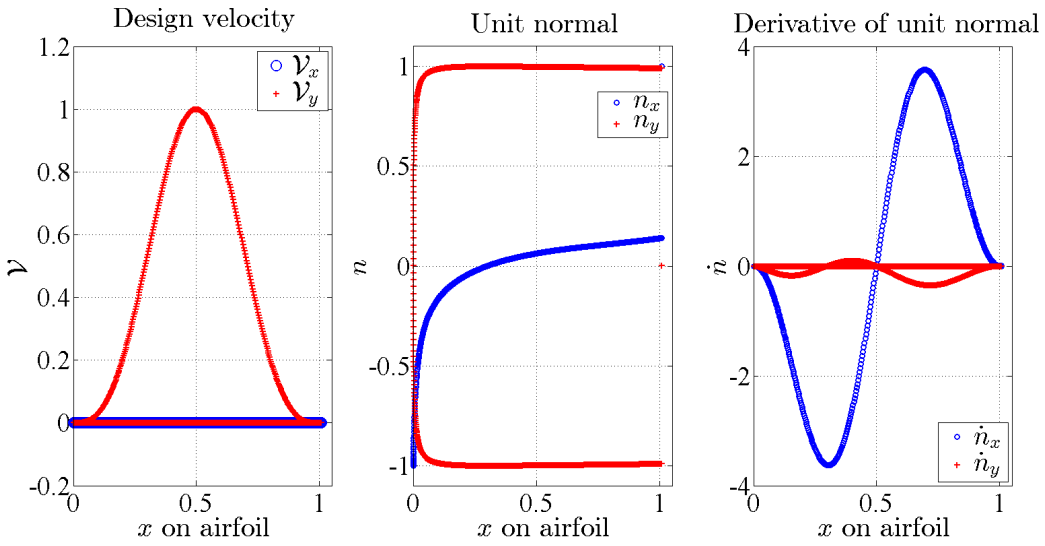

Figure 2.22: Design velocity, unit normal and derivative of unit normal on the airfoil surface.

The components of unit normal along the airfoil boundary are plotted in Figure 2.22. The airfoil geometry and the design variable defined by the Hicks-Henne bump function (2.46) are used to derive the design velocity and derivative of the unit normal, which are also plotted in Figure 2.22.

The grids used for the current study are high-quality grids prepared by Vassberg and Jameson specifically to perform a grid convergence study, based on the outcome of the 4th AIAA Drag Prediction Workshop (AIAA, 2009). The airfoil mesh is shown in shown in Figure 2.23. The grids were developed using the Karman-Treffetz transformation and are based on the standard O-mesh topology. Each quadrilateral cell of the mesh has an aspect ratio of one, and the intersecting grid lines are essentially orthogonal at each vertex within the mesh. The farfield boundary is approximately 150 chord lengths away from the airfoil. The seven meshes used in the current study have cells ranging from 32×32 cells for the coarsest mesh to 2048×2048 cells (i.e., 4 million cells) for the finest mesh. The refinement of grids is uniform such that with each refinement the edge length halves and the number of cells quadruples.

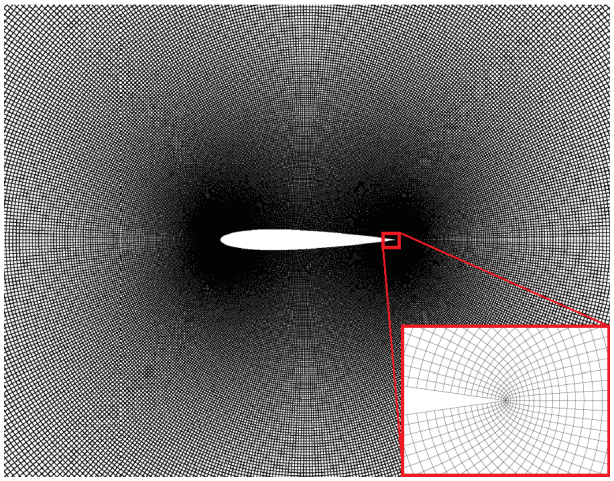

Figure 2.23: Close-up of airfoil 512×512 O-mesh and details near the trailing edge (Vassberg and Jameson, 2010).

Table 2.3: Details of the flow analysis

Parameter	Value
Mach number, M	0.5
Angle of attack,α	1.25°
Reference chord length, C_{ref}	1.0
Reference moment center, X_{ref}	0.25
Free stream pressure, p_∞	101325 N/m^2
Free stream temperature, T_∞	288.15 K
Flux scheme	Roe's II order upwind
Convergence criteria	$\log_{10}$ of L_2 norm of continuity residual less than -10

2.4.4.2 Flow Analysis

The reference values and other details used for flow analysis are shown in Table 2.3. The subcritical lifting case of Mach number $M = 0.5$ and angle of attack $\alpha = 1.25°$ was chosen for the current study. The flow analysis was done using SU2 and Fluent.

Results for the flow analysis are shown in Figures 2.24 and 2.25. The Mach number contours and C_p plots match closely to the results presented by Vassberg and Jameson (Vassberg and Jameson, 2010) for the same test case. The flow results from Fluent are almost indistinguishable from the SU2 results. Results such as these are output at each mesh level. Since the true or exact solution for the coefficients of lift and drag, C_L and C_D are not known for Euler flow analysis of NACA0012 airfoil, the best approximation of the continuum values

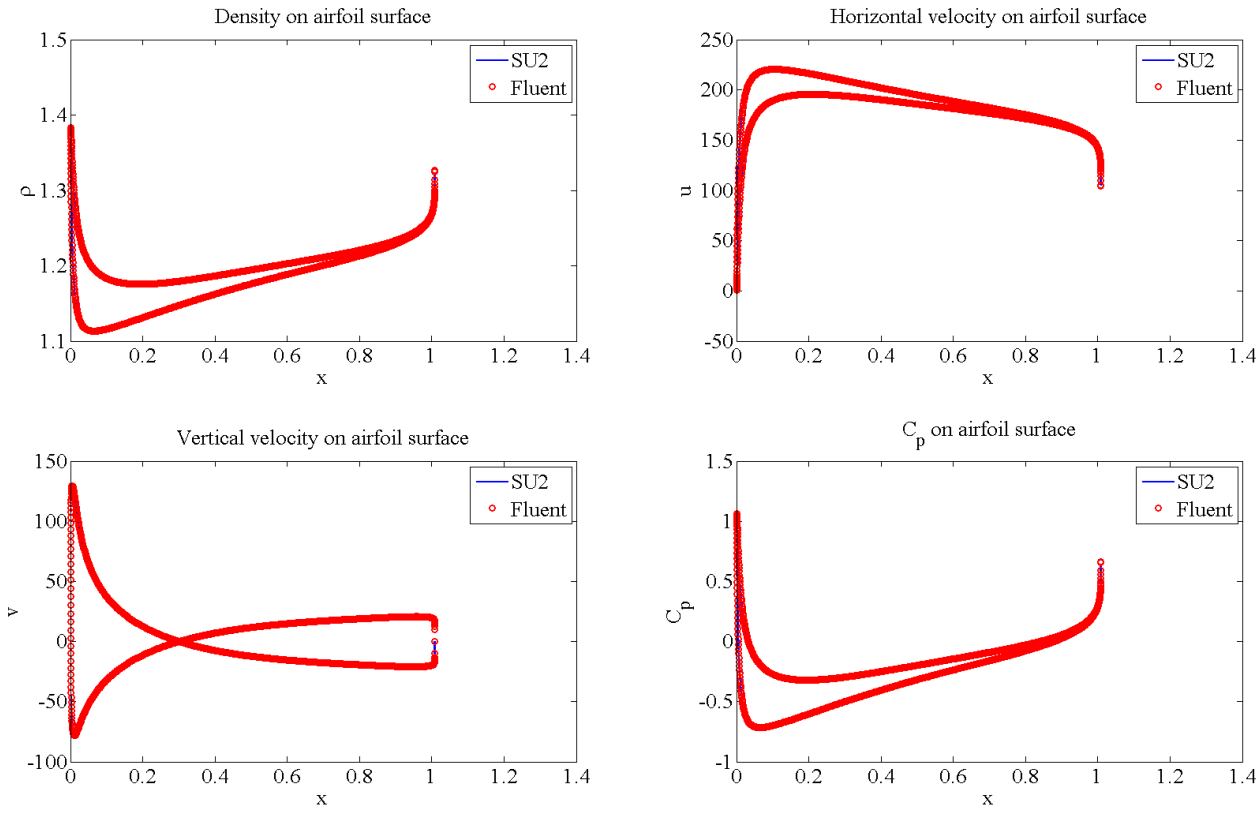

Figure 2.24: Flow solution over NACA0012 airfoil with 1024×1024 O-mesh obtained using SU2 and Fluent.

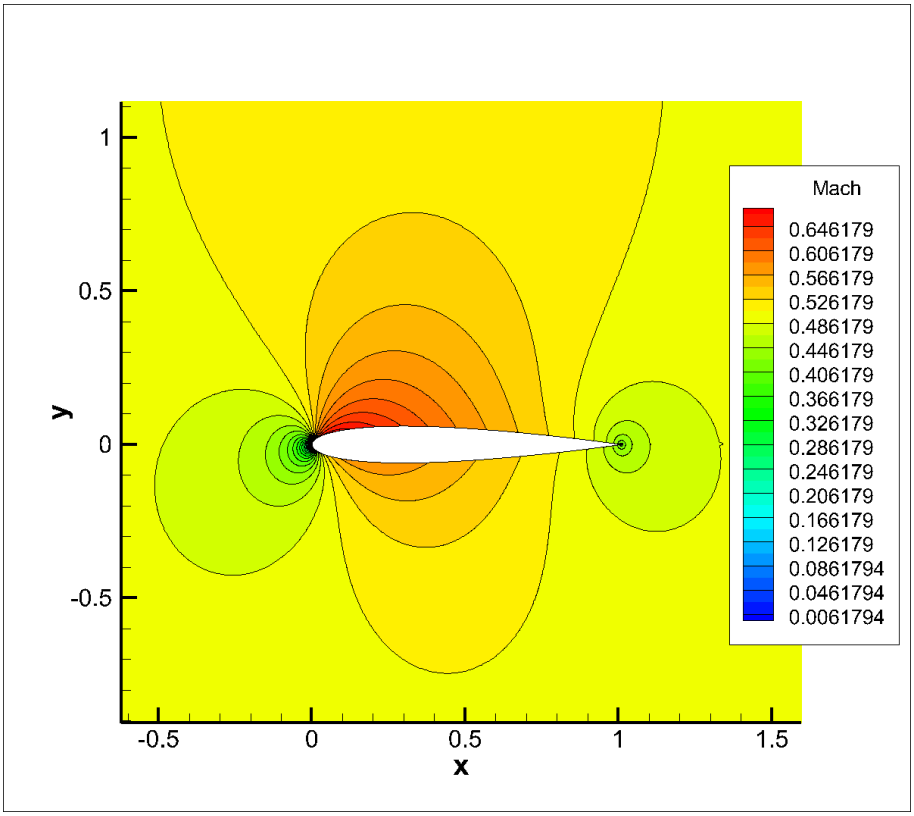

Figure 2.25: Mach line contours for flow over NACA0012 airfoil with 1024×1024 O-mesh obtained using SU2, $M_{min} = 0.0061794$, $M_{max} = 0.674184$, $\Delta M = 0.02$.

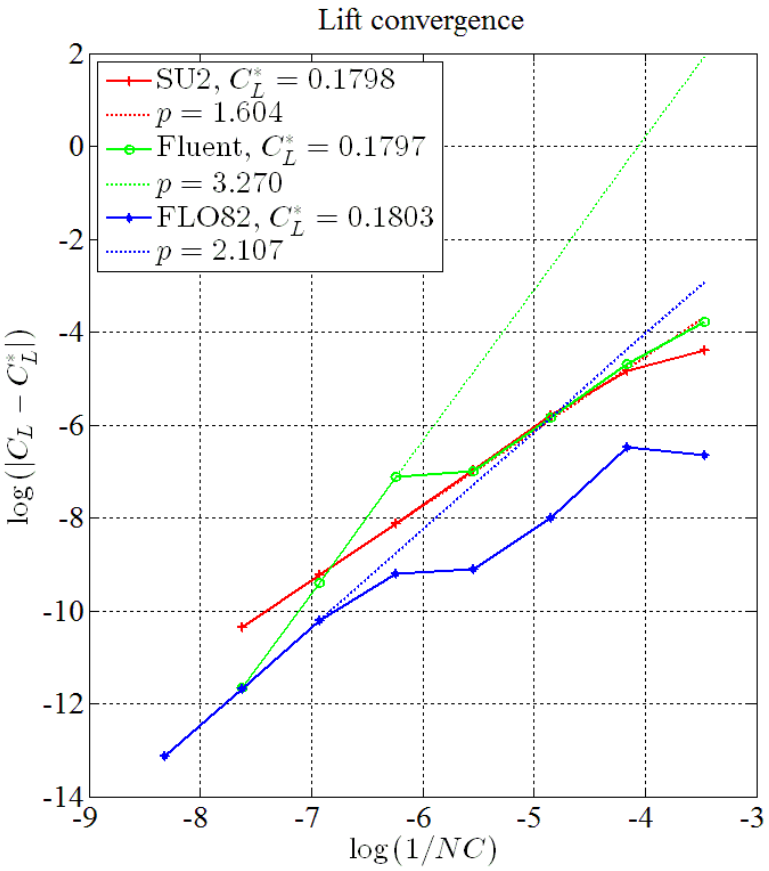

Figure 2.26: Lift convergence obtained with different solvers.

for these coefficient C_L^* and C_D^* are calculate using Richardson extrapolation and the value of the rate of convergence p based on results on the set of fine, medium and coarse grids, exactly as outlined by Vassberg and Jameson (Vassberg and Jameson, 2010). The results are plotted on the log-log scale in Figures 2.26–2.27.

Since a spatially second-order flux scheme is used for generating the results, the expected rate of convergence is 2. It is seen that the rate of convergence for Fluent results is much better compared to SU2 and FLO82 (Vassberg and Jameson, 2010) codes. The continuum values of C_L^* obtained from all three solvers are quite close. The continuum values of C_D^* obtained from SU2 and Fluent are close, but much different than the value reported for FLO82. For the SU2 and Fluent solver, the rate of convergence of C_D is better than that for C_L.

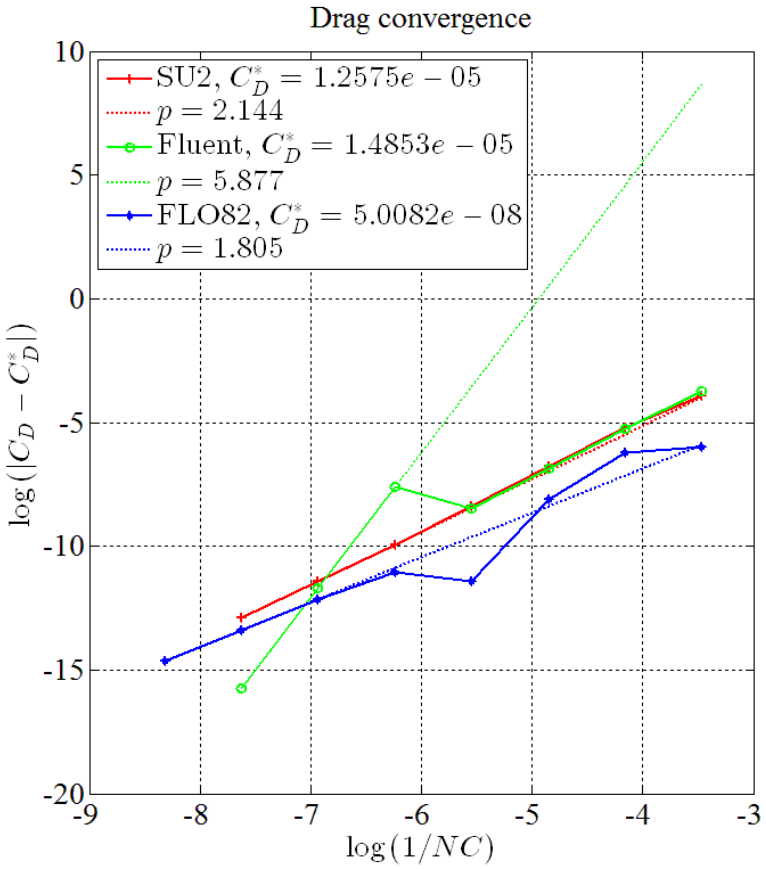

Figure 2.27: Drag convergence obtained with different solvers.

Based on the metric followed by Vassberg and Jameson (Vassberg and Jameson, 2010), the impact of functional errors on the rate of convergence can be judged by a non-dimensional error parameter β. Assume that the numerical solution of Euler equations with the use of a discrete solver gives inaccurates value of a function $\mathcal{F}$, such as coefficient of lift or drag, with an error ϵ. Then the approximate value of the function obtained from the solver would be

$$\hat{\mathcal{F}} = \mathcal{F} \pm \epsilon. \tag{2.65}$$

Then the parameter β is defined as

$$\beta = \frac{\epsilon}{|\mathcal{F}_f - \mathcal{F}_m|}, \tag{2.66}$$

where, $\mathcal{F}_f$ and $\mathcal{F}_m$ are values of the function calculated on the fine and medium meshes respectively. The effect of β on the rate of convergence p is illustrated in Figure 2.28 with upper and lower bounds on p. For example, if $\beta = 1/8$, a code which is supposed to be second-order accurate, can exhibit a rate as low as 1.58 or as high as 2.50.

To find out approximate values of β for the three codes, SU2, Fluent and FLO82, it was assumed that the values of p have errors. The expected rate of convergence is 2. This data is plotted with markers in Figure 2.28. We can deduce from the plot $\beta_{\text{SU2}} \approx 1/8$, $\beta_{\text{FLO82}} \approx 1/16$ and $\beta_{\text{Fluent}} \approx 1/3$ or higher. These values will be used later to get an idea about the error in rate of convergence of the flow derivatives. The flow analysis results are used to set BCs for the CSEs as explained next.

2.4.4.3 Spatial Gradient of Velocity and the CSE Transpiration Boundary Condition

As explained in the flowchart, Figure 2.3, results of the flow analysis and design velocity are used to set CSE BCs. Specifically, for the NACA0012 problem considered here, the CSE farfield BC (2.47) is homogeneous whereas the CSE transpiration BC (2.58) is non-

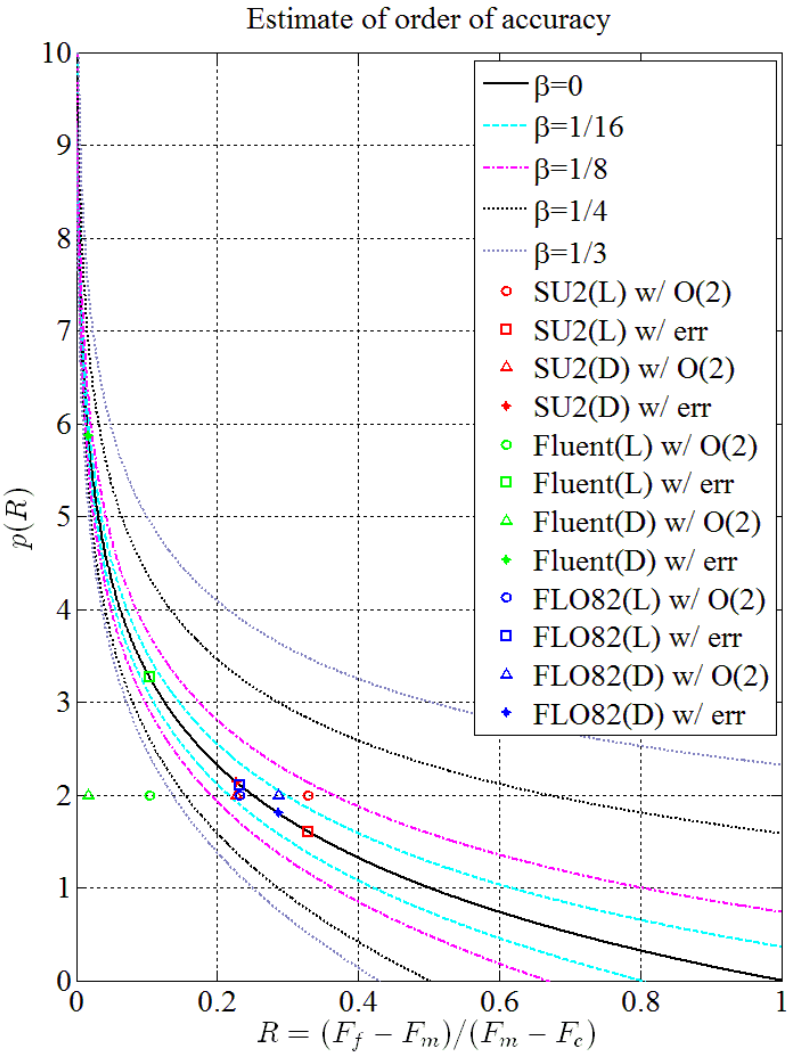

Figure 2.28: Effect of functional error on rate of convergence. Here, (L) represents data for C_L, (D) represents data for C_D, "w/ err" represents erroneous values and "w/ O(2)" represents expected values with second-order rate of convergence.

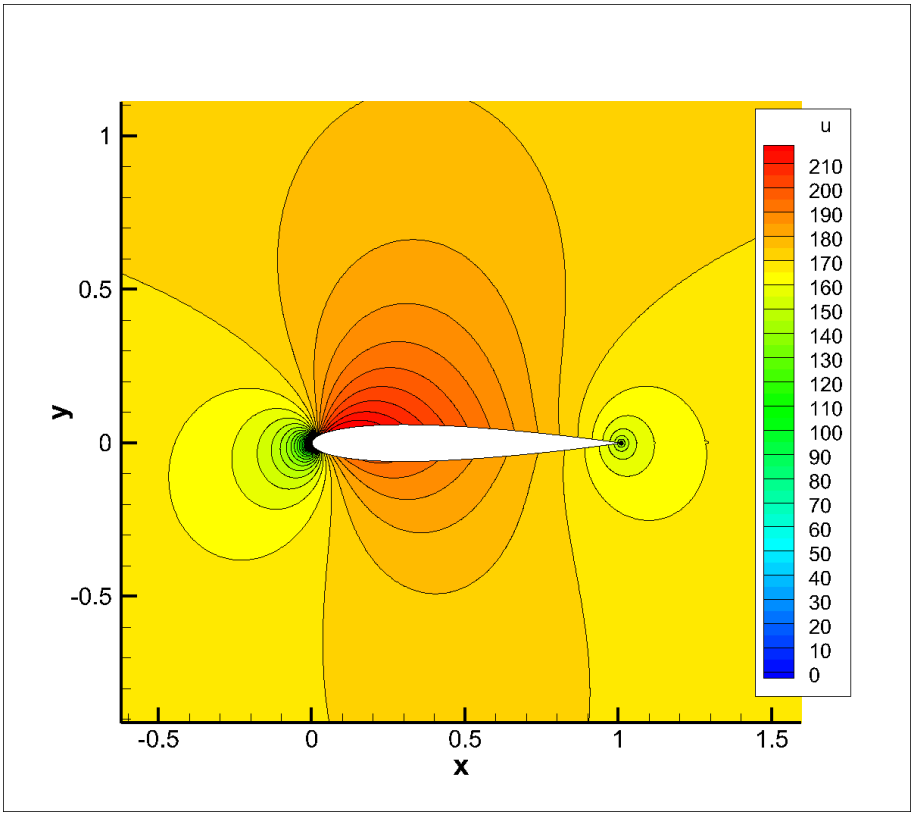

Figure 2.29: Horizontal velocity contours of flow over NACA0012 airfoil for 1024×1024 O-mesh obtained using SU2.

homogeneous. The right side of this BC, defined by g_b in Eq. (2.57), requires spatial gradients of velocities on the airfoil boundary. These spatial gradients are calculated using SGR. An SGR patch constitutes a collection of points which are used to get the spatial gradients at an expansion point. Examples of SGR patches are illustrated in Figure 2.2.

To visualize variation of velocity on the airfoil surface, the horizontal velocities are plotted in Figures 2.29–2.30. For the subcritical Mach number 0.5 considered in the present study, most of the velocity variation occurs near the airfoil leading edge, as seen in Figures 2.31–2.32. It is clear that the spatial gradients $\partial u/\partial x$, $\partial u/\partial y$, $\partial v/\partial y$, $\partial v/\partial y$ will have higher values near the leading edge than at other locations on the airfoil boundary.

Since the Hicks-Henne bump function (2.46) leads to changes only in the y coordinates of the airfoil, the design velocity component $\mathcal{V}_x$ is zero as evident in Figure 2.22. Therefore, only the y spatial gradients $\partial u/\partial y$ and $\partial v/\partial y$ are relevant. These spatial gradients obtained

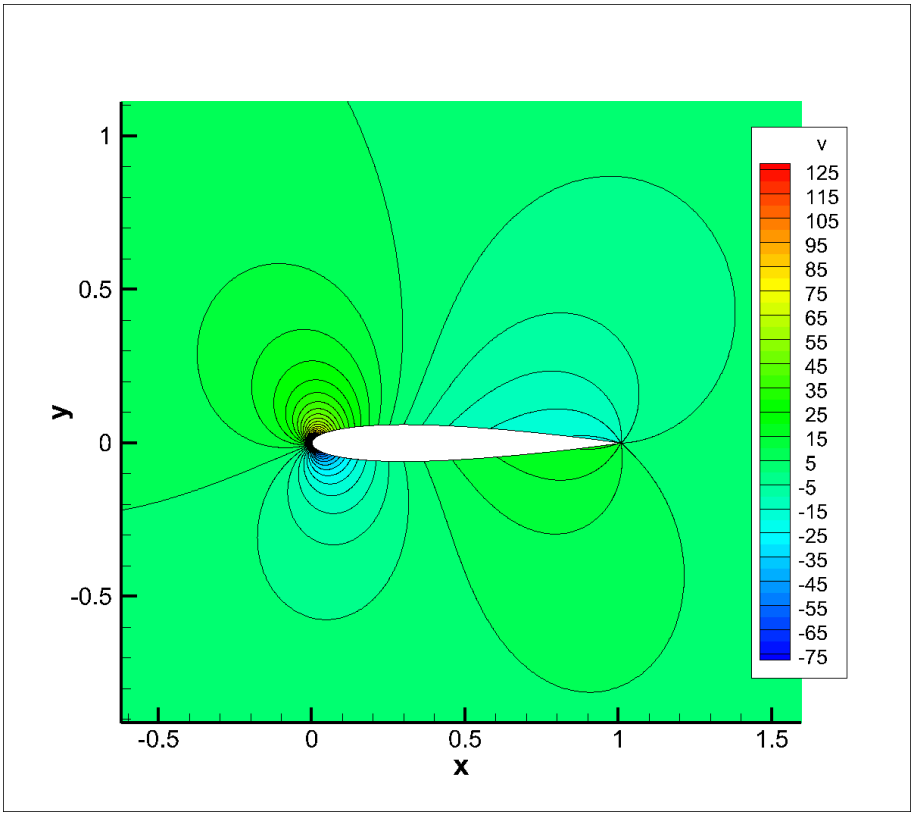

Figure 2.30: Vertical velocity contours of flow over NACA0012 airfoil for 1024×1024 O-mesh obtained using SU2.

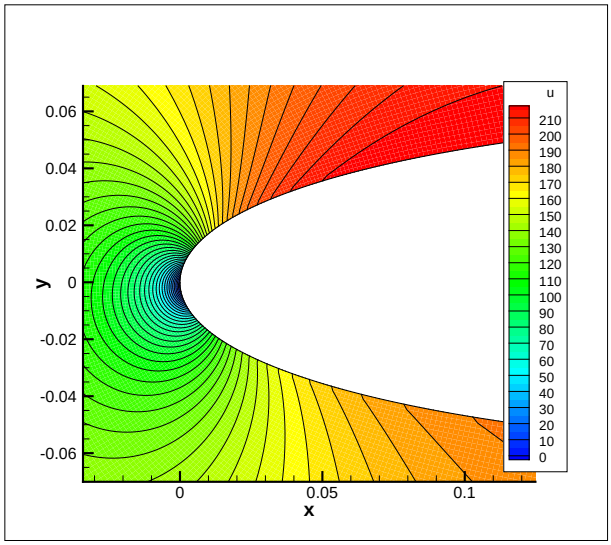

Figure 2.31: Horizontal velocity contours, near leading edge, of flow over NACA0012 airfoil for 1024×1024 O-mesh obtained using SU2.

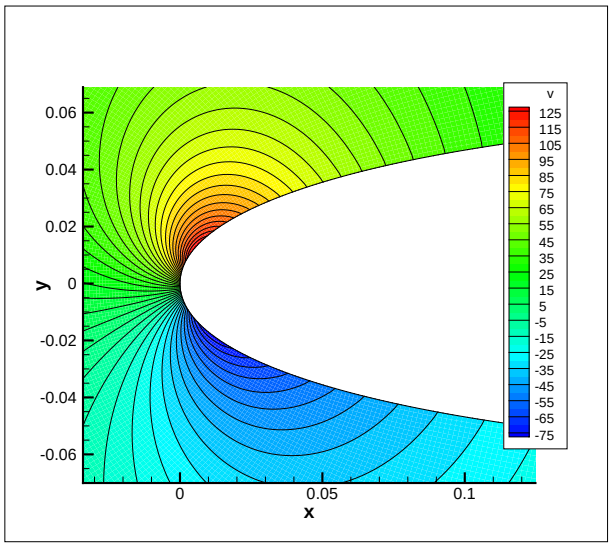

Figure 2.32: Vertical velocity contours, near leading edge, of flow over NACA0012 airfoil for 1024×1024 O-mesh obtained using SU2.

using 2-layer ($L = 2$) patches for SGR are shown in Figure 2.33. The inset in this figure shows the variation of u and v near the leading edge. Due to the difficulty of setting up the patch correctly near the trailing edge, numerical errors arise which lead to unrealistically high values of the spatial gradients near the trailing edge as seen in Figure 2.33.

Although the spatial gradients peak near the leading edge, the locations where they contribute significantly to the CSE BCs and thus to the local and total design derivatives are different from the locations near the leading edge. In the grid convergence study of shape derivative calculated using CSA with SGR, Cross and Canfield (Cross and Canfield, 2016) state that contribution of the spatial gradients, appearing in the CSE BCs, at a particular location to the design derivative solution can be quantified by the relative magnitude of the convective term at that location. Hence, to get the locations which are important from the point of view of obtaining accurate spatial gradients, the convective terms

$$C_{u_k} = \left(\frac{\partial u_k}{\partial x}\mathcal{V}_{x_k} + \frac{\partial u_k}{\partial y}\mathcal{V}_{y_k}\right) = \frac{\partial u_k}{\partial y}\mathcal{V}_{y_k}, \tag{2.67}$$

$$C_{v_k} = \left(\frac{\partial v_k}{\partial x}\mathcal{V}_{x_k} + \frac{\partial v_k}{\partial y}\mathcal{V}_{y_k}\right) = \frac{\partial v_k}{\partial y}\mathcal{V}_{y_k}, \tag{2.68}$$

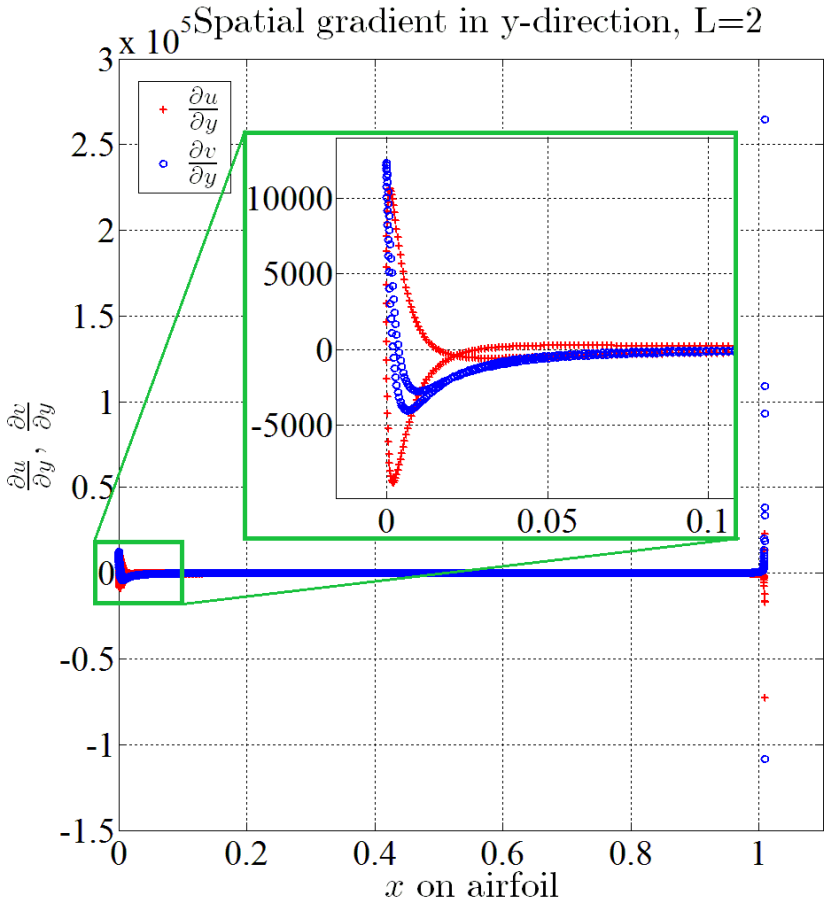

Figure 2.33: Spatial gradients of velocity on the airfoil obtained using 2-layer ($L = 2$) SGR for 1024×1024 O-mesh.

which appear in the CSE BC (2.56) are plotted at nodes x_k on the airfoil as shown in Figure 2.34. Note that the simplification in Eqs. (2.67–2.68) is possible because $\mathcal{V}_x = 0$ at all locations on the airfoil. Since the shape design variable considered in the present study perturbs only the shape of the upper surface of the airfoil, as shown in Figure 2.19, the design velocity on the lower surface is zero which results in zero C_u and C_v on the lower surface. This is seen as the horizontal lines in Figure 2.34. At all other locations, the convective terms are nonzero. Based on the relative magnitude of these convective terms, we can judge that the locations where $\partial u/\partial y$ is the most influential are near $x = 0.6256$ the locations where $\partial v/\partial y$ is the most influential are near $x = 0.4805$. Note that the numerical errors in calculation of $\partial u/\partial y$ and $\partial v/\partial y$ at the trailing edge results in small amounts of noise in the convective terms near the trailing edge; however, it does not significantly affect the CSE transpiration BC (2.56).

The convergence of the spatial derivatives at different locations with one, two and three layers used for SGR is shown in Figures (2.35–2.36). One and two layer SGR uses first-order Taylor series where as three layer SGR uses second-order Taylor series. It is seen that the convergence of the spatial gradients is much better near the leading edge, where the derivatives peak, than at the locations $x = 0.6606$, 0.4805 where they influence the boundary condition the most. One reason for this discrepancy may be that the up to 3 SGR layers may be enough to capture the local gradient accurately, which is the case at locations near the leading edge, whereas more layers may be required to capture the lower value derivatives at $x = 0.6606$, 0.4805. Another reason for the poor spatial gradients could be that the SU2 flow analysis solution, on which SGR is applied, is sub-second-order accurate. Inaccuracies in rate of convergence of the spatial gradients may also arise due to functional errors in the flow solution for which we predicted the value of $\beta = 1/8$. Based on the conclusions of Cross and Canfield (Cross and Canfield, 2016), who have done a similar study, a weighted least-squares computation might improve the SGR calculations. There is a slight improvement in the rate of convergence with 2-layer SGR, over 1 and 3-layer SGR. The next task is to find

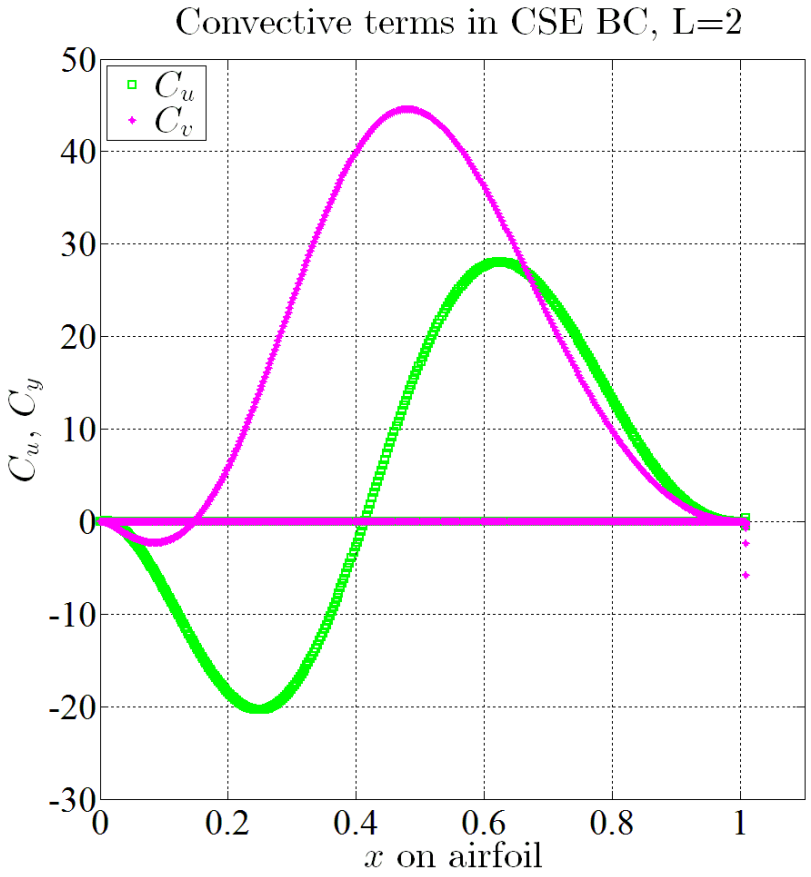

Figure 2.34: Convective terms C_u and C_v appearing in the CSE transpiration BC, calculated using using 2-layer ($L = 2$) SGR, for the 1024×1024 O-mesh.

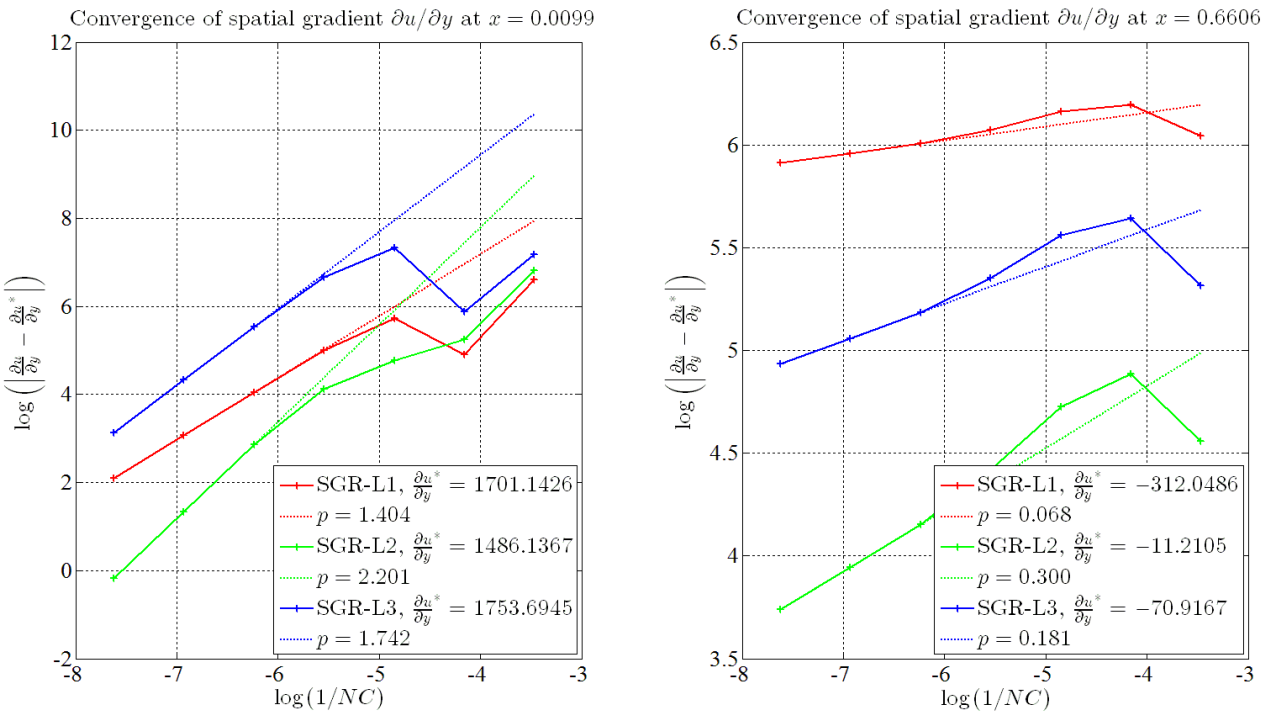

Figure 2.35: Convergence of spatial gradient $\partial u/\partial y$ at two locations using one, two and three layers for SGR; one near leading edge (left) and one near mid-chord (right).

out the effect of the accuracy of spatial gradients on the transpiration BC.

The convective terms C_u and C_y are multiplied with the components of the unit normal respectively, and further added to the $\dot{\hat{n}}$term to get the value of the transpiration velocity g_b for the CSE BC (2.56) at each node on the airfoil. This boundary condition is imposed weakly (Hirsch, 1990) as explained in Section 2.4.3. As a result, the CSE transpiration BC $u'n_x + v'n_y = g_b$ is applied exactly only in the continuum limit, i.e. as the accuracy with which this BC is imposed increases with increase in the number of cells. The value of g_b that is used to impose the BC and the value of the term $(u'n_x + v'n_y)$ after solution of the CSEs is shown in Figure 2.37. It is clear that the transpiration BC is not accurately satisfied on the airfoil owing to the weak implementation strategy. The grid convergence of the L_2 norm

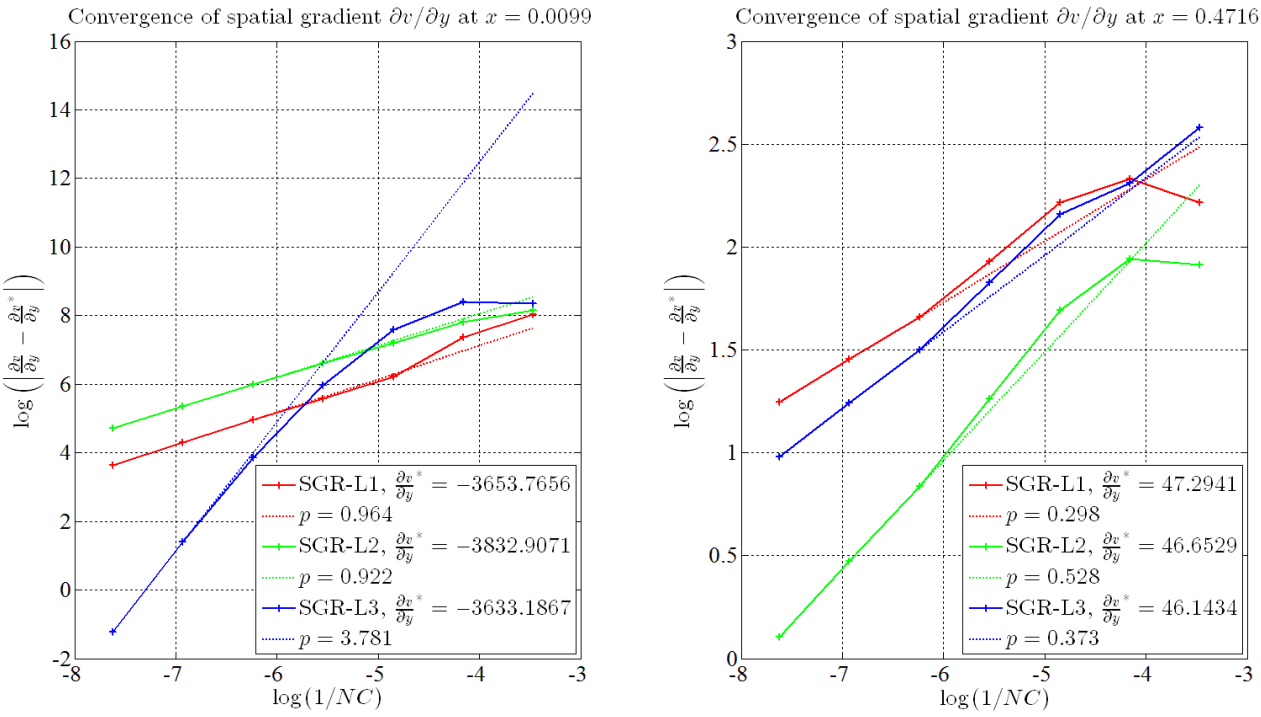

Figure 2.36: Convergence of spatial gradient $\partial v/\partial y$ at two locations using one, two and three layers for SGR; one near leading edge (left) and one near mid-chord (right).

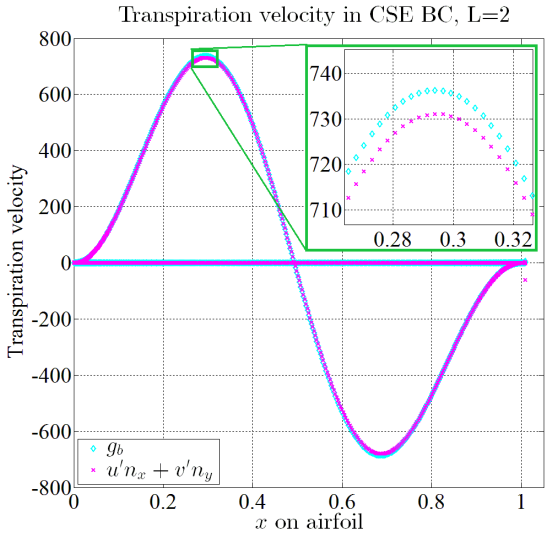

Figure 2.37: Transpiration velocity on the airfoil calculated using using 2-layer ($L = 2$) SGR with first-order Taylor series, for the 1024×1024 O-mesh.

of the transpiration velocity, defined by

$$\bar{g}_b = \sqrt{\frac{1}{N}\sum_{k=1}^{N} g_{b_k}^2} \tag{2.69}$$

where N is the number of nodes on the airfoil, is plotted in Figure 2.38. The number of layers does not have a significant effect on the norm of the transpiration velocity, and thus a similar trend could be expected for the local and total derivatives.

2.4.4.4 Total Derivatives

The total derivatives of the primary flow variables obtained using CSA and SGR with 2-layers are plotted in Figure 2.39 for two cases, (a) "CSA (SU2)" when primary analysis was done SU2, and Jacobian matrix could be output from the flow solver, according to the process outlines in Figure 2.20, and (b) "CSA (FLUENT)" when primary analysis was done using FLUENT, and Jacobian matrix could not be output from the flow solver, as outlined in Figure 2.21. In case (b) the Jacobian used for CSA is inconsistent with the primary analysis. The CSA results compare well with the finite difference derivatives obtained from SU2,

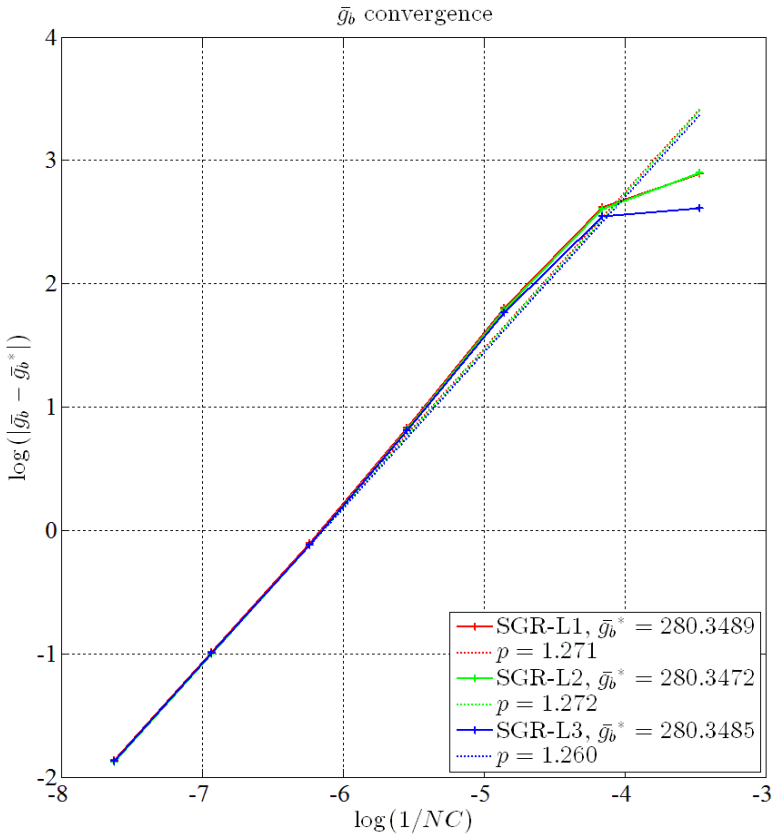

Figure 2.38: Convergence of norm of the transpiration velocity, for 1, 2, and 3 layer SGR.

labelled as "FD (SU2)" and Fluent, labelled as "FD (Fluent)." These direct derivatives are used further to obtain the derivatives of the lift and drag coefficients. The grid convergence of the lift and drag coefficients is shown in Figures (2.40–2.41). The results are compared with the derivatives obtained using the Algorithmic Differentiation (AD) method implemented in SU2. The AD results for the last couple of fine meshes are not converged and are shown with black square markers. The number of layers used in SGR does not significantly affect the CSA results. Based on the values of the absolute error, we can see that CSA yields accurate values of the lift and drag coefficient derivatives.

2.5 Conclusion

The method of CSA with SGR was used for calculating the material derivatives of 1-D and 2-D flows with respect to shape design parameters. Our focus is on the flow solutions of compressible Euler equations and incompressible Navier-Stokes equations. Most of the previous work in this area has been done using finite difference and finite element discretizations. However, challenges arise when a finite volume discretization is used due the specific way in which boundary conditions are applied. To illustrate this, an example of flow over a NACA0012 airfoil was presented that highlights the effect of the accuracy of the sensitivity boundary conditions on the derivatives of integrated flow quantities such as lift and drag. The spatial gradients of flow velocities, calculated using SGR, contribute significantly to the transpiration sensitivity boundary condition and thus affect the accuracy of total derivatives of the flow variables. Also, the weak imposition of boundary conditions, which is typical in finite volume formulations, leads to errors in the solution to the sensitivity equations. It is believed that this may be one of the reasons negatively affecting convergence of flow derivatives obtained using CSA. Nevertheless, the CSA results match well with the derivatives calculated using the automatic differentiation method and the finite difference method.

Another contribution of the current work is the nonintrusive implementation (without

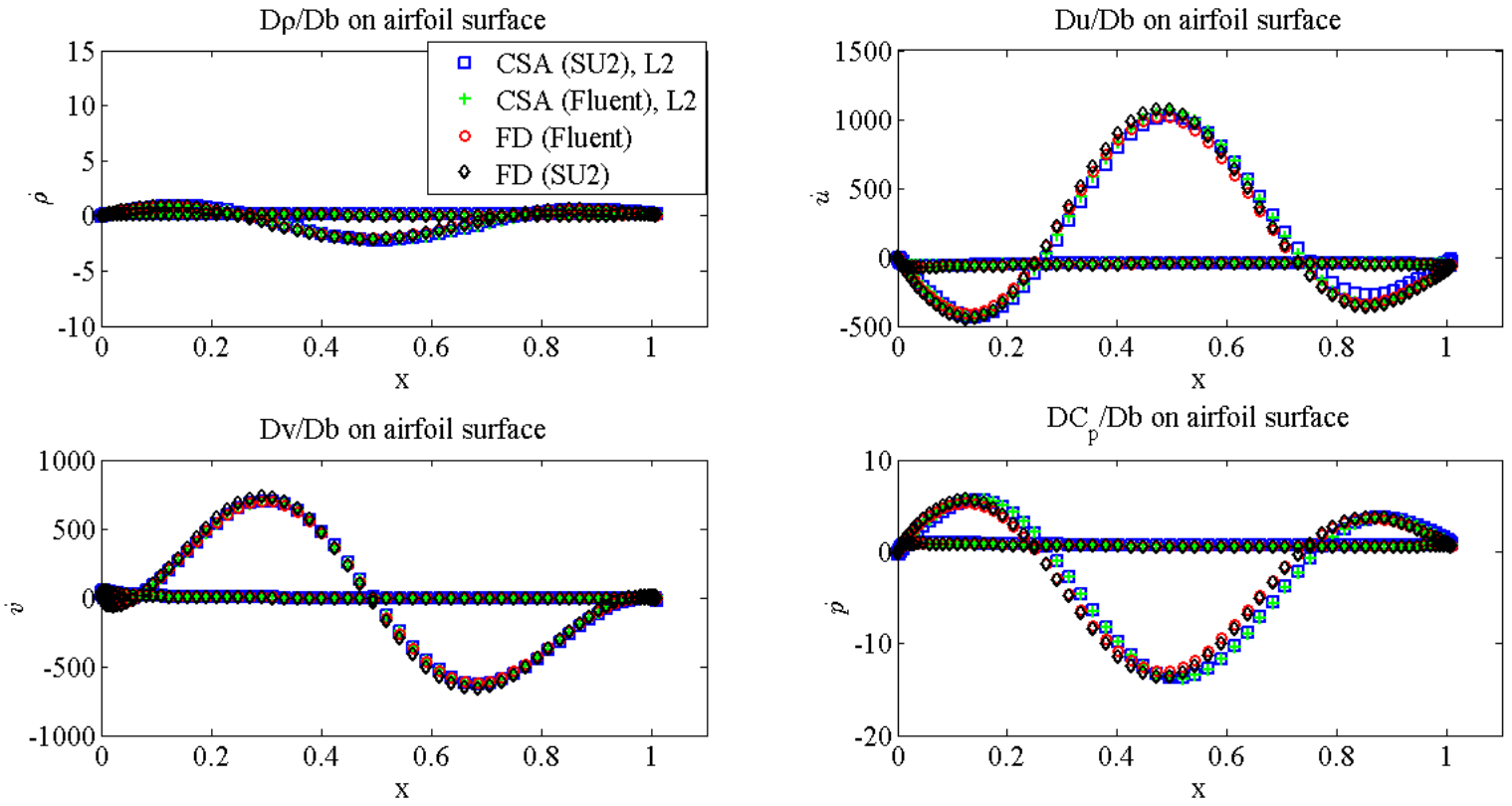

Figure 2.39: Total derivatives of density, velocity and pressure on the airfoil surface obtained using CSA and SGR with two layers and first-order Taylor series. "CSA (SU2)" and "CSA (FLUENT)" represent CSA derivatives with SU2 and FLUENT used for primary analysis, respectively. "FD (FLUENT)" and "FD (SU2)" represent finite difference derivatives calculated using FLUENT and SU2, respectively.

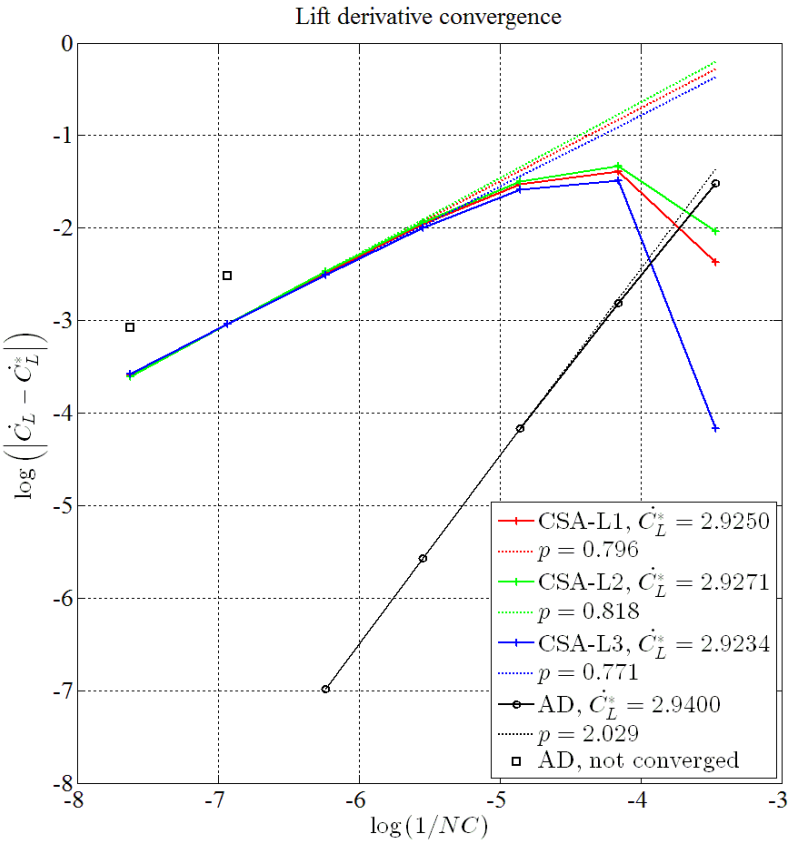

Figure 2.40: Convergence of lift derivative for 1, 2 and 3 layer SGR.

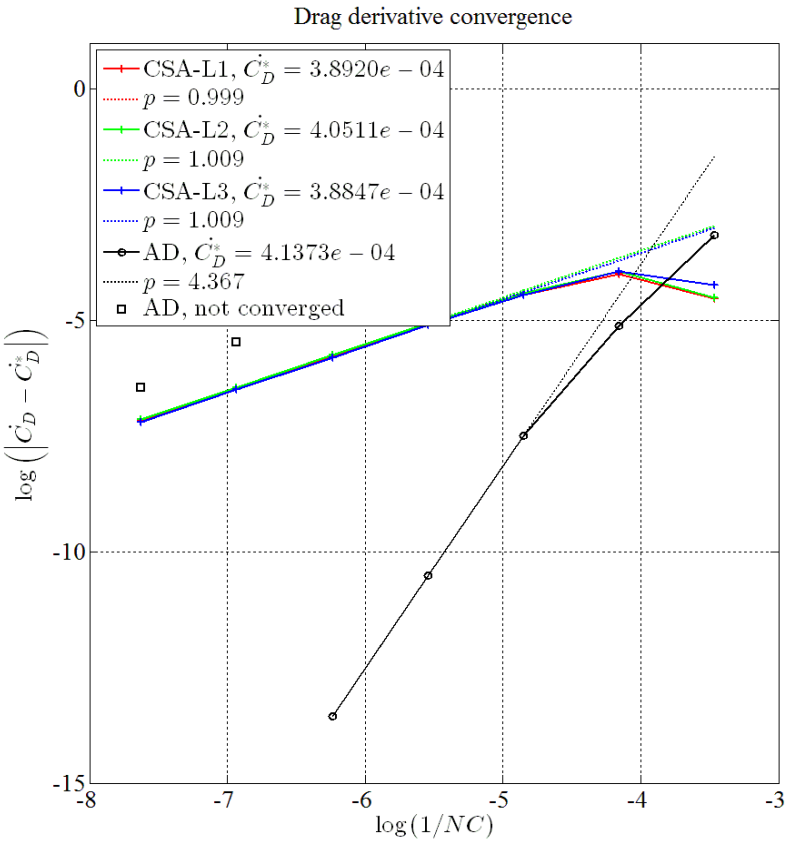

Figure 2.41: Convergence of drag derivative for 1, 2 and 3 layer SGR.

modifying the "black-box" analysis source code) of CSA for analyzing fluid systems. Particularly, we focus on the use of commonly used CFD codes, which use finite volume discretization, for solving the flow variables and their shape sensitivities. The effect of using inconsistent flow solution and tangent, or Jacobian, matrix was studied. For the NACA0012 airfoil, CSA derivatives were calculated using flow solution from the Fluent solver, and a tangent matrix exported from the SU2 solver. This establishes the use of black-box codes, such as Fluent, for obtaining flow sensitivities using the CSA approach.

A variety of examples were used to illustrate the nonintrusive application of CSA. We presented cavity flow solved using Navier-Stokes (incompressible and viscous) equations with a structured grid, finite difference spatial discretization, and an explicit temporal discretization. Flow over a NACA0012 airfoil example was solved using Euler equations with a unstructured grid, finite volume spatial discretization, and an implicit temporal discretization. A value parameter sensitivity example was also presented (Appendix A) which involves quasi-1-D flow in a convergent-divergent nozzle.

Chapter 3

Nonintrusive Continuum Sensitivity Analysis for Structural Applications: Direct and Adjoint Formulations

ABSTRACT

Continuum Sensitivity Analysis (CSA) is an approach for calculating analytic design derivatives. A direct CSA formulation is advantageous for computing derivatives of many state variables or performance functions with respect to a few shape design variables. An adjoint formulation of CSA is beneficial for computing derivatives with respect to many design variables, although continuous adjoint CSA boundary conditions can be problematic. For the continuum-discrete hybrid adjoint approach presented here, the adjoint variable is introduced after discretization, which simplifies boundary conditions. Thus, the hybrid adjoint formulation of CSA computes the same design derivatives as those obtained from direct CSA, but makes the analysis efficient for the case of a large number of design variables. Another contribution of the current work is the nonintrusive, or black-box, implementation of CSA using codes such as NASTRAN. We demonstrate that the presented method is element agnostic and can be applied with minimal modification to different element types. One-, two- and three-dimensional test cases are used to demonstrate the application of the current method.

3.1 Introduction

Continuum Sensitivity Analysis (CSA) has been developed to compute gradients for optimization of structural response (Dems and Mroz, 1984; Dems and Haftka, 1988; Arora, 1993; Choi and Kim, 2005) and fluid response (Borggaard and Burns, 1997; Stanley and Stewart, 2002; Duvigneau and Pelletier, 2006; Kulkarni et al., 2014a). Recently, it has also been used successfully to compute aeroelastic response sensitivity with respect to shape variations for coupled fluid-structure interaction problems using the boundary velocity (local) formulation for both linear and nonlinear structural analysis (Liu and Canfield, 2013a; Cross and Canfield, 2014). The motivation for using CSA for shape sensitivity of aeroelastic response is twofold: (a) gradients are computed from analytic expressions and are therefore more accurate and efficient than finite difference, and (b) mesh sensitivity is avoided, which is a drawback of the discrete analytic shape sensitivity approach.

Although the derivatives of a large number of state variables or performance measures can be calculated efficiently using the direct formulation of CSA, it requires that a linear system of equations be solved for each design variable. This makes direct CSA inefficient for a large number of design variables. On the other hand, adjoint methods require solution to only one linear system (at each time step) for each performance measure even for a large number of design variables. However, the boundary conditions associated with the continuous adjoint method are often difficult to formulate (Lozano and Ponsin, 2012; Duivesteijn et al., 2005).

In earlier work (Kulkarni et al., 2016) we introduced a hybrid adjoint formulation of CSA, based on appending discretized continuum sensitivity equations (CSEs) to the discrete performance sensitivity, weighted by an algebraic adjoint vector. A unique advantage of this method is that it allows us to derive the sensitivity boundary conditions directly for the state variables, without consideration of adjoint boundary conditions. The hybrid formulation requires a linear system of equations to be solved only once for obtaining the gradient of a performance measure with respect to many design variables. This makes CSA efficient for many design variables.

The hybrid adjoint is derived in Section 3.3 for the boundary velocity (local) formulation of CSA, so that it inherits the accuracy benefits of CSA with SGR (Cross and Canfield, 2014) used for the CSA boundary conditions. In fact, the hybrid adjoint method yields exactly the same results as by the direct local CSA method. The results computed using hybrid adjoint CSA will differ from results computed using a conventional discrete adjoint or continuous adjoint. How hybrid adjoint CSA results compare with results for the continuous adjoint formulation depends upon the manner in which the latter equations are discretized. However, unlike discrete adjoint or (discretized) continuous adjoint, hybrid adjoint CSA avoids calculation of the mesh sensitivity needed for the design derivative of the global stiffness matrix. Hybrid adjoint CSA results are potentially more accurate than the discrete adjoint results, compared to the true shape derivatives of the continuous problem, depending upon the SGR and discretization.

The local continuum sensitivity equations are derived in Section 3.2, followed by the direct and hybrid adjoint formulation in Section 3.3. An example of a 1-D axial bar is presented in Section 3.4. A 3D cantilever beam example is presented in Section 3.5.

3.2 Local Continuum Sensitivity Analysis

3.2.1 Continuous Domain Equations and Boundary Conditions

The partial differential equation governing structural response can be written compactly as

$$\boldsymbol{R}\left(\boldsymbol{u}, t; b\right) = \boldsymbol{\mathcal{A}}\left(\boldsymbol{u}, L\left(\boldsymbol{u}\right)\right) - \boldsymbol{f}\left(\boldsymbol{x}, t; b\right) = 0 \quad \text{on } \Omega, \tag{3.1}$$

with the corresponding boundary conditions (BCs)

$$\boldsymbol{\mathcal{B}}\left(\boldsymbol{u}, L\left(\boldsymbol{u}\right)\right) = \boldsymbol{g}\left(\boldsymbol{x}, t; b\right) \quad \text{on } \Gamma, \tag{3.2}$$

where the vector of dependent (state) variables $\boldsymbol{u}(\boldsymbol{x}, t; b)$ are functions of the spatial and temporal independent coordinates, $\boldsymbol{x}$ and t, respectively and depend implicitly on design

variable b. The domain and boundary in Cartesian space are shown in Figure 3.1. The linear differential operator L has terms such as $\left\{\frac{\partial}{\partial t}, \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial y^2}, \ldots\right\}$ that appear in the governing equations or boundary conditions. $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ are algebraic or integral operators acting on $\boldsymbol{u}$ and $L(\boldsymbol{u})$ in general possibly in nonlinear fashion. The distributed body force applied on the system is given by $\boldsymbol{f}$ in (3.1), and the general BCs in (3.2) can be either Dirichlet (essential or geometric) such as a prescribed value

$$\boldsymbol{\mathcal{B}}_e(\boldsymbol{u}) \equiv \boldsymbol{u}|_{\Gamma_e} = \boldsymbol{g}_e \equiv \bar{\boldsymbol{u}} \tag{3.3}$$

on the boundary Γ_e, or they may involve a differential operator for Neumann (nonessential or natural) BCs such that

$$\boldsymbol{\mathcal{B}}_n(L(\boldsymbol{u}))|_{\Gamma_n} = \boldsymbol{g}_n \tag{3.4}$$

on the boundary Γ_n. Since the current work involves steady-state analysis, the time term t in these equations is suppressed hereafter.

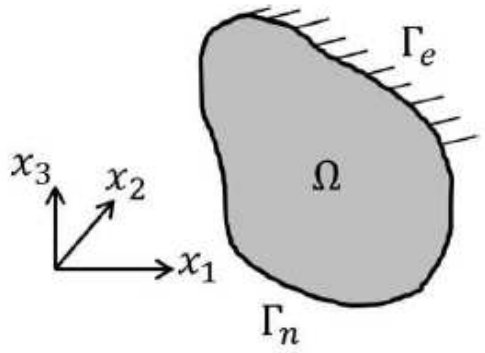

Figure 3.1: Domain, Ω, with boundary Γ.

3.2.2 Differentiation of the Continuous Equations

Consider the problem of obtaining the derivative of the steady-state response $\boldsymbol{u}(\boldsymbol{x}; b)$ with respect to design parameter b at all points in the domain. The response depends on the

spatial variable $\boldsymbol{x}$ and it may have an explicit or implicit dependence on the shape variable b, as indicated by the semicolon. The boundary velocity (local) formulation of CSA results in CSEs that are posed in terms of the local derivatives of the response, $\boldsymbol{u'} = \partial \boldsymbol{u}/\partial b$. Hence, solution of the CSEs yields the local derivative. The total or material derivative $\dot{\boldsymbol{u}} = D\boldsymbol{u}/Db$ is then obtained by adding the convective term to the local derivative.

$$\frac{D\boldsymbol{u}}{Db} = \frac{\partial \boldsymbol{u}}{\partial b} + \sum_{i=1}^{3} \frac{\partial \boldsymbol{u}}{\partial x_i}\frac{\partial x_i}{\partial b} \quad \Longleftrightarrow \quad \dot{\boldsymbol{u}} = \boldsymbol{u'} + \boldsymbol{\nabla}_{\boldsymbol{x}}(\boldsymbol{u}) \cdot \boldsymbol{\mathcal{V}} \tag{3.5}$$

The convective term consists of the spatial gradients of the response $\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u} = \partial \boldsymbol{u}/\partial \boldsymbol{x}$, and the geometric sensitivity or design velocity $\boldsymbol{\mathcal{V}}(\boldsymbol{x}) = \partial \boldsymbol{x}/\partial b$, which depends on the geometric parametrization of the domain. For value design parameters independent of shape, the convective term goes to zero, because the design velocity is zero, and so the material derivative is same as the local derivative. However, for shape design variables, the design velocity is not zero and hence there is a need to calculate the convective term for CSE boundary conditions and transformation to material derivative wherever necessary.

CSA is based on the philosophy of "differentiate and then discretize" and involves differentiating Eqs. (3.1) and (3.2) with respect to b, followed by discretization and solution of the resulting discretized system. Based on the type of differentiation, CSA is categorized as either local form CSA or total form CSA (Liu and Canfield, 2016) in accordance with Eq. (3.5). The local form CSA involves partial differentiation of Eq. (3.1), while the total form CSA involves total (material or substantial) differentiation. Due to the advantages of the local form CSA over the total form CSA, the current work focuses on the local form CSA. The CSEs are obtained by partial differentiation of Eq. (3.1) as

$$\frac{\partial \boldsymbol{R}}{\partial b} = \frac{\partial \boldsymbol{\mathcal{A}}(\boldsymbol{u}, L(\boldsymbol{u}))}{\partial \boldsymbol{u}}\boldsymbol{u'} + \frac{\partial \boldsymbol{\mathcal{A}}(\boldsymbol{u}, L(\boldsymbol{u}))}{\partial L} L(\boldsymbol{u'}) - \frac{\partial \boldsymbol{f}(\boldsymbol{x}, t; b)}{\partial b} = 0. \tag{3.6}$$

Since the material boundary changes due to a change in the shape design parameter, the boundary conditions for the CSEs are obtained by total or material differentiation of the

original boundary conditions (3.2) and moving the convective terms to the right side

$$\frac{\partial \boldsymbol{\mathcal{B}}}{\partial u}\boldsymbol{u'} + \frac{\partial \boldsymbol{\mathcal{B}}}{\partial L} L\left(\boldsymbol{u'}\right) = \dot{g}\left(\boldsymbol{x}, t; b\right) - \boldsymbol{\mathcal{V}}\left(\boldsymbol{x}\right) \cdot \left(\frac{\partial \boldsymbol{\mathcal{B}}}{\partial u} \boldsymbol{\nabla}_{\boldsymbol{x}} \boldsymbol{u} + \boldsymbol{\nabla}_{\boldsymbol{x}} \left(\boldsymbol{\mathcal{B}} \left(L\left(\boldsymbol{u}\right)\right)\right)\right), \tag{3.7}$$

where $\dot{g}\left(\boldsymbol{x}, t; b\right)$ is the material derivative of the prescribed boundary condition, typically zero for Dirichlet boundary conditions. Nevertheless, even when the the boundary condition (3.2) is homogeneous $\left(\boldsymbol{u}|_{\Gamma_e} = 0\right)$, the CSE boundary condition (3.7) is in general non-homogeneous due to the convective term: $\boldsymbol{u'}|_{\Gamma_e} = \dot{\boldsymbol{g}}_e - \boldsymbol{\nabla}_{\boldsymbol{x}} \boldsymbol{u} \cdot \boldsymbol{\mathcal{V}}\left(\boldsymbol{x}\right)$, even for $\dot{\boldsymbol{g}}_e = 0$. The commutation of derivatives on the left side of Eq. (3.7) is possible when the derivatives are local. The CSEs (3.6) with the boundary conditions (3.7) form a linear system of equations in terms of sensitivity variable $\boldsymbol{u'}$, which can be solved by the same or different numerical method used for solving the analysis problem.

Eqs. (3.6) and (3.7) may be restated as

$$\frac{\partial \boldsymbol{R}}{\partial b} = \boldsymbol{\mathcal{A}}_b \left(\boldsymbol{u}, L\left(\boldsymbol{u'}\right)\right) - \boldsymbol{f'}\left(\boldsymbol{x}, t; b\right) = 0 \quad \text{ on } \Omega, \tag{3.8}$$

with the corresponding sensitivity BCs

$$\boldsymbol{\mathcal{B}}_b \left(\boldsymbol{u}, L\left(\boldsymbol{u'}\right)\right) = \boldsymbol{g}_b \left(\boldsymbol{x}, t; b\right) \quad \text{ on } \Gamma, \tag{3.9}$$

where $\boldsymbol{g}_b$ is the right side of Eq. (3.7). The similarity of Eqs. (3.8) and (3.9) to Eqs. (3.1) and (3.2) motivates the same solution method for each set of equations with the same mesh for the discretized form. For linear governing Eqs. (3.1), $\boldsymbol{\mathcal{A}}_b = \boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}_b = \boldsymbol{\mathcal{B}}$. For nonlinear governing equations, the solution $\boldsymbol{u}$ can be obtained from the analysis solution of Eq. (3.1) for use in Eqs. (3.8) and (3.9). The load or forcing terms for the CSEs appear at two places:

- Body loads $\boldsymbol{f'}$: These loads can be calculated as

$$\boldsymbol{f'} = \frac{\partial f}{\partial b} = \frac{D\boldsymbol{f}}{Db} - \boldsymbol{\nabla}_{\boldsymbol{x}} \left(\boldsymbol{f}\right) \cdot \boldsymbol{\mathcal{V}} \tag{3.10}$$

and are typically zero for a shape variable, unless the body loads $\boldsymbol{f}$ explicitly or implicitly depend on the shape variable b.

- Boundary loads $\boldsymbol{g}_b$: These loads arise due to the non-homogeneous CSE boundary conditions (3.9) and are typically the main forcing terms that drive the local derivatives $\boldsymbol{u}'$. As shown in Eq. (3.7), $\boldsymbol{g}_b$ includes spatial derivatives $\boldsymbol{\nabla}_x \boldsymbol{u}$ of the response. Thus, the accuracy of the spatial gradients $\boldsymbol{\nabla}_x \boldsymbol{u}$ directly affects the accuracy of the local derivatives $\boldsymbol{u}'$.

Eqs. (3.8) and (3.9) which govern local derivatives are derived on the continuous domain Ω bounded by Γ; hence, the name of the method (local) Continuum Sensitivity Analysis. The next step involves discretization of the domain and numerical solution of the CSEs on the discretized domain.

3.3 Direct and Hybrid Adjoint Formulations

The local CSEs are obtained by partial differentiation of the continuum domain equation, whereas for shape design variables the corresponding sensitivity boundary conditions are obtained by material (or total) differentiation of the continuum boundary conditions. Once the sensitivity equations and boundary conditions are set, they can be discretized and solved using the available method. The discretized CSEs are a system of linear equations. The solution procedure of this linear system of equations can be changed to get either the direct or the adjoint formulation, as shown next.

3.3.1 Direct Formulation of Continuum Sensitivity Analysis

The present work is restricted to static analysis, or analysis at a particular instance of time during a dynamic analysis. Therefore, the variable for time t is omitted. Dynamic equations are recommended for future work. Similarly to the discrete system $[K]\{\boldsymbol{u}\} = \{\boldsymbol{F}\}$ for Eqs. (2.1), after discretization, CSEs (2.4) can be represented by

$$[K]\{\boldsymbol{u'}\}_i = \{\boldsymbol{F'}\}_i \,. \tag{3.11}$$

Here $[K]$ is the system, or stiffness, matrix of the discretized primary system, $\{\boldsymbol{u'}\}_i = \partial\{\boldsymbol{u}\}/\partial b_i$ denotes the solution for local derivatives and b_i is one of the n_b design variables from the set $\boldsymbol{\{b\}} = [b_1, b_2, ..., b_{n_b}]^T$. The vectors $\{\boldsymbol{u}\}$, $\{\boldsymbol{F}\}$, $\{\boldsymbol{u'}\}_i$, and $\{\boldsymbol{F'}\}_i$ are of size $N \times 1$ while the matrix $[K]$ is of size $N \times N$, where N is the number of degrees of freedom in the finite element model. For nonlinear analysis $[K]$ in Eq. (3.11) would be the converged tangent stiffness matrix (Borggaard and Burns, 1994, 1997; Cross and Canfield, 2014) of the discretized primary system. As stated earlier, although the original system may be nonlinear with respect to the state variables, the CSEs are always linear with respect to the sensitivity variables. Also, the CSEs may be solved with the same discretization used to solve the primary analysis.

The solution vector $\{\boldsymbol{u}\}$, obtained from the solution of the primary analysis, can be partitioned as

$$\{\boldsymbol{u}\} = \begin{Bmatrix} \boldsymbol{u_1} \\ \boldsymbol{u_2} \end{Bmatrix}.$$

Here, the notation $\{\boldsymbol{u}\}$ stands for a discretized version of the continuous solution $\boldsymbol{u}$ on the domain Ω. The vector $\{\boldsymbol{u_1}\} = \{\boldsymbol{u}_e\}$ corresponds to the degrees of freedom $\{\boldsymbol{u}_e\}$ which are constrained by essential (or geometric) boundary conditions on the boundary Γ_e. The vector $\{\boldsymbol{u_2}\} = \{\boldsymbol{u}_n, \boldsymbol{u}_\Omega\}^T$ consists of the degrees of freedom $\boldsymbol{u}_n$ constrained by the natural boundary conditions on the boundary Γ_n, and the unconstrained degrees of freedom $\boldsymbol{u}_\Omega$ in the interior of the domain Ω. Thus, Eq. (3.11) can be partitioned as

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u'_1} \\ \boldsymbol{u'_2} \end{Bmatrix}_i = \begin{Bmatrix} \boldsymbol{F'_1} \\ \boldsymbol{F'_2} \end{Bmatrix}_i . \tag{3.12}$$

In this equation $\left\{\boldsymbol{u'_1}\right\}_i$ represents the known local derivatives of the constrained degrees of freedom and $\left\{\boldsymbol{u'_2}\right\}_i$ represents the unknown local derivatives. Subscript i represents derivat-

ives with respect to design variable b_i for $i = 1, 2, ..., n_b$. The values of $\left\{\boldsymbol{u'_1}\right\}_i$ are obtained from the CSE essential boundary conditions. Specifically, the values of $\left\{\boldsymbol{u'_1}\right\}_i$ are obtained by discretizing a particular case of Eq. (2.5) on Γ_e,

$$\boldsymbol{u'}|_{\Gamma_e} = \dot{\boldsymbol{g}}_e - \boldsymbol{\nabla_x u} \cdot \boldsymbol{\mathcal{V}}(\boldsymbol{x}). \tag{3.13}$$

The effect of perturbing a boundary is incorporated through design velocity $\boldsymbol{\mathcal{V}}(\boldsymbol{x})$ in the last term. To account for the CSE essential boundary conditions, which are typically non-homogeneous, Eq. (3.12) is reduced as

$$[K_{22}]\left\{\boldsymbol{u'_2}\right\}_i = \left\{\boldsymbol{F'}_{\text{local}}\right\}_i \tag{3.14}$$

where

$$\left\{\boldsymbol{F'}_{\text{local}}\right\}_i \equiv \left\{\boldsymbol{F'_2}\right\}_i - [K_{21}]\left\{\boldsymbol{u'_1}\right\}_i, \tag{3.15}$$

$$\left\{\boldsymbol{F'_2}\right\}_i = \left\{\begin{array}{c} \boldsymbol{F'_n} \\ \boldsymbol{F'_\Omega} \end{array}\right\}_i. \tag{3.16}$$

The term $\left\{\boldsymbol{F'_n}\right\}_i$ represents the CSE boundary loads which would arise due to a natural BC (3.4) for the primary analysis. The corresponding CSE BC can be stated as

$$\boldsymbol{\mathcal{B}}_b\left(L\left(\boldsymbol{u'}\right)\right)|_{\Gamma_n} = \dot{g}_n\left(\boldsymbol{x}, t; b\right) - \boldsymbol{\mathcal{V}}(\boldsymbol{x}) \cdot \boldsymbol{\nabla_x}\left(\boldsymbol{\mathcal{B}}_n\left(L\left(\boldsymbol{u}\right)\right)\right). \tag{3.17}$$

Cross and Canfield (Cross and Canfield, 2014) showed that the loads $\left\{\boldsymbol{F'_n}\right\}_i$ can be calculated based on the first-order spatial gradients of the secondary variables, such as forces and stresses, which are usually output from the structural solver, rather than calculating the loads from higher-order derivatives of the primary variables such as displacements. The

term $\left\{\boldsymbol{F}'_{\boldsymbol{\Omega}}\right\}_i$ is known from the prescribed body force on the domain. In other words, $\left\{\boldsymbol{F}'_{\boldsymbol{\Omega}}\right\}_i$ is the discretization of the load $\boldsymbol{f'}$ given in Eq. (3.10). The force term $\left\{\boldsymbol{F}'_{\boldsymbol{\Omega}}\right\}_i$ is typically zero, unless the prescribed body force changes due to the shape variable.

The direct CSA formulation involves solving the discrete sensitivity equations (3.11), or after applying the boundary conditions, solving the reduced system (3.14). The advantages of the direct formulation are:

- The derivatives of response $\boldsymbol{u}$ are obtained at all locations in the domain.
- One linear system of equations is solved per design variable to obtain the local design derivatives at all locations.
- Analysis BCs (2.2) are differentiated directly to obtain CSE BCs (3.13) and (3.17).

3.3.2 Adjoint Formulation of Continuum Sensitivity Analysis

Let a performance measure for the discretized system be given by

$$\psi_j = \{\boldsymbol{z}\}_j^T \{\boldsymbol{u}\} \tag{3.18}$$

where $j = 1, 2, ..., n_\psi$ for n_ψ number of performance measures and $\{\boldsymbol{z}\}_j$ is assumed constant here to simplify the presentation. The so-called virtual load vector $\{\boldsymbol{z}\}_j$, as defined by Haug and Arora (Arora and Haug, 1979), may be an index to the response at a point of interest or result from numerical quadrature of an integral. The definition of the performance measure indicates that it is based on discretized solution $\{\boldsymbol{u}\}$ instead of the continuous solution $\boldsymbol{u}$.

The hybrid adjoint approach is based on appending the discretized CSE (3.14) to the local derivative of the performance measure (3.18). The local derivative of ψ_j with respect to a shape design variable b_i is given by

$$\psi'_{j,i} = \frac{\partial \psi_j}{\partial u_k}\frac{\partial u_k}{\partial b_i} = \{\boldsymbol{z}\}_j^T \{\boldsymbol{u'}\}_i = \left[\begin{array}{cc} \boldsymbol{z_1}^T & \boldsymbol{z_2}^T \end{array}\right]_j \left\{\begin{array}{c} \boldsymbol{u'_1} \\ \boldsymbol{u'_2} \end{array}\right\}_i \tag{3.19}$$

where $j = 1, 2, ..., n_\psi$ for n_ψ number of performance measures and $i = 1, 2, ..., n_b$ for n_b number of design variables. Without loss of generality, we can assume that $\{\boldsymbol{z}\} = \partial\psi_j/\partial u$ does not depend on the design variable b_i. Here $\{\boldsymbol{z_1}\}$ and $\{\boldsymbol{z_2}\}$ are partitions of $\{\boldsymbol{z}\}$ corresponding to $\{\boldsymbol{u_1}\}$ and $\{\boldsymbol{u_2}\}$, respectively. We then append CSE (3.14) to Eq. (3.19) using the adjoint variable vector $\{\boldsymbol{\lambda}\}$.

$$\begin{aligned}
\psi'_{ji} &= \{\boldsymbol{z}\}_j^T \{\boldsymbol{u'}\}_i + \{\boldsymbol{\lambda}\}_j^T \left([K_{22}] \left\{\boldsymbol{u'_2}\right\}_i - \left\{\boldsymbol{F'}_{\text{local}}\right\}_i \right) \\
&= \left[\begin{array}{cc} \boldsymbol{z_1}^T & \boldsymbol{z_2}^T \end{array} \right]_j \left\{ \begin{array}{c} \boldsymbol{u'_1} \\ \boldsymbol{u'_2} \end{array} \right\}_i + \{\boldsymbol{\lambda}\}_j^T \left([K_{22}] \left\{\boldsymbol{u'_2}\right\}_i - \left\{\boldsymbol{F'}_{\text{local}}\right\}_i \right) \\
&= \{\boldsymbol{z_1}\}_j^T \left\{\boldsymbol{u'_1}\right\}_i + \{\boldsymbol{z_2}\}_j^T \left\{\boldsymbol{u'_2}\right\}_i + \{\boldsymbol{\lambda}\}_j^T \left([K_{22}] \left\{\boldsymbol{u'_2}\right\}_i - \left\{\boldsymbol{F'}_{\text{local}}\right\}_i \right) \\
&= \{\boldsymbol{z_1}\}_j^T \left\{\boldsymbol{u'_1}\right\}_i - \{\boldsymbol{\lambda}\}_j^T \left\{\boldsymbol{F'}_{\text{local}}\right\}_i + \left(\{\boldsymbol{z_2}\}_j^T + \{\boldsymbol{\lambda}\}_j^T [K_{22}] \right) \left\{\boldsymbol{u'_2}\right\}_i
\end{aligned} \tag{3.20}$$

Further, we let $\{\boldsymbol{z_2}\}_j^T + \{\boldsymbol{\lambda}\}_j^T [K_{22}] = 0$ to yield solution to the discrete adjoint variable vector.

$$\{\boldsymbol{\lambda}\}_j = -[K_{22}]^{-T} \{\boldsymbol{z_2}\}_j \tag{3.21}$$

This expression for the adjoint vector is the same as for the conventional discrete adjoint vector (Arora and Haug, 1979). Indeed it is the same as Eqs. (10) and (23) found in (Arora and Haug, 1979). However, the manner in which it is used differs in the following equation for the performance derivative. Substitution of Eq. (3.21) into the first part of (3.20), reduces the performance measure local derivative to

$$\psi'_{ji} = \{\boldsymbol{z_1}\}_j^T \left\{\boldsymbol{u'_1}\right\}_i - \{\boldsymbol{\lambda}\}_j^T \left\{\boldsymbol{F'}_{\text{local}}\right\}_i . \tag{3.22}$$

The term $\left\{\boldsymbol{F'}_{\text{local}}\right\}_i$ multiplying the adjoint vector $\{\boldsymbol{\lambda}\}_j$ in Eq. (3.22), according to Eq. (3.15), involves the derivative of the response on the boundary, known from Eq. (3.13). It differs from the term multiplying the adjoint vector in the discrete adjoint method, which contains instead the product of the response vector with the global stiffness matrix design

derivative, as given by Eq. (13) in (Arora and Haug, 1979). This highlights the benefit of the current approach in avoiding mesh sensitivity needed for the global stiffness matrix design derivative. Both the hybrid and discrete methods include, as well, the derivative of externally applied load $\left\{\boldsymbol{F_2'}\right\}_i$ with respect to the design variable, if it is nonzero. Yang and Botkin derived a similar expression in their discretization of the continuous adjoint method (Yang and Botkin, 1986), except that they limited the discretization to a layer of elements adjacent to the boundary and implicitly relied upon element shape function spatial derivatives in their derivation. They related their discrete approximation to a finite difference approximation of the stiffness matrix, assuming an unperturbed finite element mesh aside from elements on the boundary that undergo a shape change. They thereby established a relationship of variational design sensitivity analysis to discrete semi-analytic sensitivity analysis.

The total or material derivative of ψ_j can be obtained by addition of the convective term as follows.

$$\dot{\psi}_{ji} = \{\boldsymbol{z}\}_j^T \{\dot{\boldsymbol{u}}\}_i = \{\boldsymbol{z}\}_j^T \left\{\{\boldsymbol{u'}\}_i + \text{diag}\left([\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u}]^T [\boldsymbol{\mathcal{V}}]_i\right)\right\} = \psi'_{ji} + \{\boldsymbol{z}\}_j^T \left\{\text{diag}\left([\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u}]^T [\boldsymbol{\mathcal{V}}]_i\right)\right\} \tag{3.23}$$

The last, convective term in Eq. (3.23) is comprised of the spatial gradient matrix $[\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u}]$ of size $3 \times N$ and the design velocity matrix $[\boldsymbol{\mathcal{V}}]_i$ of size $3 \times N$. The spatial gradient matrix is computed by spatial gradient reconstruction (SGR) (Cross and Canfield, 2014) and the design velocity matrix is either prescribed or can be computed using an intrusive or nonintrusive procedure (Kulkarni et al., 2014b). The procedure for implementation of hybrid adjoint CSA formulation is illustrated in Figure 3.2.

3.3.3 Approximation of Spatial Gradients

The forcing terms for the CSEs consist of the body loads $\boldsymbol{f'}$, as given in Eq. (3.10), and the loads arising from the CSE boundary conditions $\boldsymbol{g}_b$. Typically the loads $\boldsymbol{f'}$ can be derived

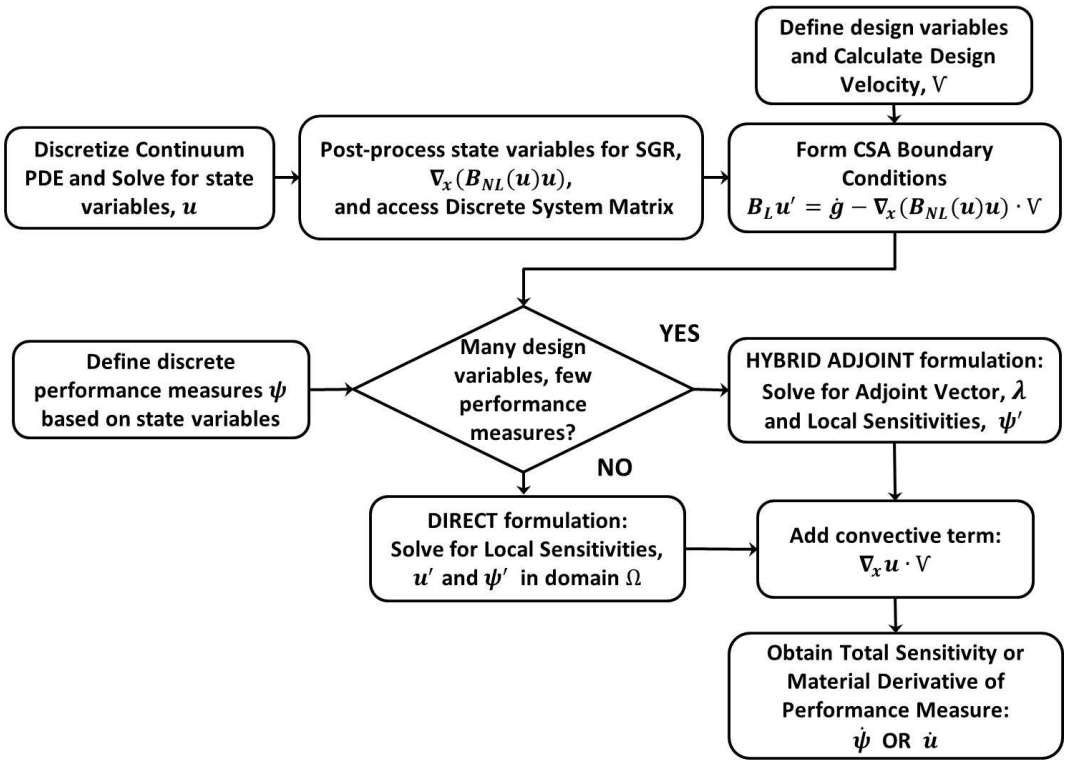

Figure 3.2: Flowchart for implementation of the hybrid adjoint Continuum Sensitivity Analysis

analytically, since the analysis body loads $\boldsymbol{f}$ are known. However, $\boldsymbol{g}_b$ requires calculation of the spatial gradient $\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u}$ or $\boldsymbol{\nabla}_{\boldsymbol{x}}\left(\boldsymbol{\mathcal{B}}_n\left(L\left(\boldsymbol{u}\right)\right)\right)$ as shown in Eqs. (3.13) and (3.17). After discretization, this leads to the load terms $\left\{\boldsymbol{F}'_{\boldsymbol{n}}\right\}_i$ and $\left\{\boldsymbol{u}'_{\mathbf{1}}\right\}_i$. Essentially, spatial gradients of analysis primary responses such as displacements, or of secondary responses such as forced or stresses, at the boundary Γ are required to be approximated from the analysis solution to obtain loads for the sensitivity analysis. Finally, after the solution to local derivatives is obtained, spatial gradients of the responses are required to calculate the convective term $\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u}\cdot\boldsymbol{\mathcal{V}}$ which is added to get the total derivatives according to Eq. (2.3). In summary, accuracy of the total derivatives obtained using CSA is significantly affected by the accuracy of the boundary spatial gradients (Duvigneau and Pelletier, 2006). Elsewhere we have shown that reliance upon low-order element shape functions limits the accuracy of these spatial gradients thus limiting the accuracy of the local and effectively total derivatives (Liu and Canfield, 2013a). As a remedy, Cross and Canfield (2014; 2015; 2016) proposed the spatial gradient reconstruction (SGR), which is based on the least-squares patch-recovery approach used by Duvigneau and Pelletier (2006). The difference between the latter two approaches is that SGR is used to reconstruct first-order gradients of the natural boundary conditions directly, instead of reconstructing the higher-order gradients that appear in them. In the current work, SGR was applied to get 2D and 3D spatial gradients of displacements and stresses. As an example, for a 4-layer, third-order SGR, the following third-order Taylor

series expansion is used to solve the least-squares problem.

$$\begin{aligned}
\phi(x+\Delta x,\, y+\Delta y,\, z+\Delta z) &= \phi(x,\, y,\, z) + \phi_{,x}\Delta x + \phi_{,y}\Delta y + \phi_{,z}\Delta z \\
&+ \frac{1}{2}\phi_{,xx}\left(\Delta x\right)^2 + \frac{1}{2}\phi_{,yy}\left(\Delta y\right)^2 + \frac{1}{2}\phi_{,zz}\left(\Delta z\right)^2 \\
&+ \phi_{,xy}\Delta x\Delta y + \phi_{,yz}\Delta y\Delta z + \phi_{,zx}\Delta z\Delta x \\
&+ \frac{1}{6}\phi_{,xxx}\left(\Delta x\right)^3 + \frac{1}{6}\phi_{,yyy}\left(\Delta y\right)^3 + \frac{1}{6}\phi_{,zzz}\left(\Delta z\right)^3 \\
&+ \frac{1}{2}\phi_{,xxy}\left(\Delta x\right)^2\Delta y + \frac{1}{2}\phi_{,xxz}\left(\Delta x\right)^2\Delta z + \frac{1}{2}\phi_{,yyz}\left(\Delta y\right)^2\Delta z \\
&+ \frac{1}{2}\phi_{,yyx}\left(\Delta y\right)^2\Delta x + \frac{1}{2}\phi_{,zzx}\left(\Delta z\right)^2\Delta x + \frac{1}{2}\phi_{,zzy}\left(\Delta z\right)^2\Delta y \\
&+ \phi_{,xyz}\Delta x\Delta y\Delta z, \qquad (3.24)
\end{aligned}$$

where ϕ could be a scalar such as a component of displacement or stress and $\phi_{,x}$, $\phi_{,y}$, $\phi_{,z}$ are the spatial gradients that are computed.

The advantages of SGR are as follows:

- Spatial gradients $\boldsymbol{\nabla_x u}$ can be approximated without the need of element formulation.
- Accuracy of the spatial gradients can be increased by choosing the correct number of layers and order of Taylor series for SGR.
- SGR can be done as a post processing step, following the structural analysis using any black-box tool, and this makes CSA amenable to nonintrusive implementation.

In our current work, SGR is used to approximate the spatial gradients required for CSA.

3.3.4 Application of Continuum Sensitivity Analysis to Two- and Three-Dimensional Structures

The hybrid adjoint method presented here is formulated irrespective of the dimensionality of the problem, because the performance measure local derivative ψ_j' is based on appending the discretized CSEs after the spatial dimension is accounted for in the CSA boundary

conditions, whether the domain is 1-D (Wickert et al., 2010; Kulkarni et al., 2016), 2-D (Cross and Canfield, 2014) or 3-D (Kulkarni et al., 2015). The dimensionality of the problem affects the solution in two ways: (a) the spatial gradients have to be calculated with respect to each dimension of the domain Ω, and (b) the CSA loads have to be calculated on the boundaries of the domain. For example, for a 1-D problem such as an axial bar, the spatial gradients operator $\boldsymbol{\nabla}_{\boldsymbol{x}}$ defined in Eq. (2.5) includes only one spatial derivative $\partial/\partial x$, and the CSA loads are applied on the end points of the axial bar. For a 3-D problem consisting of solid finite elements (Kulkarni et al., 2015), the spatial gradient operator involves derivatives along each spatial dimension, i.e. $\boldsymbol{\nabla}_{\boldsymbol{x}} = \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \end{bmatrix}$, and the CSA loads are applied on the boundary surfaces of the three-dimensional domain e.g. the faces of the solid structure. Regardless of whether one, two, or three spatial dimensions are involved, the hybrid adjoint method proceeds in the same fashion.

3.3.5 Summary and Highlights of the Hybrid Adjoint Method

The continuum-discrete hybrid adjoint approach presented here is convenient for the case when performance measure sensitivity is required with respect to a large number of design variables, whereas direct formulation of CSA suffers from the limitation of being efficient only for small number of design variables. This approach is called a hybrid adjoint formulation because it starts with CSA for state variable derivatives, but the final adjoint equations are based on the discretized CSEs and a discrete performance measure. The hybrid adjoint approach enjoys the following benefits.

- The sensitivity boundary conditions are imposed on the state variable derivatives and are accounted for in the reduced discretized CSEs. This is done before introduction of the adjoint vector $\{\boldsymbol{\lambda}\}$, so that there are no boundary conditions to derive for the adjoint variable vector.

- SGR is performed for points only on the boundary surfaces and at locations contrib-

uting to the performance metric through the vector $\{\boldsymbol{z}\}$ as seen from Eq. (3.23), no matter how many design variables.

- Only one linear system (3.21) is solved to obtain the value of the adjoint vector $\{\boldsymbol{\lambda}\}$ for each performance measure.
- Once $\{\boldsymbol{\lambda}\}$ is obtained, the inner product calculation in Eq. (3.22) can be done inexpensively for as many design variables as required.
- The derived formulation can be easily extended from 1D to 2D and 3D structural applications.

The hybrid adjoint was initially demonstrated for a 1D axial bar (Kulkarni et al., 2016), and is presented in the next section. In the following sections, we present application of the hybrid adjoint method to 2D and 3D applications.

3.4 Axial Bar

Wickert et al. (Wickert et al., 2010) presented the following tutorial problem of finding shape sensitivity of the axial displacement of a bar using CSA. This simple one-dimensional problem illustrates the process of CSA for two different shape parameterizations. In this section, the problem will be solved using the direct CSA formulation and the hybrid adjoint formulation, and solutions will be compared with the exact solution.

3.4.1 Problem Description

The derivatives of axial displacement and axial stress in a bar are sought with respect to the length of the bar. The bar with the applied linearly varying load and clamped boundary conditions is shown in Figure 3.3.

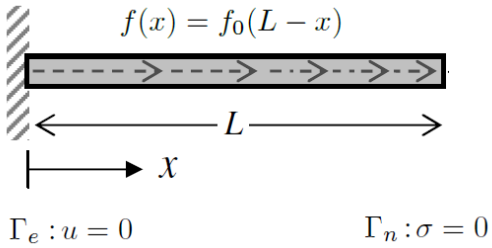

Figure 3.3: Elastic bar with axial load

3.4.1.1 Design Velocity

The design variable is the length of the bar, L; however, a designer may parameterize the domain $\Omega : \ x \in [0,\ L]$ in a number of ways. Two such parameterizations are shown in Figure 3.4. The difference in these parameterizations can be realized from the design velocities for the two cases, as explained next. The two design variables associated with the two shape parameterizations will be used in Section 3.4.4 to illustrate how the hybrid adjoint is applied to solve a single linear system for multiple shape design variables.

The change in shape of a structure can be expressed by a mapping function $\boldsymbol{T}$ from an initial structural geometry Ω to a new structural geometry Ω_b. Here b denotes a design variable that results in a change of shape. If $\boldsymbol{x}$ and $\boldsymbol{x}_b$ represent the spatial coordinates in the initial geometry Ω and the perturbed geometry Ω_b, respectively, then the mapping $\boldsymbol{T} : \Omega \to \Omega_b$ is given by

$$\boldsymbol{x}_b = \boldsymbol{T}(\boldsymbol{x}, b).$$

A particular choice of the mapping $\boldsymbol{T}$ will determine the transformation of shape from Ω to Ω_b. Parameterizations 1 and 2 shown in Figure 3.4 correspond to two such mappings. The process of shape change through a mapping is the concept of design velocity, also known as geometric sensitivity. Following Choi and Kim (Choi and Kim, 2005), design velocity $\boldsymbol{\mathcal{V}}$ is defined as the rate of change of the transformation $\boldsymbol{T}$ with respect the design variable b.

$$\boldsymbol{\mathcal{V}}(x, b) = \frac{\partial \boldsymbol{x}_b}{\partial b} = \frac{\partial \boldsymbol{T}}{\partial b}. \tag{3.25}$$

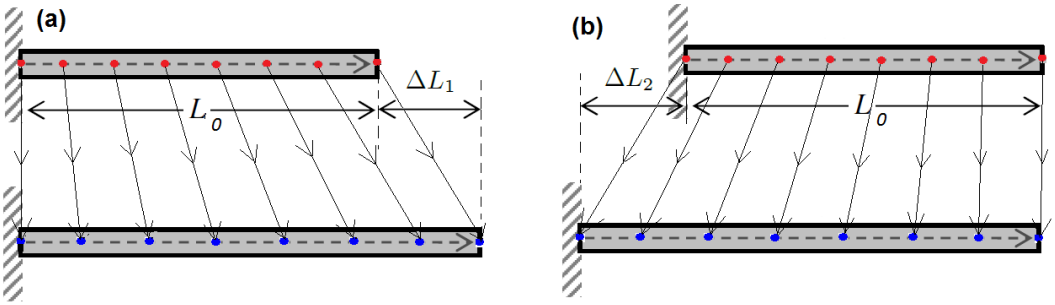

Figure 3.4: Different parametric representations of the elastic bar: (a): Parameterization 1: Material points move to the right, (b): Parameterization 2: Material points move to the left.

Next the design velocities for parameterizations 1 and 2 are derived. The first parameterization can be interpreted as increasing the length from an initial value L_0 by pulling the free end to the right by an amount $b_1 = \Delta L_1$, while maintaining the location of the clamped end. In this case, all the material points move to the right as shown in Figure 3.4. Let $\xi = x/L$ be a non-dimensional parameter such that $\xi = 0$ corresponds to the left end and $\xi = 1$ corresponds to the right end. Then the transformation for the first parameterization is given by

$$\boldsymbol{x}_{b_1} = \xi \left(L_0 + b_1 \right) \hat{\boldsymbol{\imath}}.$$

Here, $\hat{\boldsymbol{\imath}}$ represents unit vector in the horizontal direction of the Cartesian coordinate system. According to the definition of design velocity given by Equation (3.25), the design velocity is

$$\boldsymbol{\mathcal{V}}_{\mathbf{1}} = \frac{\partial \boldsymbol{x}_{b_1}}{\partial b_1} = \xi \hat{\boldsymbol{\imath}} = \frac{x}{L} \hat{\boldsymbol{\imath}}. \tag{3.26}$$

In the second parameterization, the length of the bar is extended by pulling the clamped end to the left by an amount $b_2 = \Delta L_2$ leaving the material point at the free end in place. The transformation in this case is given by

$$\boldsymbol{x}_{b_2} = \left[\xi \left(L_0 + b_2 \right) - b_2 \right] \hat{\boldsymbol{\imath}}.$$

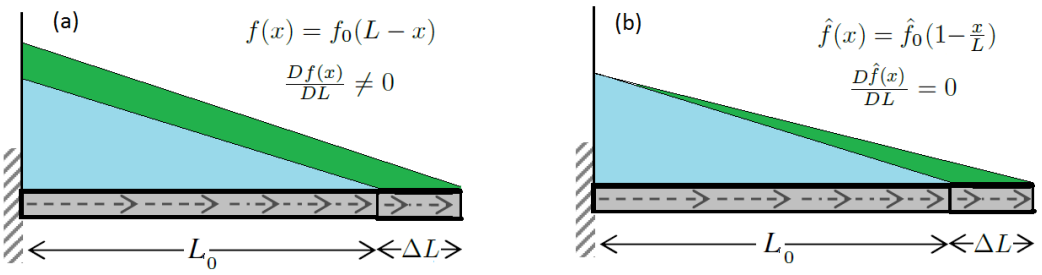

Figure 3.5: Types of linearly varying loads

Then, according to Equation (3.25), the design velocity for parameterization 2 is

$$\boldsymbol{\mathcal{V}_2} = \frac{\partial \boldsymbol{x}_{b_2}}{\partial b_2} = (\xi - 1)\,\hat{\boldsymbol{\imath}} = \left(\frac{x}{L} - 1\right)\hat{\boldsymbol{\imath}}. \tag{3.27}$$

The two parameterizations shown above are just two different parameterizations of the same shape design variable viz. length of the bar, L. This difference is characterized by the design velocities in each case. Hence, the material derivatives $(D\phi/DL)$ of any response variable ϕ will be the same for both parameterizations, although the local derivatives $(\partial\phi/\partial L)$ will be different.

3.4.1.2 Distributed Axial Loading

The axial bar is subjected to a linearly varying distributed load, highest at the clamped end and zero at the free end. The load can be represented in either of the two ways shown in Figure 3.5. The difference in the loads: (a) $f(x) = f_0(L - x)$, and (b) $\hat{f}(x) = \hat{f}_0(1 - x/L)$ is realized when the initial length L_0 is increased by an amount ΔL. In case (a), the load intensity at each material point increases with an increase in the length. This is mathematically expressed as $\dot{f} = Df(x)/DL \neq 0$. In case (b), the load intensity at each material point remains the same as the length increases. This is expressed as $\dot{\hat{f}} = D\hat{f}(x)/DL = 0$. Wickert et al. Wickert et al. (2010) used a notation similar to (a) where they assumed $f_0 = 1$. We have modified that expression to included the factor f_0 which has dimensions of load per unit length.

3.4.2 Analytical Sensitivity by Differentiating Exact Solution

The governing equations for the axial stress and displacement in a bar with a constant cross sectional area A and an elastic modulus E are

$$A\sigma_{,x} + f = 0, \tag{3.28}$$

$$Eu_{,x} - \sigma = 0, \tag{3.29}$$

subjected to the following boundary conditions

$$\text{Essential boundary condition at boundary } x = 0,\ \Gamma_e \ : \ u = 0, \tag{3.30}$$

$$\text{Nonessential boundary condition at boundary } x = L,\ \Gamma_n \ : \ \sigma = 0. \tag{3.31}$$

Here, $(.)_{,x} = \frac{\partial}{\partial x}(.)$. Loading is linearly varying as shown in Figure 3.5(a), $f(x) = f_0(L - x)$. The exact solutions for the axial stress σ and axial displacement u are

$$\sigma = \frac{f_0}{A}\left\{\frac{x^2}{2} - Lx + \frac{L^2}{2}\right\}, \tag{3.32}$$

$$u = \frac{f_0}{AE}\left\{\frac{x^3}{6} - \frac{Lx^2}{2} + \frac{L^2 x}{2}\right\}. \tag{3.33}$$

A finite element displacement solution with 50 linear elements and the exact solution are shown in Figure 3.6.

The exact local derivatives can be determined by partial differentiation of the above equations.

$$u' = \frac{\partial u}{\partial L} = \frac{f_0}{AE}\left(-\frac{x^2}{2} + Lx\right) \tag{3.34}$$

$$\sigma' = \frac{\partial \sigma}{\partial L} = \frac{f_0}{A}\big(-x + L\big) \tag{3.35}$$

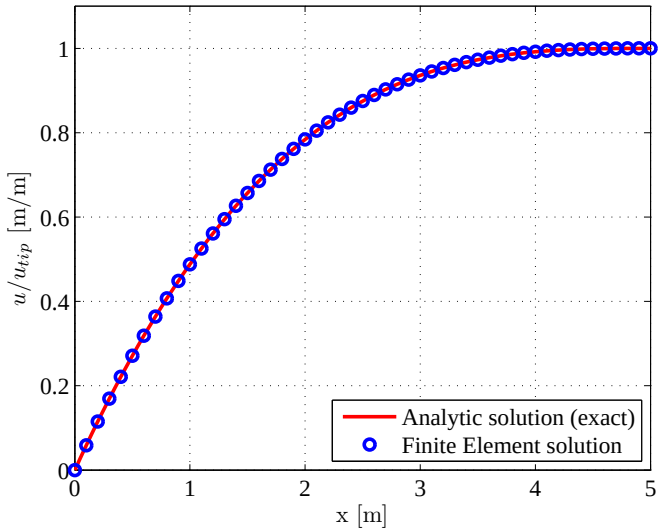

Figure 3.6: Exact and finite element solutions to the axial bar problem.

Therefore, the exact total derivatives at the tip ($x = L$) are

$$\dot{u}|_{x=L} = \left.\frac{Du}{DL}\right|_{x=L} = \frac{\partial}{\partial L}\left(u|_{x=L}\right) = \frac{\partial}{\partial L}\left(\frac{f_0 L^3}{6AE}\right) = \frac{f_0 L^2}{2AE}, \tag{3.36}$$

$$\dot{\sigma}|_{x=L} = \left.\frac{D\sigma}{DL}\right|_{x=L} = \frac{\partial}{\partial L}\left(\sigma|_{x=L}\right) = \frac{\partial}{\partial L}\Big(0\Big) = 0. \tag{3.37}$$

The goal of the current exercise is to obtain the local derivative $\partial u/\partial L$ and total derivatives Du/DL using direct CSA and hybrid adjoint CSA methods for the two parameterizations described in Section 3.4.1.1.

3.4.3 Analytic Solution Using Direct Local CSA

3.4.3.1 Parameterization 1

The local CSEs are obtained by partial differentiation of the governing equations (3.28) and (3.29).

$$A\sigma'_{,x} + f' = 0 \tag{3.38}$$

$$Eu'_{,x} - \sigma' = 0 \tag{3.39}$$

Next, the load f' for the current parameterization is obtained as follows.

$$f' = \dot{f} - \boldsymbol{\nabla}_{\boldsymbol{x}} f . \boldsymbol{\mathcal{V}}_{\mathbf{1}}$$

The material derivative of the applied load, $\dot{f}$, and the spatial gradient of the load, $\boldsymbol{\nabla}_{\boldsymbol{x}} f$, do not depend on the parameterization.

$$\dot{f} = \frac{D}{DL}(f) = \frac{D}{DL}(f_0(L-x)) = f_0 \frac{D}{DL}(L - \xi L) = f_0(1-\xi) \tag{3.40}$$

$$\boldsymbol{\nabla}_{\boldsymbol{x}} f = \frac{\partial f}{\partial x}\hat{\boldsymbol{\imath}} = -f_0 \hat{\boldsymbol{\imath}} \tag{3.41}$$

Therefore, the load term f' in the CSEs changes with the parameterization according to the corresponding design velocity. The design velocity for the first parameterization was derived in Equation (3.26).

$$f' = f_0(1-\xi) - (-f_0)(\xi) = f_0 \tag{3.42}$$

The sensitivity boundary conditions require spatial gradients of displacement and stress, which can be obtained analytically from the exact solution, Equations (3.32) and (3.33), as:

$$\boldsymbol{\nabla}_{\boldsymbol{x}} u = \frac{\partial u}{\partial x}\hat{\boldsymbol{\imath}} = \frac{f_0}{AE}\left\{\frac{x^2}{2} - Lx + \frac{L^2}{2}\right\}\hat{\boldsymbol{\imath}}, \tag{3.43}$$

$$\boldsymbol{\nabla}_{\boldsymbol{x}} \sigma = \frac{\partial \sigma}{\partial x}\hat{\boldsymbol{\imath}} = \frac{f_0}{A}\{x - L\}\hat{\boldsymbol{\imath}}. \tag{3.44}$$

Thus, the CSE boundary conditions are

$$u'|_{x=0} = 0 - \boldsymbol{\nabla}_{\boldsymbol{x}} u . \boldsymbol{\mathcal{V}}|_{x=0 \, or \, \xi=0} = -\left(\frac{f_0 L^2}{2AE}\right)(0) = 0, \tag{3.45}$$

$$\sigma'|_{x=L} = 0 - \boldsymbol{\nabla}_{\boldsymbol{x}} \sigma . \boldsymbol{\mathcal{V}}|_{x=L \, or \, \xi=1} = -(0)(1) = 0. \tag{3.46}$$

To calculate local derivatives, Equations (3.38) and (3.39) may be solved subject to the

loading (3.42), and the boundary conditions (3.45) and (3.46). Parameterization 1 results in homogeneous essential boundary condition (3.45). Consequently, after discretization, $\left\{\boldsymbol{u_1'}\right\} = 0$ in Equation (3.12), and $\left\{\boldsymbol{F'_{local}}\right\} = \left\{\boldsymbol{F_2'}\right\}$ in Equation (3.14). Also $\left\{\boldsymbol{F_2'}\right\} \neq 0$ according to Equation (3.42). Solving the CSEs (3.38) and (3.39) by simple integration, we obtain local derivatives

$$\sigma' = \frac{f_0}{A}\left\{-x + L\right\}, \tag{3.47}$$

$$u' = \frac{f_0}{AE}\left\{-\frac{x^2}{2} + Lx\right\}. \tag{3.48}$$

The total derivatives are obtained by adding the convective terms to the local derivatives.

$$\dot{u} = u' + \boldsymbol{\nabla_x} u.\boldsymbol{\mathcal{V}} \tag{3.49}$$

$$\dot{\sigma} = \sigma' + \boldsymbol{\nabla_x}\sigma.\boldsymbol{\mathcal{V}} \tag{3.50}$$

$$\therefore \dot{u} = \frac{f_0}{AE}\left\{-\frac{x^2}{2} + Lx\right\} + \frac{f_0}{AE}\left\{\frac{x^2}{2} - Lx + \frac{L^2}{2}\right\}\frac{x}{L} = \frac{f_0}{AE}\left\{\frac{x^3}{2L} - \frac{3x^2}{2} + \frac{3Lx}{2}\right\} \tag{3.51}$$

$$\therefore \dot{\sigma} = \frac{f_0}{A}\left\{-x + L\right\} + \frac{f_0}{A}\left\{x - L\right\}\frac{x}{L} = \frac{f_0}{A}\left\{\frac{x^2}{L} - 2x + L\right\} \tag{3.52}$$

Thus, evaluation of Equations (3.51) and (3.52) at the tip, $x = L$, matches the exact derivatives given by Equations (3.36) and (3.37), determined by direct differentiation of the analytic analysis solution in Section 3.4.2, and given in the paper by Wicket et al. (Wickert et al., 2010)

$$\begin{aligned} \dot{u}|_{x=L} &= \frac{f_0 L^2}{2AE} \\ \dot{\sigma}|_{x=L} &= 0. \end{aligned}$$

3.4.3.2 Parameterization 2

The expressions of $\dot{f}$ and $\boldsymbol{\nabla_x} f$ will not change because the second parameterization is just a new parameterization for the same design variable L. Hence, from Equations (3.40) and (3.41), and with the design velocity derived in Equation (3.27), we obtain load f' for the current parameterization.

$$f' = \dot{f} - \boldsymbol{\nabla_x} f . \boldsymbol{\mathcal{V}_2} = f_0 (1 - \xi) - (-f_0)(\xi - 1) = 0 \tag{3.53}$$

The boundary conditions for the CSEs for parameterization 2 are given by

$$u'|_{x=0} = 0 - \boldsymbol{\nabla_x} u . \boldsymbol{\mathcal{V}}|_{x=0 \, or \, \xi=0} = -\left(\frac{f_0 L^2}{2AE}\right)(-1) = \frac{f_0 L^2}{2AE}, \tag{3.54}$$

$$\sigma'|_{x=L} = 0 - \boldsymbol{\nabla_x} \sigma . \boldsymbol{\mathcal{V}}|_{x=L \, or \, \xi=1} = -(0)(0) = 0. \tag{3.55}$$

Unlike parameterization 1, the essential boundary condition (3.54) is now non-homogeneous. The following aspects in parameterization 2 are noteworthy:

- The load term in the CSEs is zero, i.e., $f' = 0$. When the CSEs are discretized, this results in $\{\boldsymbol{F'}\} = 0$ in Equation (3.11), or equivalently $\left\{\boldsymbol{F'_1}\right\} = \left\{\boldsymbol{F'_2}\right\} = 0$.

- The essential boundary condition for the primary analysis ($u|_0 = 0$) is converted to a non-homogeneous boundary condition for the CSEs ($u'|_0 = \frac{f_0 L^2}{2AE}$). This translates to $\left\{\boldsymbol{u'_1}\right\} \neq 0$ in Equation (3.12).

- Therefore, the load term $\left\{\boldsymbol{F'_{local}}\right\}$ in the reduced form of the discretized CSEs (3.14)

is nonzero.

To obtain local derivatives, we solve CSEs (3.38) and (3.39) subject to the loading (3.53), and the boundary conditions (3.54) and (3.55) by simple integration to obtain the local derivatives,

$$\sigma' = 0, \tag{3.56}$$

$$u' = \frac{f_0 L^2}{2AE}. \tag{3.57}$$

The above equations differ Equations (3.47–3.48) for parameterization 1. Nevertheless, the material derivatives recovered by adding the convective term are the same as given by (3.51) and (3.52).

$$\dot{u} = \frac{f_0 L^2}{2AE} + \frac{f_0}{AE}\left\{\frac{x^2}{2} - Lx + \frac{L^2}{2}\right\}\left(\frac{x}{L} - 1\right) = \frac{f_0}{AE}\left\{\frac{x^3}{2L} - \frac{3x^2}{2} + \frac{3Lx}{2}\right\} \tag{3.58}$$

$$\dot{\sigma} = 0 + \frac{f_0}{A}\{x - L\}\left(\frac{x}{L} - 1\right) = \frac{f_0}{A}\left\{\frac{x^2}{L} - 2x + L\right\} \tag{3.59}$$

As expected, these total derivatives match the exact derivatives calculated in Section 3.4.2 and the total derivatives calculated for parameterization 1 in Section 3.4.3.1 and in the paper by Wicket et al. (Wickert et al., 2010). A graphical comparison of these results is shown in the next section.

3.4.4 Solution Using Hybrid Adjoint Formulation

Next, the local and total derivatives are obtained using the hybrid adjoint formulation. A finite element model consisting of 50 linear elements was used. The two design variables are: (1) length L with parameterization 1, and (2) length L with parameterization 2. As discussed in Section 3.3, calculation of the derivative of tip displacement at a particular location

involves solving the linear system (3.21) only once to get the adjoint vector $\{\boldsymbol{\lambda}\}$. The values of $\left\{\boldsymbol{F}'_{\mathbf{local}}\right\}$ are obtained for the two design variables, and the local derivatives are calculated by the inexpensive inner product in Equation (3.22). As against this, the direct CSA method requires solving one linear system (3.14) for local derivatives with respect to each design variable, which is certainly not preferred when the number of design variables outnumber the performance measures. It is obvious that the direct CSA method is more expensive when the problem involves many design variables. In the current example, derivatives are obtained at five locations $x = 0.1$, 1.1, 2.1, 3.1, and 4.1 using the hybrid adjoint method.

The local derivatives obtained by direct CSA and hybrid adjoint method are shown in Figure 3.7. The local derivatives differ for parameterizations 1 and 2, as expected. The derivatives computed with the hybrid adjoint match with those computed by direct CSA at the five locations. The convective term $\boldsymbol{\nabla}_{\boldsymbol{x}} u.\boldsymbol{\mathcal{V}}$ for the two parameterizations is plotted in Figure 3.8. The difference in the convective term is due to different design velocities in the two cases. Analytic spatial gradients given by Equations (3.43) and (3.44) were used to compute the convective term in this example. These could instead be approximated from the computed finite element solutions for axial displacement and stress. Cross and Canfield (Cross and Canfield, 2014) have proposed the spatial gradient reconstruction (SGR) method to approximate spatial gradients. This method leads to a better estimate of the spatial gradients than that obtained by the use of element shape functions. In fact it also provides a means for nonintrusive implementation of CSA (Figure 2.3).

In the above example, there are five performance measures viz. the displacements at the five selected locations. Hence, we will solve for five adjoint vectors according to Equation (3.21), each corresponding to the performance measure at that particular location. These adjoint vectors are plotted in Figure 3.9. The adjoint vectors are normalized as $\bar{\boldsymbol{\lambda}} = \boldsymbol{\lambda}/\left(L/AE\right)$. The inner product $\{\boldsymbol{\lambda}\}^T \left\{\boldsymbol{F}'_{\mathbf{local}}\right\}$ yields the local derivative of displacement at these locations, as given by Equation (3.22).

Addition of the convective term to the local derivatives results in the total derivatives,

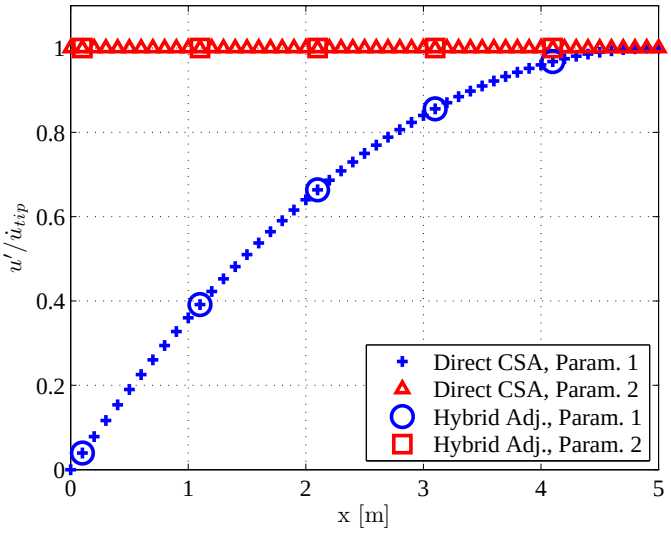

Figure 3.7: Non-dimensional local derivative of the axial bar displacement with respect to its length.

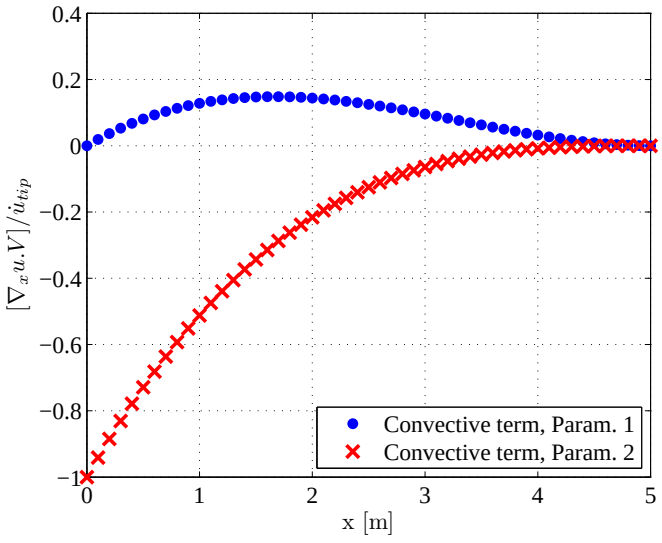

Figure 3.8: Non-dimensional convective term for parameterization 1 and parameterization 2.

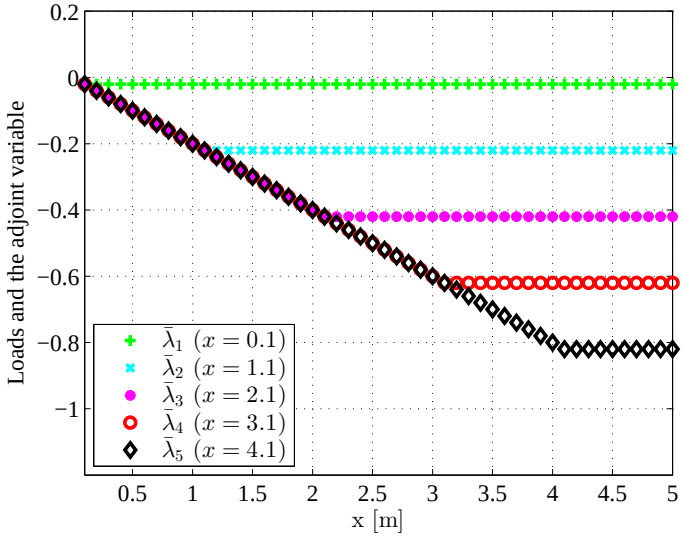

Figure 3.9: Discrete adjoint variable vectors.

according to Equation (2.3) for state variables and according to Equation (3.23) for a performance measure by the hybrid adjoint approach. The total derivatives obtained from direct CSA and hybrid adjoint method are shown in Figure 3.10 along with the exact derivatives. All the five lines in this figure overlap, which means that: (a) the total derivatives obtained for parameterizations 1 and 2 match with the exact values, and (b) the hybrid adjoint method yields corresponding sensitivity values at the five locations.

3.5 Cantilever Beam with Solid Elements (Direct Formulation)

3.5.1 Problem Description

A cantilever beam modelled with trilinear iso-parametric hexagonal solid finite elements is shown in Figure 3.11. The details of the finite element model are given in Table 3.1. The beam consists of in all 80 CHEXA NASTRAN elements and 165 nodes. The task is to obtain the material derivatives of transverse displacement of the beam and stress near

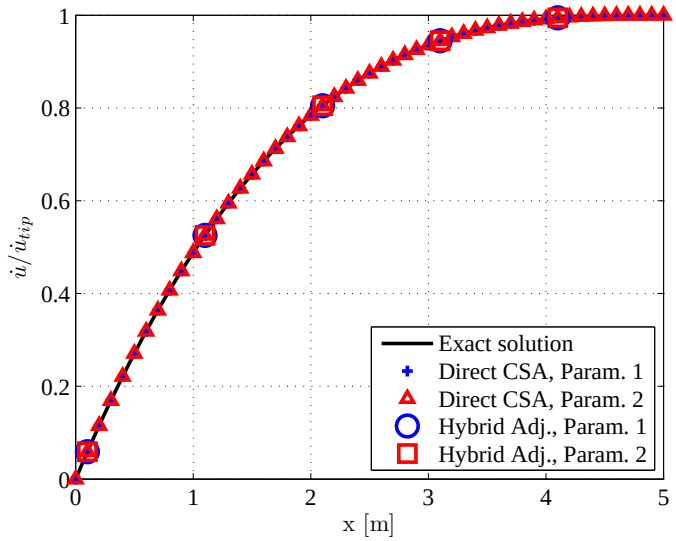

Figure 3.10: Non-dimensional total derivatives of the axial bar displacement with respect to its length.

the root with respect to six shape design variables $b_i \in \{\boldsymbol{b}\} = [b_1, b_2, b_3, b_4, b_5, b_6]^T$. This problem is adopted from the one defined in MSC NASTRAN Design Optimization Users Guide (Chapter 6, Analytic Boundary Shapes) (Nastran, 2010). The point load applied at the tip was modified to a surface traction with the same total load magnitude, and the constraint on the left-side of the beam was modified from a clamped condition to essential BCs as given by Augarde and Deeks (Augarde and Deeks, 2008). Also, the Poisson's ratio was set to zero. Although an analytic solution is not available for this problem, with these modifications, the finite element solution obtained from NASTRAN could be compared with a 2D Airy stress solution. The shape design variables $\{\boldsymbol{b}\}$ and their associated design velocities are described next.

The six shape design variables (Nastran, 2010) $\{\boldsymbol{b}\}$ chosen to demonstrate results for the hybrid adjoint formulation of CSA are shown in Figure 3.12. In these variables, the shape of the beam changes due to change in either the bottom ($z = 0$) or the top ($z = h$) surface. This shape change could be parameterized in different ways. In the current parameterization the components $\mathcal{V}_x$ and $\mathcal{V}_y$ of the design velocity are zero, while $\mathcal{V}_z$ is nonzero. Design variables b_1 and b_2 change the height of the beam uniformly along its length by displacing the bottom

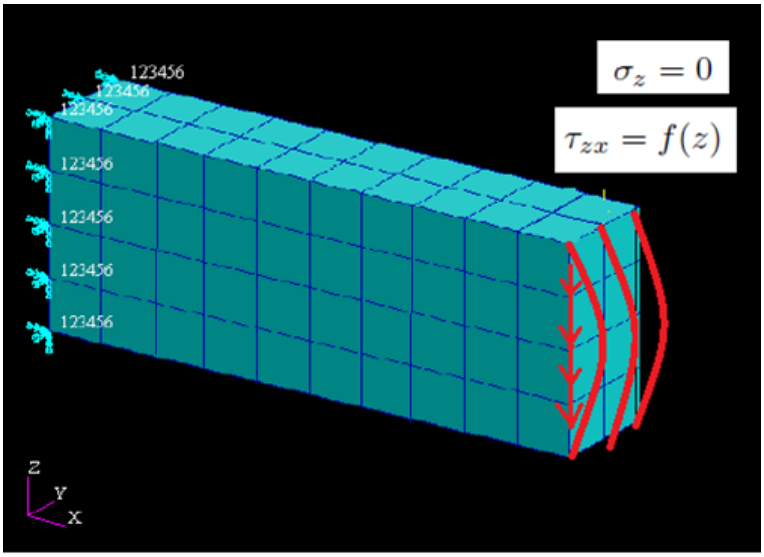

Figure 3.11: Cantilever beam modelled with solid elements and subjected to surface traction.

Table 3.1: Details of the cantilever beam finite element model for the coarsest grid.

Variable	Symbol	Value
Length, along X	L	10 in
Width, along Y	w	2 in
Height, along Z	h	4 in
Young's modulus	E	2.068×10^5 psi
Poisson's ratio	ν	0.0
Number of elements along X	n_x	10
Number of elements along Y	n_y	2
Number of elements along Z	n_z	4

and top surface, respectively. Design velocities for these variables are given by

$$\mathcal{V}_{z_{b_1}} = \left(1 - \frac{z}{h}\right), \qquad \mathcal{V}_{z_{b_2}} = \left(-\frac{z}{h}\right). \tag{3.60}$$

Design variables b_3 and b_4 provide a linear taper from root ($x = 0$) to tip ($x = L$) by displacing the bottom and top surface, respectively, while constraining translation at the root but allowing rotation of the surface. Design velocities for these variables are given by

$$\mathcal{V}_{z_{b_3}} = \left(\frac{x}{L}\right)\left(1 - \frac{z}{h}\right), \qquad \mathcal{V}_{z_{b_4}} = \left(\frac{x}{L}\right)\left(-\frac{z}{h}\right). \tag{3.61}$$

Design variables b_5 and b_6 provide a cubic taper from root to tip by displacing the bottom and top surface, respectively, while constraining translation and rotation at the root. Design velocities for these variables are given by

$$\mathcal{V}_{z_{b_5}} = \frac{1}{2}\left(3 - \frac{x}{L}\right)\left(\frac{x}{L}\right)^2\left(1 - \frac{z}{h}\right), \qquad \mathcal{V}_{z_{b_6}} = \frac{1}{2}\left(3 - \frac{x}{L}\right)\left(\frac{x}{L}\right)^2\left(-\frac{z}{h}\right). \tag{3.62}$$

3.5.2 Primary Analysis

The partial differential equations (2.1) governing a three-dimensional (3D) structural displacement response $\boldsymbol{u} = \{u_x,\, u_y,\, u_z\}$, based on linear elasticity are

$$[\boldsymbol{\partial}]\,[\mathcal{D}]\,[\boldsymbol{\partial}]^T\,\boldsymbol{u} = \boldsymbol{f}, \tag{3.63}$$

where $\boldsymbol{f} = \{f_x,\, f_y,\, f_z\}^T$ are the applied body forces at a point in the domain Ω, $[\mathcal{D}]$ is the constitutive matrix, and $[\boldsymbol{\partial}]$ is the operator matrix given by

$$[\boldsymbol{\partial}] = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial z} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}. \tag{3.64}$$

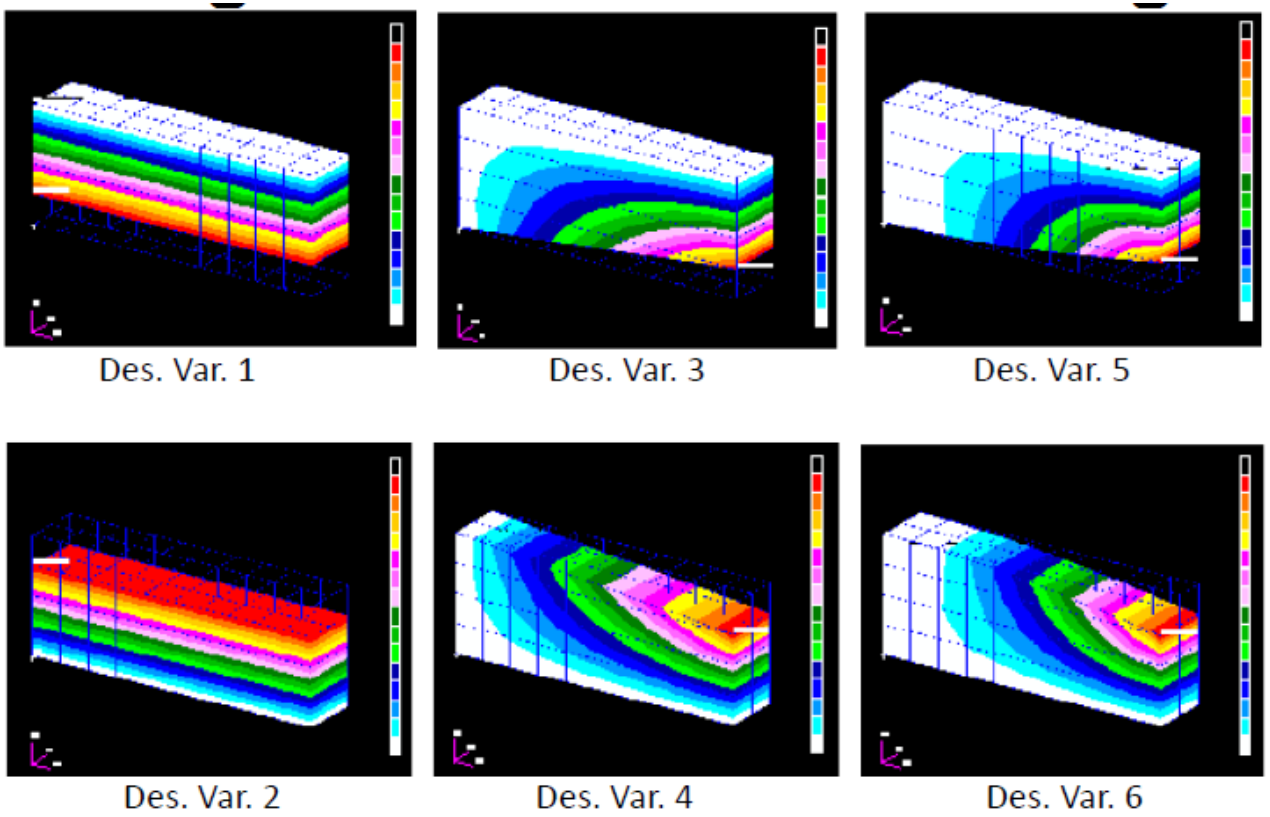

Figure 3.12: Design velocities, $\mathcal{V}_z$ for each design variable of the solid cantilever beam. The design velocities $\mathcal{V}_x$ and $\mathcal{V}_y$ are zero for all design variables.

The stresses $\boldsymbol{\sigma} = \{\sigma_x,\, \sigma_y,\, \sigma_z,\, \tau_{xy},\, \tau_{yz},\, \tau_{zx}\}^T$ are related to the strains $\boldsymbol{\epsilon} = \{\epsilon_x,\, \epsilon_y,\, \epsilon_z,\, \gamma_{xy},\, \gamma_{yz},\, \gamma_{zx}\}^T$ through the stress-strain relationship $\boldsymbol{\sigma} = [D]\,\boldsymbol{\epsilon}$, and the strains are are related to the displacement through the strain-displacement relationship $\boldsymbol{\epsilon} = [\boldsymbol{\partial}]^T\,\boldsymbol{u}$. Comparing Eq. (3.63) to the general Eq. (2.1), we can derive the PDE operator

$$\boldsymbol{\mathcal{A}}(\boldsymbol{u}, L(\boldsymbol{u})) \equiv L(\boldsymbol{u}) = [\boldsymbol{\partial}]\,\mathcal{D}\,[\boldsymbol{\partial}]^T\,\boldsymbol{u}. \tag{3.65}$$

Essential (geometric) BCs (2.2) are applied at the boundary $\Gamma_e : x = 0$ by prescribing displacements $\bar{\boldsymbol{u}}$

$$\boldsymbol{u}|_{\Gamma_e} = \bar{\boldsymbol{u}}. \tag{3.66}$$

Nonessential (natural) boundary conditions are applied at the boundaries Γ_n by prescribing surface tractions $\bar{\boldsymbol{\Phi}} = \left\{\bar{\Phi}_x,\, \bar{\Phi}_y,\, \bar{\Phi}_z\right\}^T$

$$\boldsymbol{\Phi}|_{\Gamma_n} = \bar{\boldsymbol{\Phi}}. \tag{3.67}$$

The tractions at any point on the boundary surface are related to the stresses at that point by the relation

$$\boldsymbol{\Phi} = \begin{Bmatrix} \Phi_x \\ \Phi_y \\ \Phi_z \end{Bmatrix} = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{zx} \\ \tau_{xy} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{yz} & \sigma_z \end{bmatrix} \begin{Bmatrix} n_x \\ n_y \\ n_z \end{Bmatrix} = [\boldsymbol{\sigma}]\,\boldsymbol{n}, \tag{3.68}$$

where $\boldsymbol{n} = \{n_x,\, n_y, n_z\}^T$ are the direction cosines of a vector normal to the surface and $[\boldsymbol{\sigma}]$ is the stress tensor at that point. Thus, the boundary condition (3.67) can be written in terms of the stress components as

$$\boldsymbol{\Phi}|_{\Gamma_n} = [\boldsymbol{\sigma}]\,\boldsymbol{n}|_{\Gamma_n} = \bar{\boldsymbol{\Phi}}. \tag{3.69}$$

The 3D cantilever beam domain Ω has six faces that make up the boundary Γ. Essential BC

Table 3.2: Natural BC applied on the faces of the 3-D (solid) cantilever beam.

Face of the beam	Traction description	Value, $\bar{\mathbf{\Phi}}$
$x = L$	Parabolic shear	$\bar{\mathbf{\Phi}}_{\text{tip}} = \frac{6P(h-z)z}{bh^3}$
$y = 0$	Traction free	0
$y = w$	Traction free	0
$z = 0$	Traction free	0
$z = h$	Traction free	0

(3.66) is applied at the root face $\Gamma_e : x = 0$, and natural BCs are applied at the remaining five faces which together make up the boundary Γ_n. The values of prescribed traction on each of these five faces is given in Table 3.2. Typically, in displacement based finite element solvers, such as NASTRAN, the traction free natural BCs are not applied explicitly. The displacements on the center line and at the face $x = 0$ (where essential boundary conditions are applied) are plotted in Figures (3.13–3.16). The $z-$transverse displacement of the solid cantilever beam on the center line is shown in Figures (3.13–3.14). The stress distributions are plotted in Figures (3.17–3.20). Note that the axial displacement is negligible. Also, the values of stress τ_{yz} are negligible. The values of σ_z are one order of magnitude smaller than τ_{zx}, and two orders of magnitude smaller than σ_x.

3.5.3 Continuum Sensitivity Analysis

The CSEs for the 2D problem are derived by partial differentiation of Eq. (3.63)

$$[\boldsymbol{\partial}]\,[\mathcal{D}]\,[\boldsymbol{\partial}]^T\,\boldsymbol{u'} = \boldsymbol{f'}. \tag{3.70}$$

Comparing Eq. (3.70) to the general Eq. (2.6), we confirm that for the linear structural analysis considered here, the PDE operator for the structural local derivatives is same as the PDE operator for the structural response, i.e. $\boldsymbol{\mathcal{A}}_b = \boldsymbol{\mathcal{A}}$.

According to Eq. (2.5), the essential (geometric) CSE BCs are obtained by material differentiation of (3.66), and are applied at the boundary $\Gamma_e : x = 0$ by prescribing local

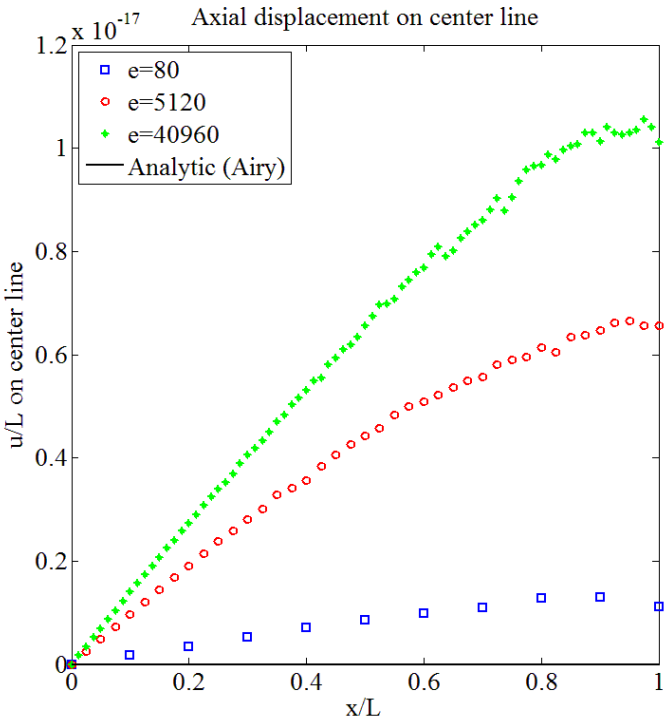

Figure 3.13: Axial displacement of the beam of the solid cantilever beam on the center line, $y = 1$, $z = 2$. The symbol e indicates total number of elements.

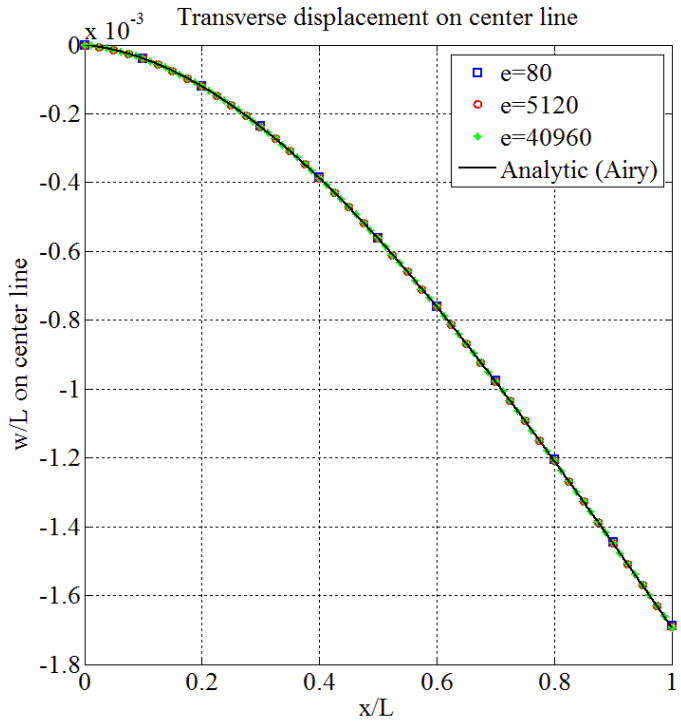

Figure 3.14: Transverse z-displacement of the solid cantilever beam on the center line, $y = 1$, $z = 2$. The symbol e indicates total number of elements.

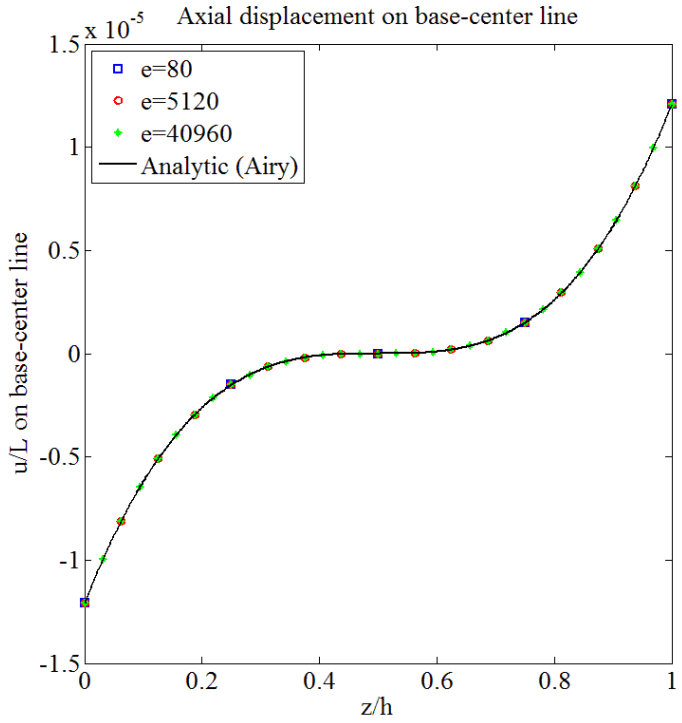

Figure 3.15: Axial displacement of the solid cantilever beam on the face $x = 0$ (where essential boundary conditions are applied). The symbol e indicates total number of elements.

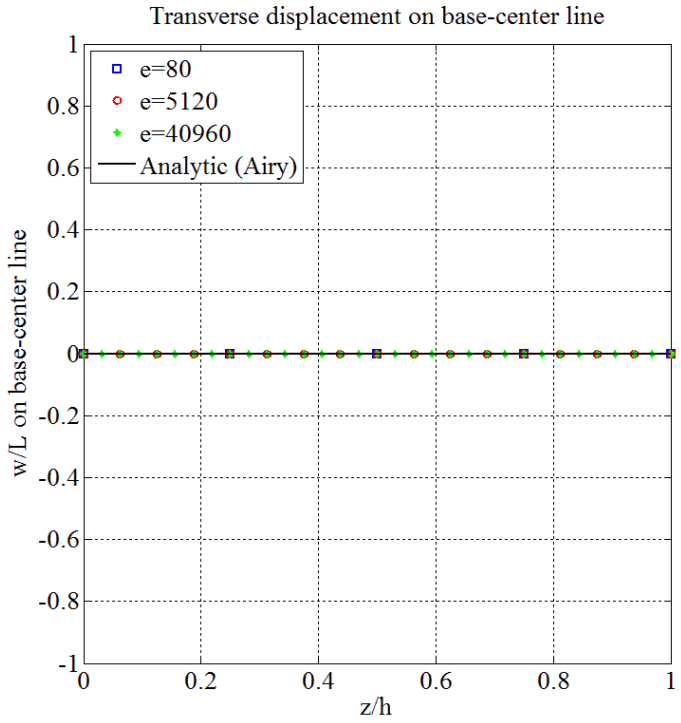

Figure 3.16: Transverse z-displacement of the solid cantilever beam on the face $x = 0$ (where essential boundary conditions are applied). The symbol e indicates total number of elements.

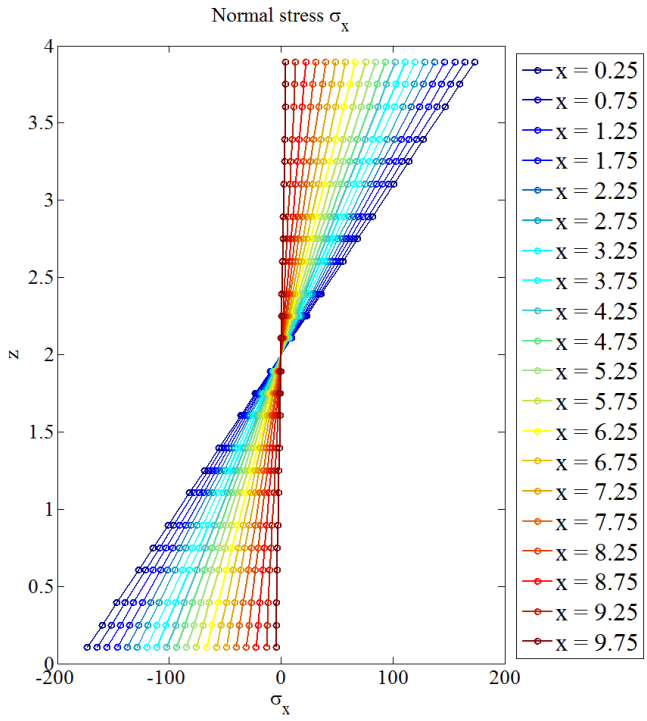

Figure 3.17: Normal stress σ_x distribution in the solid cantilever beam for $e = 640$ elements.

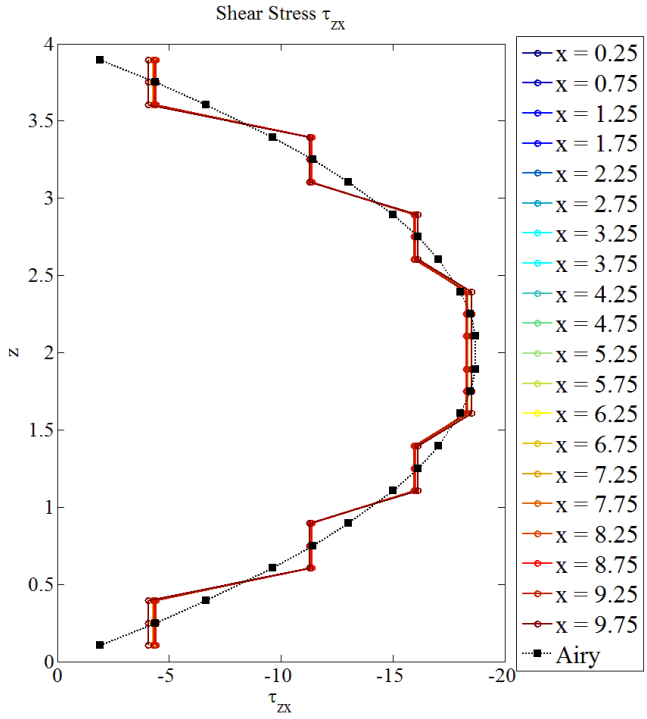

Figure 3.18: Shear stress τ_{zx} distribution in the solid cantilever beam for $e = 640$ elements. The filled marker indicates the analytic values from the Airy stresses.

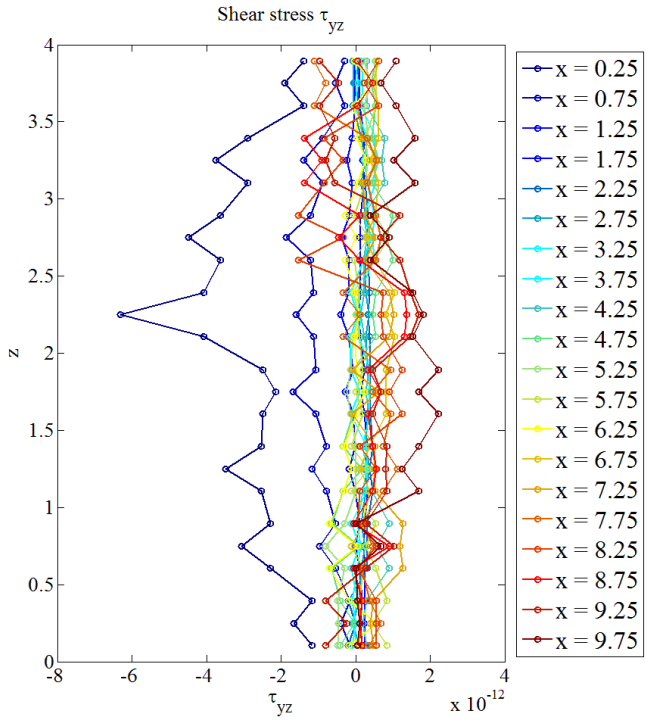

Figure 3.19: Normal stress τ_{yz} distribution in the solid cantilever beam for $e = 640$ elements. Note that the values of stress τ_{yz} are negligible.

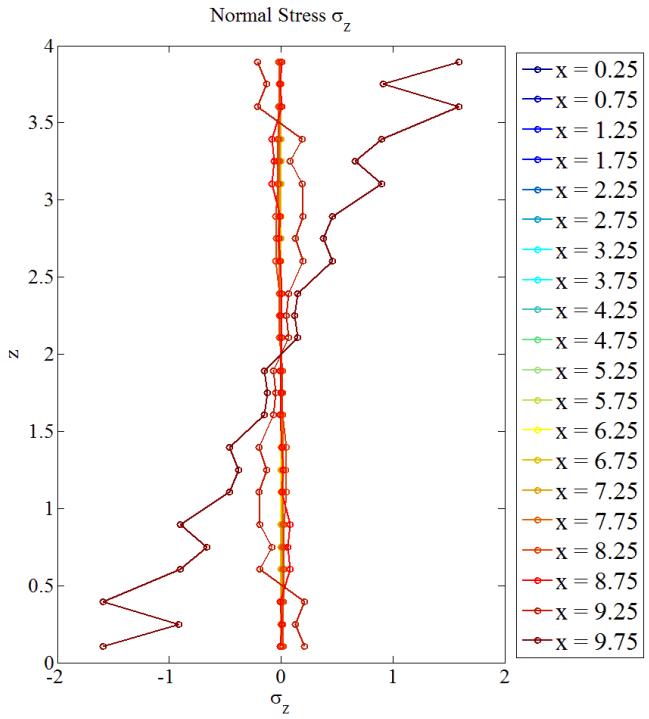

Figure 3.20: Normal stress σ_z distribution in the solid cantilever beam for $e = 640$ elements. Note that the values of σ_z are one order of magnitude smaller than τ_{zx} and two orders of magnitude smaller than σ_x.

displacement derivatives

$$\boldsymbol{u'}|_{\Gamma_e} = \dot{\tilde{\boldsymbol{u}}} - \boldsymbol{\nabla_x}(\boldsymbol{u}) \cdot \boldsymbol{\mathcal{V}}|_{\Gamma_e}. \tag{3.71}$$

This BC leads to a nonzero vector $\left\{\boldsymbol{u'_1}\right\}$ in Eq. (3.15).

Similarly, the non-essential (natural) CSE BCs are obtained by material differentiation of (3.69), and are applied at the boundary Γ_n by prescribing traction values according to

$$\boldsymbol{\Phi'}|_{\Gamma_n} = \dot{\tilde{\boldsymbol{\Phi}}} - \left(\mathcal{V}_x \frac{\partial[\boldsymbol{\sigma}]}{\partial x} + \mathcal{V}_y \frac{\partial[\boldsymbol{\sigma}]}{\partial y} + \mathcal{V}_z \frac{\partial[\boldsymbol{\sigma}]}{\partial z}\right)\boldsymbol{n} - [\boldsymbol{\sigma}]\,\dot{\boldsymbol{n}}\Big|_{\Gamma_n}, \tag{3.72}$$

where

$$\frac{\partial[\boldsymbol{\sigma}]}{\partial x_k} = \frac{\partial}{\partial x_k}\begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{zx} \\ \tau_{xy} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{yz} & \sigma_z \end{bmatrix}, \tag{3.73}$$

for the directional components $x_k = x$, y, z, and $\dot{\boldsymbol{n}}$ is the material derivative of the unit normal. Eq. (3.72) applies to each of the five faces, given in Table 3.2, that together make up the boundary Γ_n. For the current problem, since $\mathcal{V}_x = \mathcal{V}_y = 0$ at all points, BC (3.72) simplifies to

$$\boldsymbol{\Phi'}|_{\Gamma_n} = \dot{\tilde{\boldsymbol{\Phi}}} - \mathcal{V}_z \frac{\partial[\boldsymbol{\sigma}]}{\partial z}\boldsymbol{n} - [\boldsymbol{\sigma}]\,\dot{\boldsymbol{n}}\Big|_{\Gamma_n}. \tag{3.74}$$

Thus, the natural CSE BCs to be applied at the faces of the 3D cantilever beam can be stated as

$$\boldsymbol{\Phi'}|_{x=L} = \dot{\tilde{\boldsymbol{\Phi}}}_{\text{tip}} - \mathcal{V}_z \frac{\partial[\boldsymbol{\sigma}]}{\partial z}\boldsymbol{n} - [\boldsymbol{\sigma}]\,\dot{\boldsymbol{n}}\Big|_{x=L}, \tag{3.75}$$

$$\boldsymbol{\Phi'}|_{y=0} = 0 - \mathcal{V}_z \frac{\partial[\boldsymbol{\sigma}]}{\partial z}\boldsymbol{n} - [\boldsymbol{\sigma}]\,\dot{\boldsymbol{n}}\Big|_{y=0} = 0, \tag{3.76}$$

$$\boldsymbol{\Phi'}|_{y=w} = 0 - \mathcal{V}_z \frac{\partial[\boldsymbol{\sigma}]}{\partial z}\boldsymbol{n} - [\boldsymbol{\sigma}]\,\dot{\boldsymbol{n}}\Big|_{y=w} = 0, \tag{3.77}$$

$$\boldsymbol{\Phi'}|_{z=0} = 0 - \mathcal{V}_z \frac{\partial[\boldsymbol{\sigma}]}{\partial z}\boldsymbol{n} - [\boldsymbol{\sigma}]\,\dot{\boldsymbol{n}}\Big|_{z=0}, \tag{3.78}$$

$$\boldsymbol{\Phi'}|_{z=h} = 0 - \mathcal{V}_z \frac{\partial[\boldsymbol{\sigma}]}{\partial z}\boldsymbol{n} - [\boldsymbol{\sigma}]\,\dot{\boldsymbol{n}}\Big|_{z=h}. \tag{3.79}$$

Eqs. (3.76–3.77) have zero right side because $\frac{\partial[\boldsymbol{\sigma}]}{\partial z}$ and $\dot{\boldsymbol{n}}$ are zero at the faces $y = 0$ and $y = w$. Similar to the primary analysis, these traction free BCs (3.76–3.77) are not explicitly applied during the finite element solution of the CSEs. The BCs (3.75, 3.78, 3.79) lead to a nonzero vector $\left\{\boldsymbol{F}'_{\boldsymbol{n}}\right\}_i$ in Eq. (3.16). Apart from the loads arising from these CSE BCs, there can also be loads due to the dependence of the applied body loads on the shape design variable, leading to nonzero vector $\left\{\boldsymbol{F}'_{\boldsymbol{\Omega}}\right\}_i$ in Eq. (3.16); however, they are zero for the current problem.

The two most influential terms in the CSA BCs (3.75–3.79) are $\dot{\tilde{\boldsymbol{\Phi}}}_{\text{tip}}$ and the convective term $\mathcal{V}_z \frac{\partial[\boldsymbol{\sigma}]}{\partial z}\boldsymbol{n}$ on the top and the bottom surface (wherever the design velocity is the highest). The former is known analytically. The latter involves calculation of the z-spatial gradients of the stress components $\{\tau_{zx}, \tau_{zy}, \sigma_z\}$, of which $\frac{\partial \tau_{zx}}{\partial z}$ has the highest value. These spatial gradients are calculated using SGR with second-order Taylor series and four patch layers, and are shown in Figure 3.21. It is seen that the SGR approximates the spatial gradients very accurately, compared to the analytical values obtained form the Airy stresses.

In theory, if the spatial gradients are accurately known and the linear CSEs are accurately solved, the resulting local derivatives will be accurate. However, this is seldom realized in practice because discretization error is introduced during solution of the differential equations. Since the Airy stress spatial derivatives are known, the best case scenario for CSA would be when the CSE boundary conditions are calculated using the exact spatial gradients. This is used as a limiting test case.

The CSEs were solved in NASTRAN by applying the corresponding loads and BCs in the same way as the primary analysis was done. In the direct formulation of CSA, solution of the CSEs gives the values of the local derivatives $\boldsymbol{u'}$ at all points in the domain. Then, the convective term is added to obtain the total derivatives $\dot{\boldsymbol{u}}$ at locations of interest, according to Eq. (3.5). The total derivatives of transverse displacement on the center line are shown in Figure 3.22. The results labelled "CSA-SGR BC, e=..." were calculated using SGR with second-order Taylor series and four layer patches. The symbol e represents number of

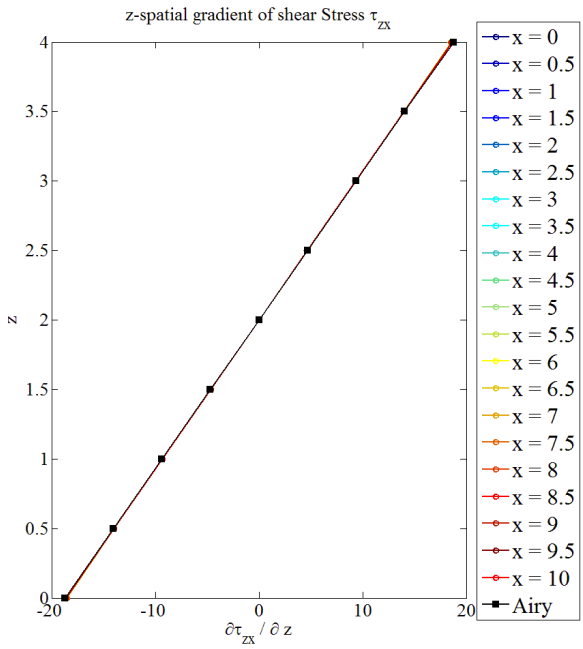

Figure 3.21: Distribution of the spatial gradient of the shear stress, $\frac{\partial \tau_{zx}}{\partial z}$, in the solid cantilever beam with $e = 640$ elements, calculated using SGR with second-order Taylor series and four patch layers. The filled markers indicate the analytic values from the Airy stresses.

elements. The CSA results obtained by using Airy stress spatial gradients to calculate the CSE BCs are labelled "CSA-Airy BC." Analytical solution to the displacement and stress derivatives is only available for the design variables 1 and 2, and are labelled "Analytic (Airy)." It is seen that the displacement derivatives match the analytical solution closely. Although unexpected, the "CSA-Airy BC" result for design variables 1 and 2 is further away from the "Analytic (Airy)" result than the "CSA-SGR BC, e=..." result. This is an intriguing result which will be investigated further.

In the direct formulation of CSA, the local derivatives of stresses are also obtained when NASTRAN is run to solve the CSEs. The stress convective term can be added to these local stress derivatives to obtain the total stress derivatives at locations of interest. The distributions of stress derivatives are shown in Figures (3.23–3.24). The results labelled "CSA-SGR BC" were calculated using SGR with second-order Taylor series and four layer patches. As for displacement derivatives, the stress derivatives also match the analytical

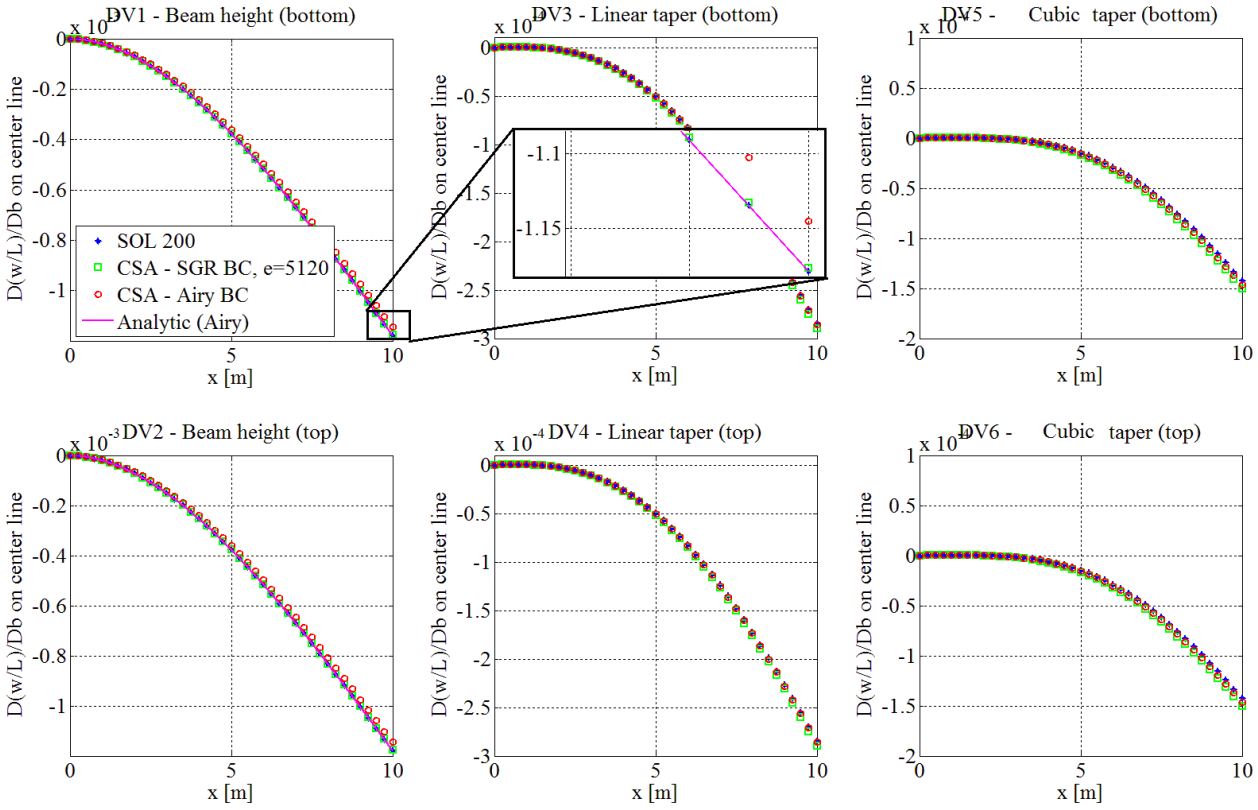

Figure 3.22: Total derivatives of z-transverse displacement of the solid cantilever beam modeled with $e = 5120$ elements. "CSA-SGR BC" results were calculated using SGR with second-order Taylor series and four layer patches.

solution closely.

3.5.4 Grid Convergence Study

A grid convergence study was done to obtain the rate of convergence of the displacement and stress derivatives obtained using CSA. A series of four meshes is used. The coarsest mesh had $NC = 10 \times 2 \times 4 = 80$ brick elements (where the three factors indicate the number of elements in the X, Y and Z directions respectively), while the finest mesh had $NC = 80 \times 16 \times 32 = 40960$ elements. Here NC stands for number of cells or number of elements. The grid refinement was uniform such that the element edge lengths were halved at each refinement. The direct formulation of CSA gives results of displacement and stress

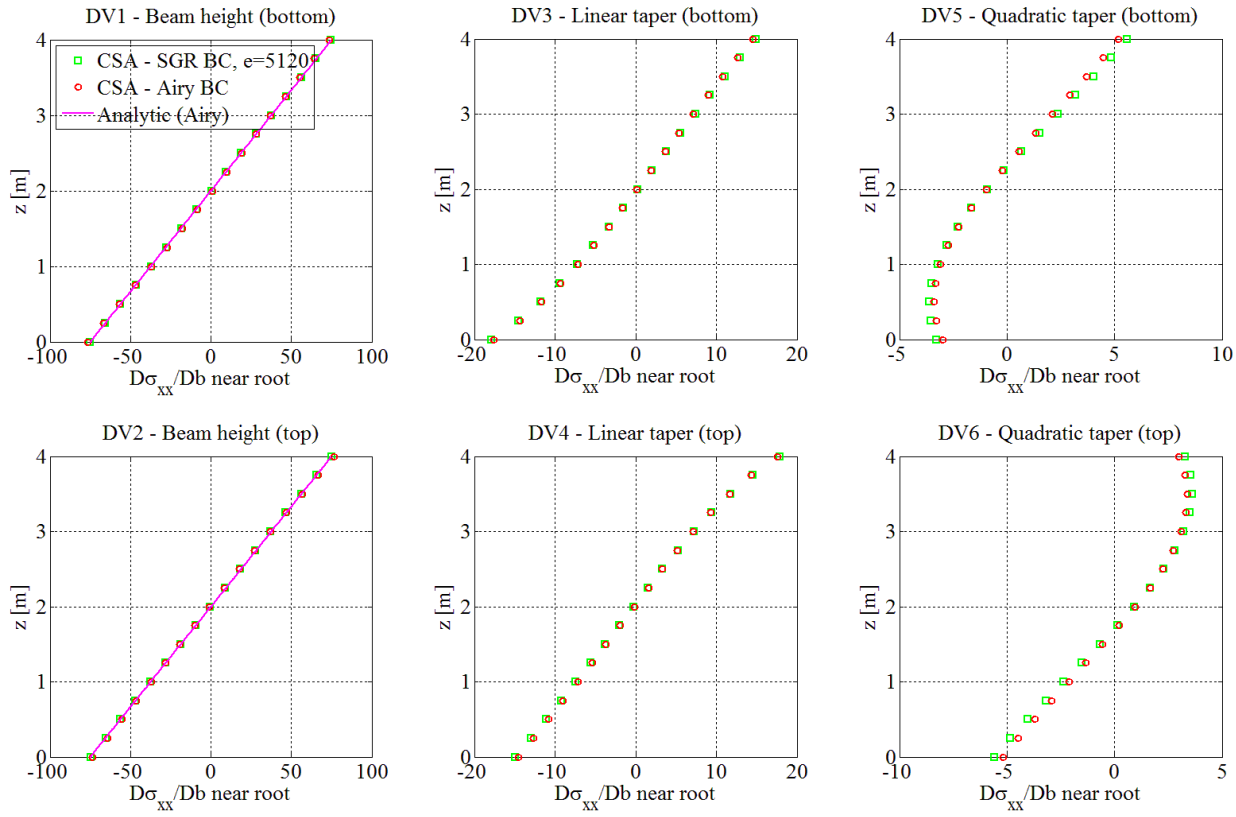

Figure 3.23: Total derivatives of normal stress $\dot{\sigma}_x$ of the solid cantilever beam modeled with $e = 5120$ elements, plotted near the root, on the line $x = 2$, $y = 1$. "CSA-SGR BC" results were calculated using SGR with second-order Taylor series and four layer patches.

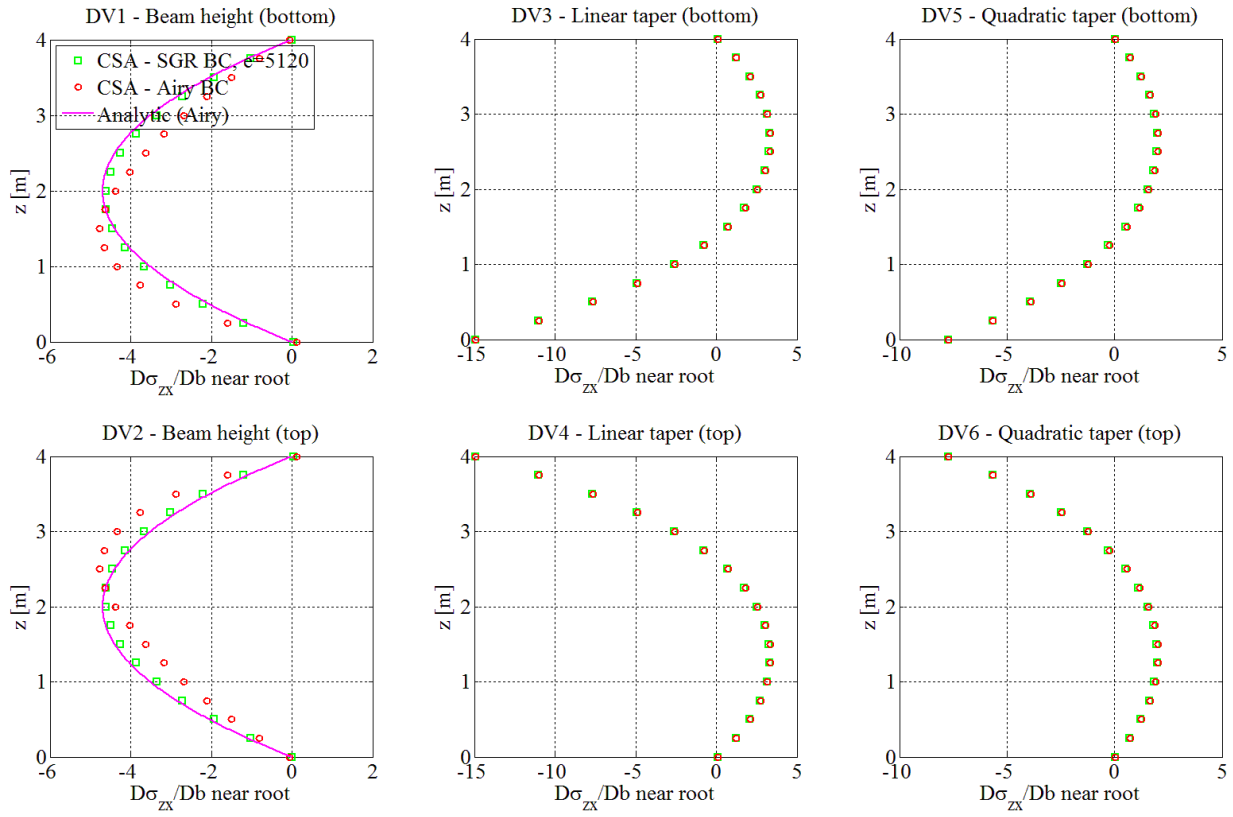

Figure 3.24: Total derivatives of shear stress $\dot{\tau}_{zx}$ of the solid cantilever beam modelled with $e = 5120$ elements, plotted on the line $x = L/2, y = 1$. "CSA-SGR BC" results were calculated using SGR with second-order Taylor series and four layer patches.

derivatives at all points in the domain. These derivatives were obtained with respect to all six design variables described in Section 3.5.1.

As an example, the convergence of displacement,stresses and their derivatives are shown only at particular points. The continuum values, labelled by an asterisk, for each of these quantities is obtained by Richardson extrapolation to the continuum based on the rate of convergence p (Vassberg and Jameson, 2010). These continuum values are expected to be close to the analytic values, which are available from Airy solution. There may be differences in the values because the NASTRAN finite element analysis is done for a 3-D model whereas the Airy solution is for a 2-D case.

First, the convergence of the primary analysis is shown in Figures 3.22–3.27. The continuum value of the displacement at the tip of the solid cantilever beam, $x = L$, $y = 1$, $z = 2$, matches almost exactly with the analytical value. The rate of convergence of the displacement is close to two. Convergence of the normal stress σ_x is studied near the root of the solid cantilever beam, where it peaks, at the point $x = 2$, $y = 1$, $z = 4$. Again, the continuum value matches almost exactly with the analytical value and the rate of convergence is close to 4. Convergence of the shear stress τ_{zx} is studied near the central portion of the solid cantilever beam, where it peaks, at the point $x = L/2$, $y = 1$, $z = 2$. Again, the continuum value matches almost exactly with the analytical value and the rate of convergence is close to 2.5.

Next, the accuracy and convergence of the CSA derivatives of displacement and stresses with respect to the design variable b_1 (uniform decrease in height by displacing the bottom surface) are shown in Figures 3.28–3.30. In all the results, CSE BCs were calculated using SGR with second-order Taylor series and four layer patches. Notice that the absolute errors of the derivatives at the finest mesh are about 7-10 orders of magnitude less than the actual derivative values. This confirms that CSA yields highly accurate results. Based on the continuum values, the derivative of transverse z-displacement, at $x = L$, $y = 1$, $z = 2$, obtained using CSA is closer to the analytical value than the corresponding NASTRAN

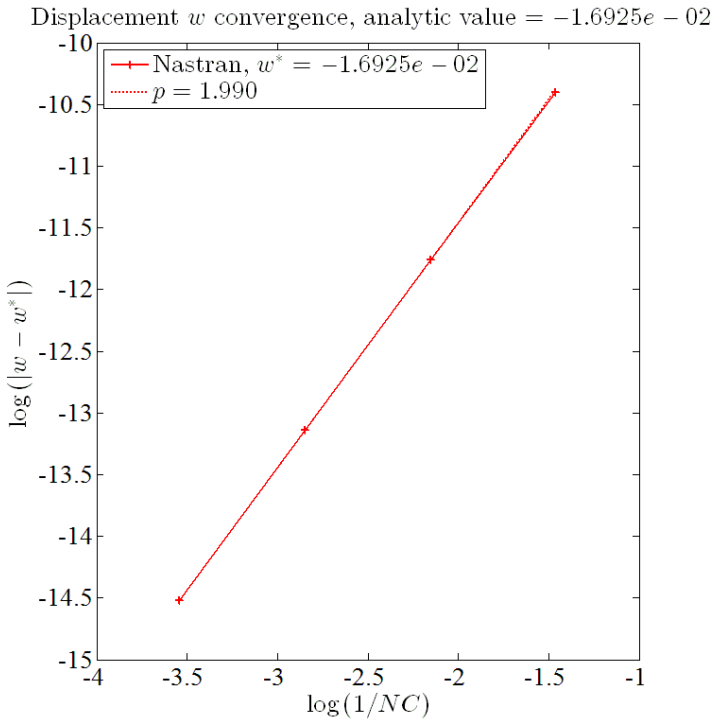

Figure 3.25: Convergence of the transverse z-displacement at the tip of the solid cantilever beam, $x = L$, $y = 1$, $z = 2$. NC stands for number of cells or number of elements.

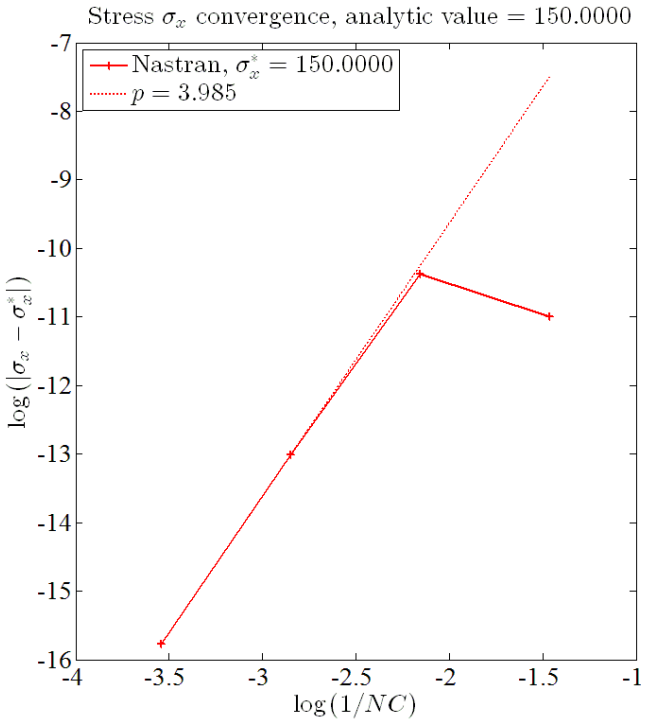

Figure 3.26: Convergence of the normal stress σ_x near the root of the solid cantilever beam, at the point $x = 2$, $y = 1$, $z = 4$. NC stands for number of cells or number of elements.

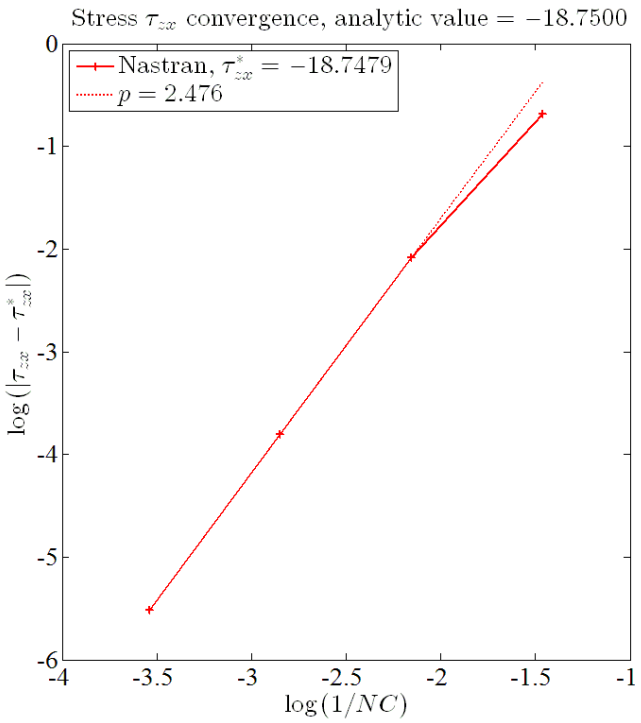

Figure 3.27: Convergence of the shear stress τ_{zx} near the central portion of the solid cantilever beam, at the point $x = L/2$, $y = 1$, $z = 2$. NC stands for number of cells or number of elements.

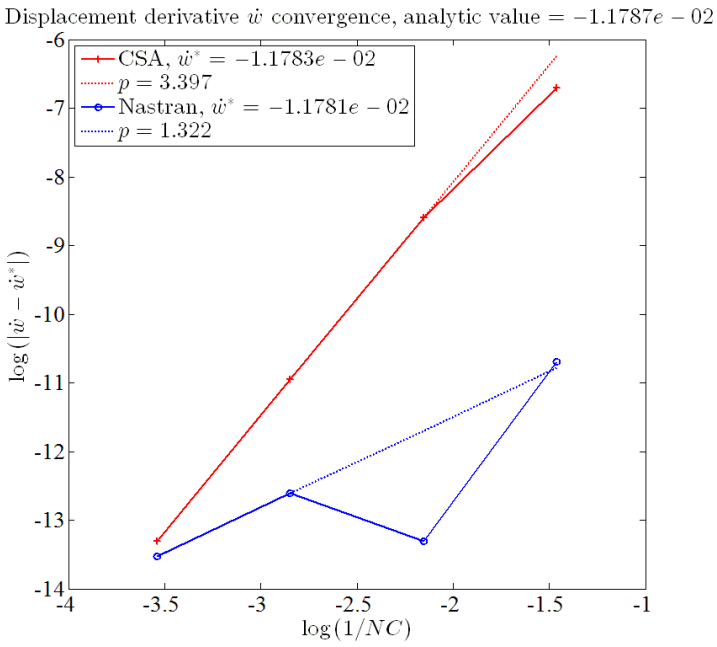

Figure 3.28: Convergence of the derivative transverse z-displacement with respect to the design variable b_1 at the tip of the solid cantilever beam, $x = L$, $y = 1$, $z = 2$. CSE BCs were calculated using SGR with second-order Taylor series and four layer patches.

derivative. The important observation regarding the displacement derivative result is that its rate of convergence is 3.4 significantly more than the rate of convergence of the primary analysis, which is 2. The NASTRAN displacement derivatives were observed not to be in the asymptotic range, although the absolute errors are of comparable to the converged CSA values. The approximate rate of convergence of the NASTRAN displacement derivatives is 1.3. The continuum values of normal and shear stress CSA derivatives also closely match the analytical values. Similar to the displacement derivative, the rates of convergence of the normal shear stress CSA derivatives are 4.5 and 3.4, respectively, i.e. the rates of convergence of CSA stress derivatives are about one more than the rates of convergence of the stresses. We can say that CSA yields super-convergent results in this case.

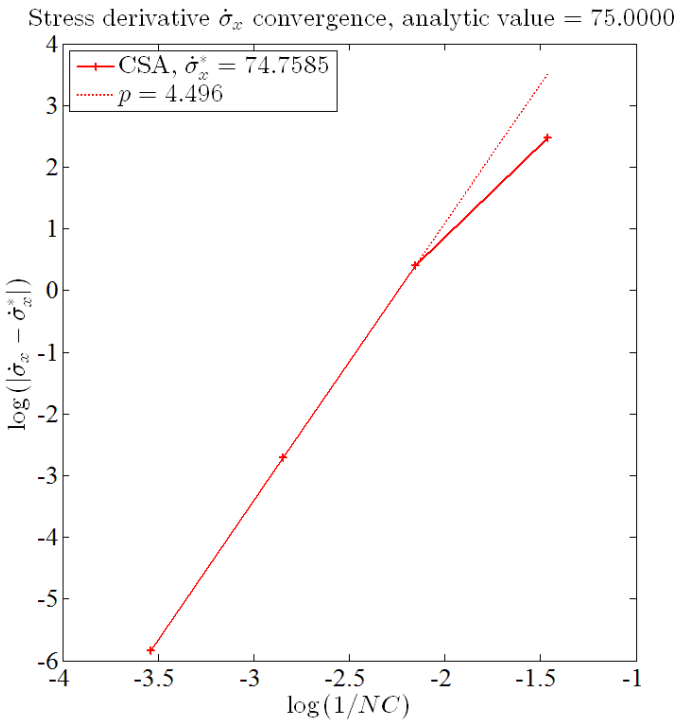

Figure 3.29: Convergence of the derivative of normal stress $\dot{\sigma}_x$ with respect to the design variable b_1 near the root of the solid cantilever beam, at the point $x = 2$, $y = 1$, $z = 4$. CSE BCs were calculated using SGR with second-order Taylor series and four layer patches.

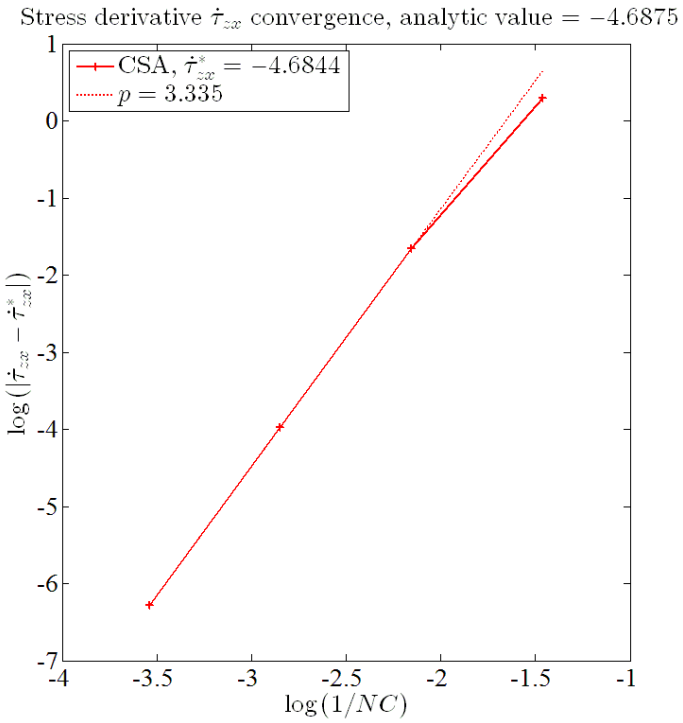

Figure 3.30: Convergence of the derivative of shear stress $\dot{\tau}_{zx}$ with respect to the design variable b_1 near the central portion of the solid cantilever beam, at the point $x = L/2$, $y = 1$, $z = 2$. CSE BCs were calculated using SGR with second-order Taylor series and four layer patches.

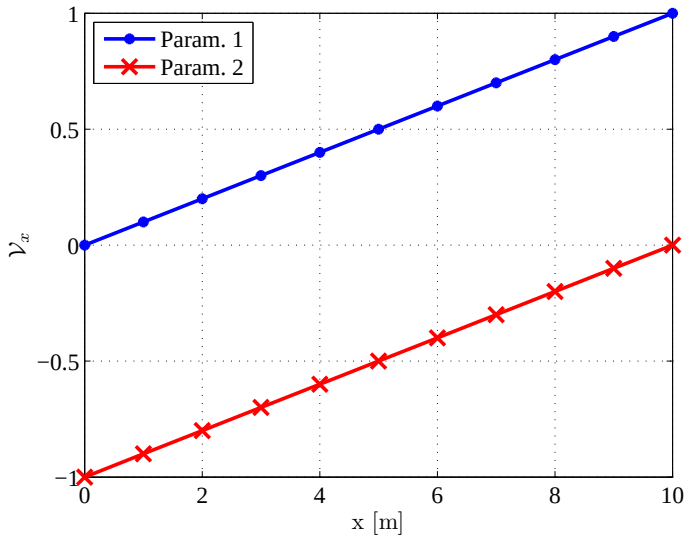

Figure 3.31: Design velocities for the two parameterizations of the cantilever beam.

3.6 Cantilever Beam with Solid Elements (Hybrid Adjoint Formulation)

3.6.1 Problem description

In this case, the same solid cantilever beam described in Section 3.5, but the design variables are chosen differently. Similar to the axial bar problem, the domain can be parameterized according to either two ways described in Section 3.4.1.1. So, the same design variable viz. length L of the beam can be considered as two separate design variables, each with its own corresponding design velocity. As a result, the design derivatives $\dot{w}$ obtained for each of these design variables should match with each other. This creates a good test case for checking the results obtained using the hybrid adjoint method. The actual three dimensional design velocity field $\boldsymbol{\mathcal{V}} = \left\{ \begin{array}{ccc} \mathcal{V}_x & \mathcal{V}_y & \mathcal{V}_z \end{array} \right\}^T$, for this simple choice of design variables and the rectangular cuboid geometry of the cantilever beam, is reduced to a one dimensional field with $\boldsymbol{\mathcal{V}} = \left\{ \begin{array}{ccc} \mathcal{V}_x & 0 & 0 \end{array} \right\}^T$. The design velocities for the two parameterizations are plotted in Figure 3.31.

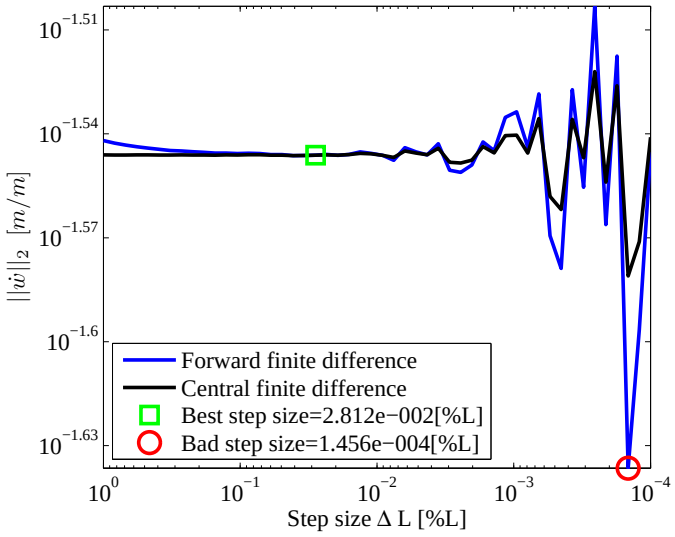

Figure 3.32: Step size study to determine the best step size: 2-Norm of the total derivative $\dot{w}$ on the center line of the beam, $y = 1$, $z = 2$, is plotted.

3.6.2 Result using finite difference method

An approximate solution was obtained using the finite difference method. It is difficult to identify a priori the correct finite difference step size that would give the least round-off error and the least truncation error. So, a step size study was conducted where 50 step sizes were chosen between $0.0001\,\%\,L$ and $1\,\%\,L$ as shown in Figure 3.32. The best finite difference result was obtained at a step size of $2.812 \times 10^{-2}\,\%\,L$ and is shown in Figure 3.33. However in this case, to realize which step size is the best, the NASTRAN finite element analysis (SOL 101) had to be conducted 51 times in all for the forward finite difference method and 100 times in all for the central finite difference method.

The finite difference result is compared to the the result obtained from Timoshenko beam theory for a 1-D case in Figure 3.33. The next sub-sections will illustrate how CSA was carried out using direct and hybrid adjoint formulations.

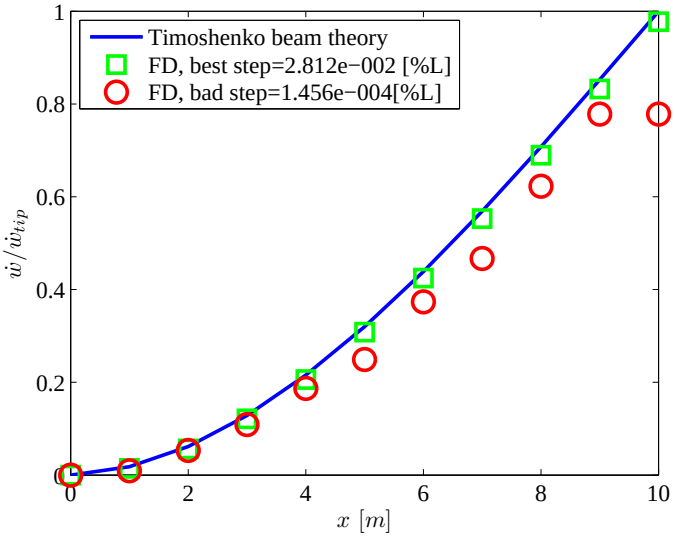

Figure 3.33: Total derivative of transverse displacement on the center line of the beam, $y = 1$, $z = 2$, obtained using finite difference method.

3.6.3 Results obtained from direct formulation of CSA

Since the detailed theoretical foundation of the direct formulation of CSA is given in Section 3.3.1, only the procedure used to get the direct CSA results nonintrusively using NASTRAN is given below.

1. Perform primary analysis (SOL 101) in NASTRAN; Output displacements (at nodes) and stresses (at Gauss points).

2. Parse the NASTRAN outputs and supply to 3D_SGR code.

3. 3D_SGR code gives the spatial gradients of the displacements and stresses on the boundary Γ.

4. Construct CSA boundary conditions, i.e. the prescribed displacements and/ or prescribed stresses.

5. Apply the CSA boundary conditions on the solid element model in NASTRAN.

6. Obtain local design derivatives by performing analysis (SOL 101) with the new boundary conditions and loads to solve Equation (3.14).

7. Add convective term to the local derivatives to obtain the total derivatives according to Equation (2.3).

Since there are two design variables, the above process for direct CSA involves three linear equation solutions in all: one for the primary analysis and one CSA in step-6 for each of the design variables. Hence, if the derivatives would have been required for n_b design variables, then the number of linear equation solutions required would be $(n_b + 1)$. However, since it is a direct formulation, the design derivatives can be obtained at all the points in the domain.

The above process is nonintrusive. In fact the same reduced stiffness matrix $[K_{22}]$ can be used to solve for local sensitivities for each of the design variable. So, step-6 in the above procedure could also be carried out without a new NASTRAN run, but with a restart for another CSA load case.

The nondimensional local and total design derivatives for the solid cantilever beam obtained using direct CSA are shown in Figures 3.34 and 3.35. The derivatives of the displacement on the center line of the beam are plotted.

3.6.4 Results obtained from hybrid adjoint formulation of CSA

The procedure outlined in Section 3.3.2 is followed for obtaining the design derivatives. Five performance measures are specified which are the transverse displacement w at five locations $x = 2$, 4, 6, 8, and 10 on the center line, $y = 1$, $z = 2$ of the cantilever beam. Once the performance measures are defined, we can get the corresponding virtual load vectors $\{\boldsymbol{z}\}$ for each performance measure. The following steps were involved in obtaining design derivatives with respect to the two design variables using the hybrid adjoint formulation of CSA for the cantilever beam problem:

1. Perform primary analysis (SOL 101) in NASTRAN; Output displacements (at nodes),

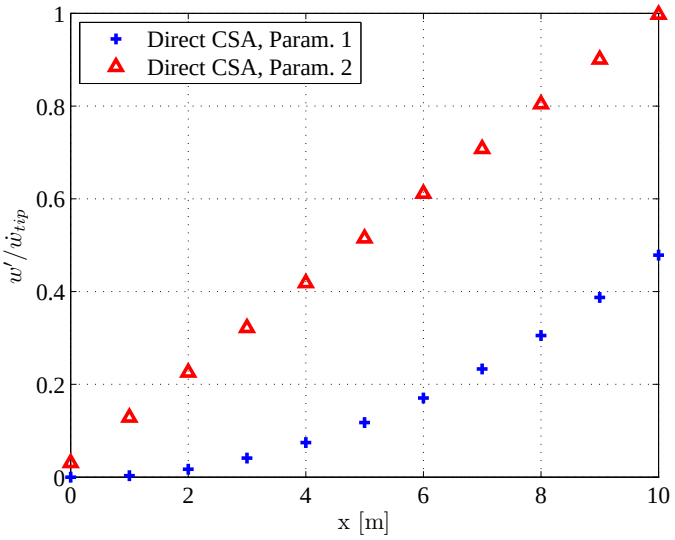

Figure 3.34: Nondimensional local derivative of the transverse displacement of cantilever beam on the center line, with respect to its length, calculated using direct CSA formulation.

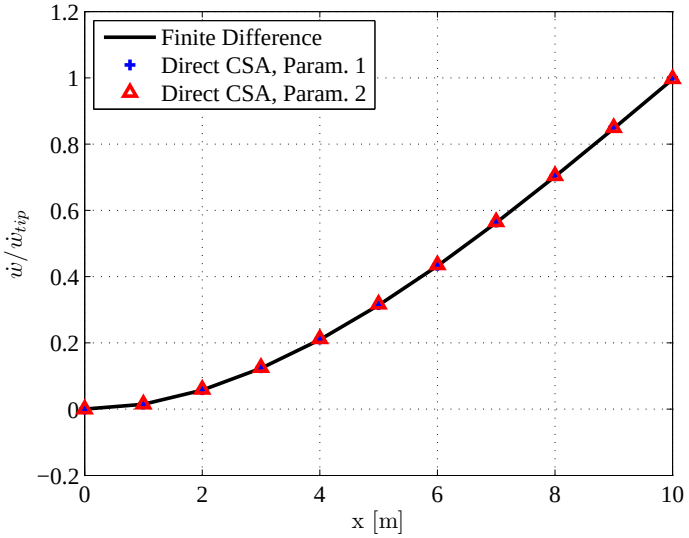

Figure 3.35: Nondimensional total derivative of the transverse displacement of cantilever beam on the center line, with respect to its length, calculated using direct CSA formulation.

stresses (at Gauss points) and the reduced stiffness matrix $[K_{22}]$

2. Parse the NASTRAN outputs and supply to 3D_SGR code

3. 3D_SGR code gives the spatial gradients of the displacements and stresses on the boundary Γ

4. Construct CSA boundary conditions, i.e. obtain the correct values of $\left\{\boldsymbol{u_1'}\right\}$ and/ or $\left\{\boldsymbol{F_2'}\right\}$ in Equation (3.12)

5. Solve for adjoint variable $\{\boldsymbol{\lambda}\}$ for each performance measure according to Equation (3.21)

6. Obtain local design derivatives by performing the inner product calculations in Equation (3.22)

7. Add convective term to the local derivatives to obtain the total derivatives as given in Equation (3.23). Here, the convective term is only required to be calculated for the the degrees of freedom included in performance measures.

In this procedure, only two linear equation solutions are involved: one for the primary analysis and another in step-5 for solution to the adjoint variable. Hence, even for n_b design variables, it would involve only two linear solutions. However, all the derivatives are for the $n_\psi = 5$ performance measures and not for the displacements at all locations in the domain. This is clearly beneficial when performance derivatives are required for large number of design variables. Again, similar to direct CSA, even adjoint CSA formulation has been implemented nonintrusively.

The nondimensional local and total design derivatives for the solid cantilever beam obtained using hybrid adjoint CSA are shown in Figures 3.36 and 3.37. It can be seen that these derivatives match not only with the direct CSA results but also with the finite difference results.

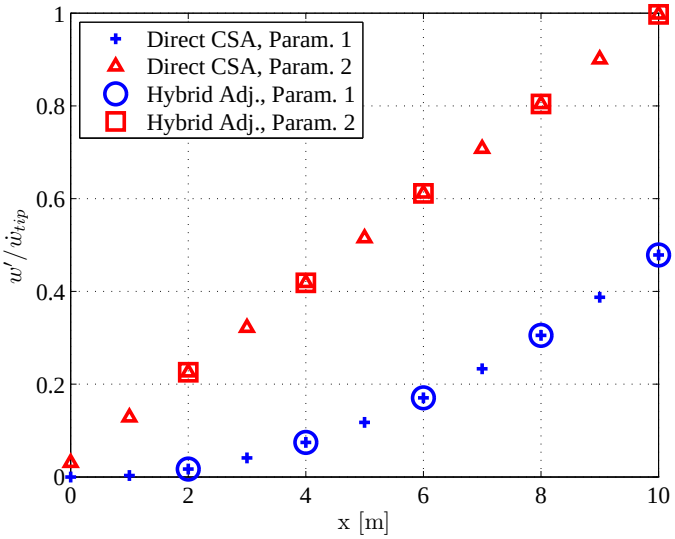

Figure 3.36: Nondimensional local derivative of the transverse displacement of cantilever beam on the center line, with respect to its length, calculated using hybrid adjoint CSA formulation.

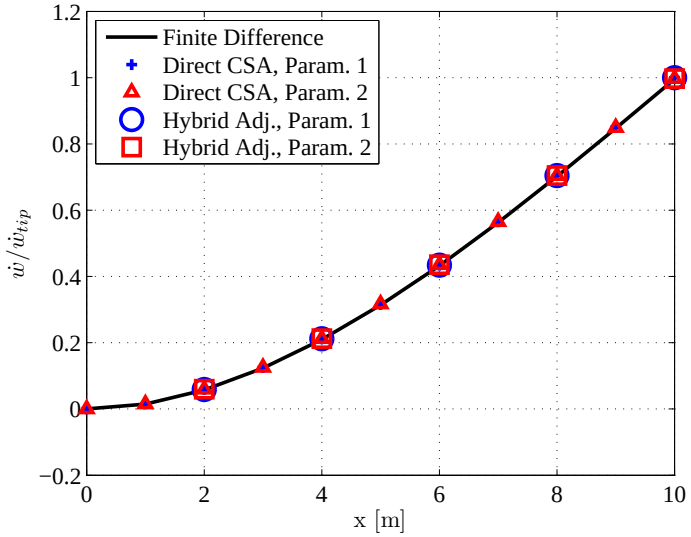

Figure 3.37: Nondimensional total derivative of the transverse displacement of cantilever beam on the center line, with respect to its length, calculated using hybrid adjoint CSA formulation.

3.7 Conclusion

A new continuum-discrete hybrid adjoint formulation was introduced that has the following advantages: (a) it involves solving a linear discrete system only once for calculating derivatives of a single performance criterion with respect to many design variables, (b) it enjoys the benefits of local CSA with SGR, namely, the sensitivities are accurate and the mesh sensitivity is avoided, and (c) unlike the continuous adjoint method (Lozano and Ponsin, 2012; Duivesteijn et al., 2005), there are no complications associated with adjoint boundary conditions. This method was applied to a 1-D axial bar example and a 3-D cantilever beam example (Nastran, 2010) modelled with solid elements. Local and total derivatives of a few responses were obtained for multiple design variables using nonintrusive hybrid adjoint CSA. The black-box used for the analyses was NASTRAN. The derivation of the hybrid adjoint is independent of the number of spatial dimensions and so, it can be easily extended form 1-D to 2-D or 3-D examples. Application of this method to other disciplines, such as fluid analysis or coupled fluid-structure analysis is straightforward. The hybrid adjoint CSA method computes the same values for shape derivatives as direct CSA. Therefore accuracy and convergence properties are the same as for the direct local CSA (Cross and Canfield, 2015). Finally, we showed that for many design variables the hybrid adjoint formulation permits computing shape derivatives with less effort as compared to the direct CSA.

Chapter 4

Nonintrusive Continuum Sensitivity Analysis for Aeroelastic Shape Derivatives

ABSTRACT

Continuum Sensitivity Analysis (CSA) provides an analytic method to obtain design derivatives of structural and fluid responses. The primary advantages of the presented local CSA formulation are that analytic derivatives are computed and mesh sensitivity is avoided. Spatial Gradient Reconstruction (SGR) has been applied for nonintrusive implementation of CSA for structural and fluid systems. In the current work, we derive CSA for a coupled fluid-structure system.

4.1 Introduction

Sensitivity analysis plays an important role in gradient-based optimization techniques. In fact, convergence of a shape optimization problem depends on the accuracy of gradients of the performance and constraint functions with respect to design variables. Sensitivity analysis methods can be broadly categorized as numeric methods (finite difference, complex step),

analytic methods (discrete analytic, continuum) or automatic differentiation method. In general, analytic methods are favored over numeric methods because of their higher accuracy and lower computational cost. Furthermore, analytic methods have the following advantages: (a) there is no need of convergence study for choosing the correct step size (required for the finite difference method), and (b) there is no requirement of the analysis code to handle complex number operations (required for the complex step method). Among the analytic methods, the discrete analytic method involves discretizing the governing equations followed by differentiation. Hence, for shape design variables it suffers from the disadvantage of calculating mesh sensitivity. Additionally, intimate knowledge of the analysis procedure is required for implementation of the discrete analytic method because it is based on "discretize then differentiate" approach. Automatic differentiation requires the actual source code for its implementation. Continuum Sensitivity Analysis (CSA) may offer a better alternative.

CSA has been used successfully to compute shape derivatives of structural response (Dems and Haftka, 1988; Haug et al., 1986; Arora, 1993; Choi and Kim, 2005) and fluid response (Borggaard and Burns, 1994, 1997; Stanley and Stewart, 2002; Turgeon et al., 2005; Duvigneau and Pelletier, 2006). However its application for aeroelastic response shape derivatives is relatively new (Etienne and Pelletier, 2005; Liu and Canfield, 2013a, 2016; Cross and Canfield, 2014). The motivation for using CSA for shape sensitivity of aeroelastic response is twofold: (a) gradients are analytic (accurate and more efficient than finite difference) and (b) mesh sensitivity calculation is avoided (a drawback of the discrete analytic shape sensitivity approach).

In the current work, we demonstrate a nonintrusive implementation of CSA for analyzing coupled fluid-structure systems. Recently, Cross and Canfield (Cross and Canfield, 2014) demonstrated a nonintrusive implementation of CSA for coupled fluid-structure system with typical section aerodynamics. Such aerodynamic theory may not work for designing systems with compressible or viscous flow. Here, we extend the implementation of CSA for fluids (Borggaard and Burns, 1994, 1997; Duvigneau and Pelletier, 2006) to CSA for an aeroelastic

system with Euler flow aerodynamics. Similar to the coupling between the structural and fluid degrees of freedom, the structural and fluid variable derivatives are also coupled. We derive the sensitivity equations for the structural and fluid systems, identify the sources of the coupling between the structural and fluid derivatives, and implement CSA nonintrusively for structural finite element and fluid finite volume formulations to obtain the aeroelastic response derivatives.

The example chosen for this purpose is of a flexible airfoil subjected to uniform subsonic flow. The interface that separated the fluid and structural domains is thus chosen to be flexible. This leads to coupling terms in the sensitivity analysis which are highlighted. In particular, the structural sensitivity boundary loads are dependent on the pressure and its derivative on the flexible airfoil. Along with this, the fluid sensitivity transpiration boundary condition is applied on the flexible fluid-structure interface. This results in the fluid sensitivity boundary conditions being dependent on the structural deformation of the flexible interface and its derivatives.

Cross and Canfield (Cross and Canfield, 2012, 2014) demonstrated that, using Spatial Gradient Reconstruction (SGR), analytic shape derivatives can be computed without any information about the finite element formulation, element shape functions, or how the element shapes change with changes to the mesh when the shape of the structure changes. In other words, for the finite element method, the presented CSA method is element-agnostic and does not require the knowledge of how the mesh changes with shape design variables. In the current work, we will use SGR for nonintrusive implementation of CSA for the flexible airfoil example.

4.2 Local Continuum Shape Sensitivity Formulation for Aeroelastic System

4.2.1 Governing Equations

Consider the following continuous nonlinear differential equation defined over a domain Ω with a boundary Γ subject to distributed body forces $\boldsymbol{f}$. We seek a solution for the state variables $\boldsymbol{u}(\boldsymbol{x};t;b)$ of the equations

$$\boldsymbol{R}(\boldsymbol{u},t;b) = \boldsymbol{\mathcal{A}}(\boldsymbol{u}, L(\boldsymbol{u})) - \boldsymbol{f}(\boldsymbol{x},t;b) = 0 \quad \text{on } \Omega, \tag{4.1}$$

with the boundary conditions (BCs)

$$\boldsymbol{\mathcal{B}}(\boldsymbol{u}, L(\boldsymbol{u})) = \boldsymbol{g}(\boldsymbol{x},t;b) \quad \text{on } \Gamma, \tag{4.2}$$

where the vector of dependent (state) variables $\boldsymbol{u}(\boldsymbol{x},t;b)$ are functions of the spatial and temporal independent coordinates, $\boldsymbol{x}$ and t, respectively and depend implicitly on design variable b. The domain and boundary in Cartesian space are shown in Figure 4.1. The linear differential operator L has terms such as $\left\{\frac{\partial}{\partial t}, \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial y^2}, \dots\right\}$ that appear in the governing equations or boundary conditions. $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ are algebraic or integral operators acting on $\boldsymbol{u}$ and $L(\boldsymbol{u})$ possibly in nonlinear fashion. Eqs. (4.1) and (4.2) are written in a general notation and apply to the fluid domain Ω_f or the structural domain Ω_s, where $\Omega = \Omega_f \cup \Omega_s$,

$$\boldsymbol{\mathcal{A}}(\boldsymbol{u}, L(\boldsymbol{u})) = \left\{ \begin{array}{c} \boldsymbol{\mathcal{A}}_f(\boldsymbol{u}_f, L_f(\boldsymbol{u}_f)) \\ \boldsymbol{\mathcal{A}}_s(\boldsymbol{u}_s, L_s(\boldsymbol{u}_s)) \end{array} \right\}, \tag{4.3}$$

and the state variables are partitioned for the fluid and structural responses: $\boldsymbol{u}(\boldsymbol{x};t;b) =$

$\left\{\boldsymbol{u}_f^T,\, \boldsymbol{u}_s^T\right\}^T$. Eq. (4.2) is a general form of representing structural or flow BCs

$$\boldsymbol{\mathcal{B}}\left(\boldsymbol{u}, L\left(\boldsymbol{u}\right)\right) - \boldsymbol{g}\left(\boldsymbol{x}; t; b\right) \equiv \left\{ \begin{array}{c} \boldsymbol{\mathcal{B}}_{ef}\left(\boldsymbol{u}_f\right) - \boldsymbol{g}_{ef}\left(\boldsymbol{x}; t; b\right) \\ \boldsymbol{\mathcal{B}}_{es}\left(\boldsymbol{u}_s\right) - \boldsymbol{g}_{es}\left(\boldsymbol{x}; t; b\right) \\ \boldsymbol{\mathcal{B}}_{nf}\left(\boldsymbol{u}_f, L\left(\boldsymbol{u}_f\right)\right) - \boldsymbol{g}_{nf}\left(\boldsymbol{x}; t; b\right) \\ \boldsymbol{\mathcal{B}}_{ns}\left(\boldsymbol{u}_s, L\left(\boldsymbol{u}_s\right)\right) - \boldsymbol{g}_{ns}\left(\boldsymbol{x}; t; b\right) \\ \boldsymbol{\mathcal{B}}_{fs}\left(\boldsymbol{u}, L\left(\boldsymbol{u}\right)\right) - \boldsymbol{g}_{fs}\left(\boldsymbol{x}; t; b\right) \end{array} \right\}, \tag{4.4}$$

which may be either Dirichlet (essential or geometric) such as a prescribed value $\boldsymbol{\mathcal{B}}_e\left(\boldsymbol{u}\right) \equiv \boldsymbol{u}|_{\Gamma_e} = \boldsymbol{g}_e \equiv \bar{\boldsymbol{u}}$ on the boundary Γ_e, or they may involve a differential operator for Neumann (nonessential or natural) BCs such that $\boldsymbol{\mathcal{B}}_n\left(L\left(\boldsymbol{u}\right)\right) = \boldsymbol{g}_n$ on the boundary Γ_n. Whereas the partial differential equation (PDE) operators for the fluid, $\boldsymbol{\mathcal{A}}_f$, and the structure, $\boldsymbol{\mathcal{A}}_s$, in Eq. (4.3) are decoupled with respect to the other domain's state variables, the fluid-structure system is coupled by the appearance of BCs of the form $\boldsymbol{\mathcal{B}}_{fs}\left(\boldsymbol{u}, L\left(\boldsymbol{u}\right)\right) - \boldsymbol{g}_{fs}\left(\boldsymbol{x}; t; b\right)$ at the fluid-structure interface boundary Γ_{fs} as given in Eq. (4.4). An example of such a boundary condition is the no-penetration BC. Since the current work involves steady-state flow and sensitivity analysis, the time term t in these equations is suppressed hereafter.

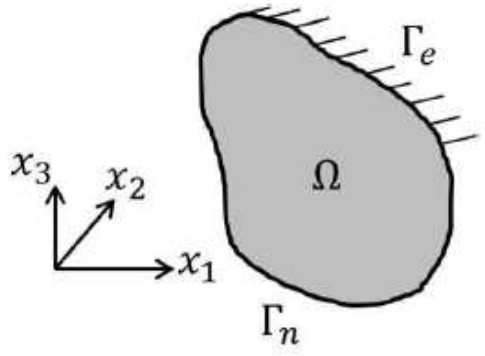

Figure 4.1: Domain, Ω, with boundary Γ.

4.2.2 Differentiation of Continuous Equations

The focus of the current work is to obtain design derivatives of flow variables at points in the domain with respect to a limited number of design variables. So, we use the direct formulation of CSA as explained next. However, an adjoint formulation of CSA is also available, which allows one to calculate efficiently derivatives of a limited number of performance measures with respect to many design variables using the same CSA boundary conditions presented here (Kulkarni et al., 2016). Consider the problem of obtaining the derivative of the steady-state response $\boldsymbol{u}(\boldsymbol{x}; b)$ with respect to design parameter b at all points in the domain. The response depends on the spatial variable $\boldsymbol{x}$ and may have an explicit or implicit dependence on the shape variable b, as indicated by the semicolon. The boundary velocity (local) formulation of CSA results in Continuum Sensitivity Equations (CSEs) that are posed in terms of the local derivatives of the response, $\boldsymbol{u'} = \partial \boldsymbol{u}/\partial b$. Hence, solution of the CSEs yields the local derivative. The total or material derivative $\dot{\boldsymbol{u}} = D\boldsymbol{u}/Db$ is then obtained by adding the convective term to the local derivative.

$$\frac{D\boldsymbol{u}}{Db} = \frac{\partial \boldsymbol{u}}{\partial b} + \sum_{i=1}^{3} \frac{\partial \boldsymbol{u}}{\partial x_i} \frac{\partial x_i}{\partial b} \quad \Longleftrightarrow \quad \dot{\boldsymbol{u}} = \boldsymbol{u'} + \boldsymbol{\nabla}_{\boldsymbol{x}} (\boldsymbol{u}) \cdot \boldsymbol{\mathcal{V}} \tag{4.5}$$

The convective term consists of the spatial gradients of the response $\boldsymbol{\nabla}_{\boldsymbol{x}} \boldsymbol{u} = \partial \boldsymbol{u}/\partial \boldsymbol{x}$, and the geometric sensitivity or design velocity $\boldsymbol{\mathcal{V}}(\boldsymbol{x}) = \partial \boldsymbol{x}/\partial b$, which depends on the geometric parametrization of the domain. For value design parameters independent of shape, the convective term goes to zero, because the design velocity is zero, and so the material derivative is same as the local derivative. However, for shape design variables, the design velocity is not zero and hence there is a need to calculate the convective term for CSE boundary conditions and transformation to material derivative wherever necessary.

CSA is based on the philosophy of "differentiate and then discretize" and involves differentiating Eqs. (4.1) and (4.2) with respect to b, followed by discretization and solution of the resulting discretized system. Based on the type of differentiation, CSA is categorized as

either local form CSA or total form CSA (Liu and Canfield, 2016) in accordance with Eq. (4.5). The local form CSA involves partial differentiation of Eq. (4.1), while the total form CSA involves total differentiation. Due to the advantages (Cross and Canfield, 2014) of the local form CSA over the total form CSA, the current work focuses on the local form CSA. The CSEs are obtained by partial differentiation of Eq. (4.1) as

$$\frac{\partial \boldsymbol{R}}{\partial b} = \frac{\partial \boldsymbol{\mathcal{A}}\left(\boldsymbol{u}, L\left(\boldsymbol{u}\right)\right)}{\partial \boldsymbol{u}}\boldsymbol{u'} + \frac{\partial \boldsymbol{\mathcal{A}}\left(\boldsymbol{u}, L\left(\boldsymbol{u}\right)\right)}{\partial L}L\left(\boldsymbol{u'}\right) - \frac{\partial \boldsymbol{f}\left(\boldsymbol{x}, t; b\right)}{\partial b} = 0. \tag{4.6}$$

Since the material boundary changes due to a change in the shape design parameter, the boundary conditions for the CSEs are obtained by total or material differentiation of the original boundary conditions (4.2) and moving the convective terms to the right side

$$\frac{\partial \boldsymbol{\mathcal{B}}}{\partial u}\boldsymbol{u'} + \frac{\partial \boldsymbol{\mathcal{B}}}{\partial L}L\left(\boldsymbol{u'}\right) = \dot{\boldsymbol{g}}\left(\boldsymbol{x}, t; b\right) - \boldsymbol{\mathcal{V}}\left(\boldsymbol{x}\right) \cdot \left(\frac{\partial \boldsymbol{\mathcal{B}}}{\partial u}\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u} + \boldsymbol{\nabla}_{\boldsymbol{x}}\left(\boldsymbol{\mathcal{B}}\left(L\left(\boldsymbol{u}\right)\right)\right)\right), \tag{4.7}$$

where $\dot{\boldsymbol{g}}\left(\boldsymbol{x}, t; b\right)$ is the material derivative of the prescribed boundary condition, typically zero for Dirichlet boundary conditions. Nevertheless, even when the the boundary condition (4.2) is homogeneous, $\boldsymbol{u}|_{\Gamma_e} = 0$, the CSE boundary condition (4.7) is in general non-homogeneous, $\boldsymbol{u'}|_{\Gamma_e} = \dot{\boldsymbol{g}}_e - \boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u} \cdot \boldsymbol{\mathcal{V}}\left(\boldsymbol{x}\right)$, due to the convective term $-\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{u} \cdot \boldsymbol{\mathcal{V}}\left(\boldsymbol{x}\right)$, even for $\dot{\boldsymbol{g}}_e = 0$. The commutation of derivatives on the left side of Eq. (4.7) is possible when the derivatives are local. The CSEs (4.6) with the boundary conditions (4.7) form a linear system of equations in terms of sensitivity variable $\boldsymbol{u'}$, which can be solved by the same or different numerical method used for solving the analysis problem.

Eqs. (4.6) and (4.7) may be restated as

$$\frac{\partial \boldsymbol{R}}{\partial b} = \boldsymbol{\mathcal{A}}_b\left(\boldsymbol{u}, L\left(\boldsymbol{u'}\right)\right) - \boldsymbol{f'}\left(\boldsymbol{x}, t; b\right) = 0 \quad \text{on } \Omega, \tag{4.8}$$

with the corresponding sensitivity BCs

$$\boldsymbol{\mathcal{B}}_b\left(\boldsymbol{u}, L\left(\boldsymbol{u'}\right)\right) = \boldsymbol{g}_b\left(\boldsymbol{x}, t; b\right) \quad \text{on } \Gamma, \tag{4.9}$$

where $\boldsymbol{g}_b$ is the right side of Eq. (4.7). The similarity of Eqs. (4.8) and (4.9) to Eqs. (4.1) and (4.2) motivates the same solution method for each set of equations with the same mesh for the discretized form. For linear governing Eqs. (4.1), $\boldsymbol{\mathcal{A}}_b = \boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}_b = \boldsymbol{\mathcal{B}}$. For nonlinear governing equations, the CSEs are still linear in $\boldsymbol{u'}$ but with nonlinear dependence on the solution $\boldsymbol{u}$, which can be obtained from the analysis solution of Eq. (4.1) for use in Eqs. (4.8) and (4.9).

In the direct formulation of CSA, Eqs. (4.8) and (4.9) are solved to obtain the local derivatives $\boldsymbol{u'}$ in the domain Ω. This may be followed by adding the convective term as shown in (4.5), at the required locations in the domain, to obtain the total derivative of the response variable $\dot{\boldsymbol{u}}$ at those locations. In shape optimization applications, one may be interested in the derivative of a performance metric such as

$$\psi = \int_{\bar{\Gamma}} \sigma \boldsymbol{u} \, d\bar{\Gamma} \tag{4.10}$$

which is based on weighted surface integral of the response $\boldsymbol{u}$. Here σ are the weights and $\bar{\Gamma} \in \Gamma$. Assuming that the weights do not depend on the shape design variable, the total derivative of the performance measure is obtained using the values of $\dot{\boldsymbol{u}}$ as

$$\dot{\psi} = \frac{D\psi}{Db} = \int_{\bar{\Gamma}} \sigma \dot{\boldsymbol{u}} \, d\bar{\Gamma} + \int_{\bar{\Gamma}} \sigma u \, \dot{d\bar{\Gamma}}, \tag{4.11}$$

where $\dot{d\bar{\Gamma}}$ denotes the infinitesimal change in the boundary $\bar{\Gamma}$ due to the shape design variable.

4.2.3 Discretization of the Differentiated Equations

Until this point, the continuous governing equations were differentiated to obtain the CSEs. Thus, there is no approximation involved in deriving the CSE system (4.8–4.9). Next, consider a discretization at mesh level h of the flow equations and a Newton-Raphson implicit

scheme for primary analysis which results in a coupled linear system of equations

$$\left[T\left(\{\boldsymbol{u}\}_h^n\right)\right]\{\Delta\boldsymbol{u}\}_h^n = \{\boldsymbol{R}\}_h^n, \tag{4.12}$$

which is solved at each iteration n for the updates to the flow variables $\{\Delta\boldsymbol{u}\}_h^n$. This update is used to get the values of flow variables at the next iteration $\{\boldsymbol{u}\}_h^{n+1} = \{\boldsymbol{u}\}_h^n + \{\Delta\boldsymbol{u}\}_h^n$. Here $\left[T\left(\{\boldsymbol{u}\}_h^n\right)\right]$ is the tangent matrix, and $\{\boldsymbol{R}\}_h^n$is the residual of the flow equations at time step n and discretization h. Similarly, the CSEs can be discretized to obtain a coupled linear system of equations

$$\left[T_{\text{CSE}}\left(\{\boldsymbol{u}\}_h\right)\right]\{\Delta\boldsymbol{u'}\}_h \quad = \quad \{\boldsymbol{R}_{\text{CSE}}\}_h. \tag{4.13}$$

The tangent matrix $\left[T_{\text{CSE}}\left(\{\boldsymbol{u}\}_h\right)\right]$ in the discretized CSEs (4.13) is independent of the sensitivity variables $\boldsymbol{u'}$ and only depends on the flow variables $\boldsymbol{u}$. Borggaard and Burns (Borggaard and Burns, 1994, 1997), Wickert et al. (Wickert et al., 2010) and Liu and Canfield (Liu and Canfield, 2013b) have shown that if the same discretization used for the analysis is used to discretize the CSEs, then

$$\left[T_{\text{CSE}}\left(\{\boldsymbol{u}\}_h\right)\right] = \left[T\left(\{\boldsymbol{u}\}_h^N\right)\right],$$

where N is the last iteration step of the flow solver until steady-state convergence is achieved. Also, since the CSEs are linear in the sensitivity variables, the local sensitivities are obtained by just a single (one-shot) solution of the linear system (4.13) for $\{\boldsymbol{u'}\}_h$, leading to $\{\boldsymbol{u'}\}_h = \{\Delta\boldsymbol{u'}\}_h$ with zero initial guess $\left(\{\boldsymbol{u'}\}_h^0 = 0\right)$ without loss of generality. Once the local derivatives are computed by solving (4.13), the total derivatives can be obtained by adding the convective term according to the discretized version of (4.5) and the derivative of the performance measure can be obtained by the discretized version of (4.11).

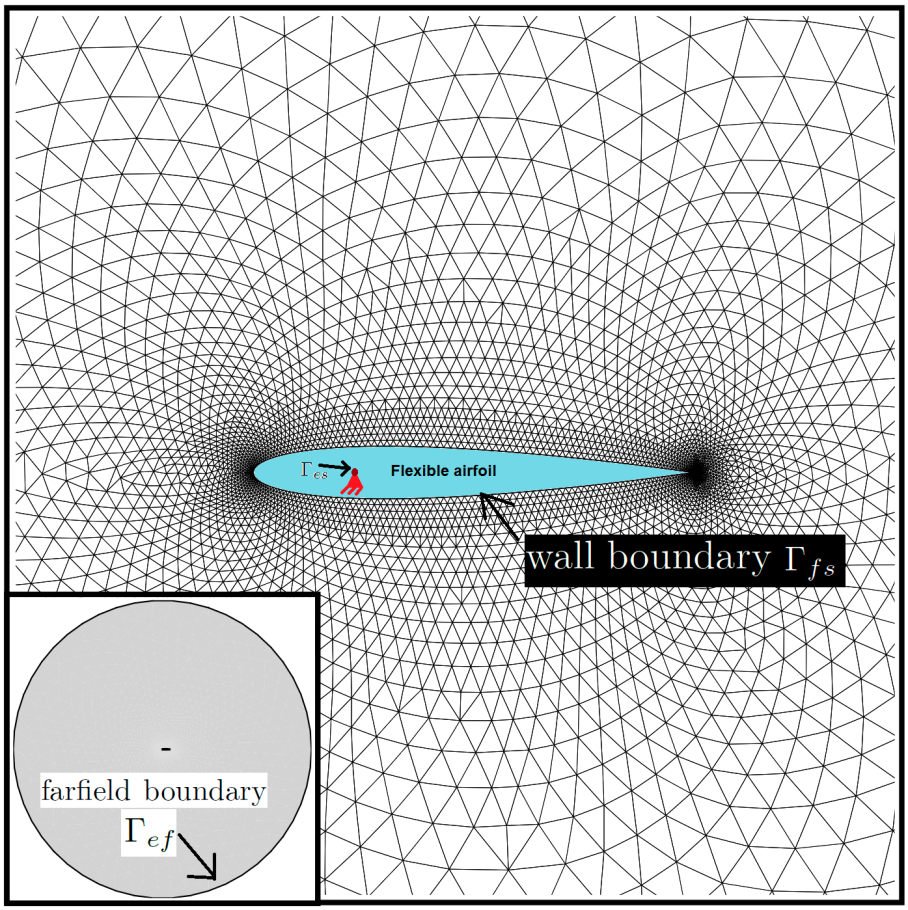

Figure 4.2: Flexible airfoil in uniform flow.

4.2.4 Two-dimensional Coupled Fluid-Structure Analysis

Consider the example of a flexible airfoil in uniform flow as shown in Figure 4.2. The airfoil is assumed to be made of elastic material which allows in-plane deformation of the airfoil boundary. The airfoil is fixed at the quarter chord by constraining the in-plane and drilling degrees of freedom. The quarter chord point is the essential boundary Γ_{es} for structural analysis. In the following sections, the undeformed airfoil boundary with respect to which the structural deformation is obtained is denoted by Γ_{ns}. The deformed airfoil boundary which is used for evaluating the flow is denoted by Γ_{fs}. The farfield boundary is denoted by Γ_{ef}.

4.2.4.1 Structural Response

The partial differential equations governing a 2D structural displacement response $\boldsymbol{u_s} = \{u_x,\, u_y\}^T$based on linear elasticity are

$$[\boldsymbol{\partial}]\,\mathcal{D}\,[\boldsymbol{\partial}]^T\,\boldsymbol{u_s} = \boldsymbol{F} \quad \text{on } \Omega, \tag{4.14}$$

where $F = \{F_x,\, F_y\}^T$ are the applied body forces at a point in the domain Ω, $\mathcal{D}$ is the constitutive matrix, and $[\boldsymbol{\partial}]$ is the operator matrix given by

$$[\boldsymbol{\partial}] = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}. \tag{4.15}$$

The stresses $\boldsymbol{\sigma} = \{\sigma_x,\, \sigma_y,\, \tau_{xy}\}^T$ are related to the strains $\boldsymbol{\epsilon} = \{\epsilon_x,\, \epsilon_y,\, \gamma_{xy}\}^T$ through the plane stress constitutive relationship $\boldsymbol{\sigma} = \mathcal{D}\boldsymbol{\epsilon}$, and the strains are are related to the displacement through the strain-displacement relationship $\boldsymbol{\epsilon} = [\boldsymbol{\partial}]^T\,\boldsymbol{u_s}$. Comparing Eq. (4.14) to the general Eq. (4.1), we can derive the structural operator

$$\boldsymbol{\mathcal{A}}_s\left(\boldsymbol{u}_s,\, L_s\left(\boldsymbol{u}_s\right)\right) \equiv L_s\left(\boldsymbol{u}_s\right) = [\boldsymbol{\partial}]\,\mathcal{D}\,[\boldsymbol{\partial}]^T\,\boldsymbol{u}_s. \tag{4.16}$$

The boundary conditions (4.2) may be either essential (geometric) boundary conditions, when the displacements $\bar{\boldsymbol{u}}_s$ are prescribed

$$\boldsymbol{u}_s\big|_{\Gamma_{es}} = \bar{\boldsymbol{u}}_s, \tag{4.17}$$

or nonessential (natural) boundary conditions, when surface tractions $\bar{\boldsymbol{\Phi}} = \left\{\bar{\Phi}_x,\, \bar{\Phi}_y\right\}^T$ are prescribed

$$\boldsymbol{\Phi}\big|_{\Gamma_{ns}} = \bar{\boldsymbol{\Phi}}. \tag{4.18}$$

The tractions at any point on the boundary surface are related to the stresses at that point by the relation

$$\boldsymbol{\Phi} = \left\{ \begin{array}{c} \Phi_x \\ \Phi_y \end{array} \right\} = \left[\begin{array}{cc} \sigma_x & \tau_{xy} \\ \tau_{xy} & \sigma_y \end{array} \right] \left\{ \begin{array}{c} n_{xs} \\ n_{ys} \end{array} \right\} = [\boldsymbol{\sigma}]\, \boldsymbol{n}_s, \tag{4.19}$$

where $\boldsymbol{n}_s = \{n_{xs},\, n_{ys}\}^T$ are the direction cosines of a vector normal to the undeformed surface Γ_{ns}. Thus, the boundary condition (4.18) can be written in terms of the stress components as

$$[\boldsymbol{\sigma}]\, \boldsymbol{n}_s|_{\Gamma_{ns}} = \bar{\boldsymbol{\Phi}}. \tag{4.20}$$

For the example of the flexible airfoil, Γ_{es} is the quarter chord point at which the airfoil is constrained and $\bar{\boldsymbol{u}}_{\boldsymbol{s}} = 0$. The drilling degree of freedom (Nastran, 2004) is also constrained to avoid rotation about the axis perpendicular to the plane of the airfoil. This boundary condition is represented as

$$\frac{1}{2}\left(\frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y}\right)\bigg|_{\Gamma_{es}} = 0, \tag{4.21}$$

The airfoil is immersed in uniform flow. This results in a pressure distribution p on the airfoil surface. Thus the traction loads $\bar{\boldsymbol{\Phi}}$ on the airfoil structure are functions of the fluid flow,

$$\bar{\boldsymbol{\Phi}} \equiv -p\boldsymbol{n}_{fs}|_{\Gamma_{fs}}\,. \tag{4.22}$$

More details of the fluid-structure coupling are given in Section 4.2.4.3.

4.2.4.2 Fluid Response

The Euler equations in conservation form for flow over an airfoil, are

$$\frac{\partial \mathscr{F}}{\partial x} + \frac{\partial \mathscr{G}}{\partial y} = 0 \tag{4.23}$$

or

$$+ \frac{\partial \mathscr{F}}{\partial \boldsymbol{u}_f}\frac{\partial \boldsymbol{u}_f}{\partial x} + \frac{\partial \mathscr{G}}{\partial \boldsymbol{u}_f}\frac{\partial \boldsymbol{u}_f}{\partial y} = 0 \tag{4.24}$$

subject to farfield boundary condition

$$\boldsymbol{u}_f|_{\Gamma_{ef}} = \boldsymbol{u}_{f\infty}, \tag{4.25}$$

and flow tangency (or wall) boundary condition

$$(v_x\hat{\imath} + v_y\hat{\jmath}) \cdot \boldsymbol{n}_{fs}|_{\Gamma_{fs}} = 0, \tag{4.26}$$

where $\boldsymbol{u}_f(x,\, y,\, t)$ is the vector of conserved variables, $\mathscr{F}(x,\, y,\, t)$ and $\mathscr{G}(x,\, y,\, t)$ are the flux vectors in the X and Y coordinate directions, and $\frac{\partial \mathscr{F}}{\partial \boldsymbol{u}_f}$ and $\frac{\partial \mathscr{G}}{\partial \boldsymbol{u}_f}$ are the respective flux Jacobian matrices. The state vector and flux vectors are as given below.

$$\boldsymbol{u}_f = \left\{ \begin{array}{c} \rho \\ \rho v_x \\ \rho v_y \\ \rho e_t \end{array} \right\}, \quad \mathscr{F} = \left\{ \begin{array}{c} \rho v_x \\ p + \rho v_x^2 \\ \rho v_x v_y \\ \rho v_x h_t \end{array} \right\}, \quad \mathscr{G} = \left\{ \begin{array}{c} \rho v_y \\ \rho v_x v_y \\ p + \rho v_y^2 \\ \rho v_y h_t \end{array} \right\}. \tag{4.27}$$

Variables ρ, p, v_x, v_y, $e_t = \left(\frac{1}{\gamma - 1}\frac{p}{\rho} + \frac{(v_x\hat{\imath} + v_y\hat{\jmath})^2}{2}\right)$, and $h_t = e_t + \frac{p}{\rho}$ denote density, pressure, horizontal velocity, vertical velocity, total energy, and total enthalpy in the domain, respectively. The pressure and density can be related to the temperature T by the equation of state $p = \rho \mathcal{R} T$, where $\mathcal{R}$ is the specific gas constant.

Equation (4.23) is nonlinear and comparing it to the general Eq. (4.1), we can derive the fluid operator and the source term

$$\boldsymbol{\mathcal{A}}_f(\boldsymbol{u}_f,\, L_f(\boldsymbol{u}_f)) = \left\{ \begin{array}{c} \boldsymbol{\mathcal{A}}_{f1} \\ \boldsymbol{\mathcal{A}}_{f2} \\ \boldsymbol{\mathcal{A}}_{f3} \\ \boldsymbol{\mathcal{A}}_{f4} \end{array} \right\}, \tag{4.28}$$

$$\boldsymbol{\mathcal{A}}_{f1} = \frac{\partial}{\partial x}\left(\rho v_x\right) + \frac{\partial}{\partial y}\left(\rho v_y\right), \tag{4.29}$$

$$\boldsymbol{\mathcal{A}}_{f2} = \frac{\partial}{\partial x}\left(p + \rho v_x^2\right) + \frac{\partial}{\partial y}\left(\rho v_x v_y\right), \tag{4.30}$$

$$\boldsymbol{\mathcal{A}}_{f3} = \frac{\partial}{\partial x}\left(\rho v_x v_y\right) + \frac{\partial}{\partial y}\left(p + \rho v_y^2\right), \tag{4.31}$$

$$\boldsymbol{\mathcal{A}}_{f4} = \frac{\partial}{\partial x}\left(\rho v_x h_t\right) + \frac{\partial}{\partial y}\left(\rho v_y h_t\right), \tag{4.32}$$

$$\boldsymbol{f}_f\left(\boldsymbol{x}; b\right) = 0. \tag{4.33}$$

The farfield boundary condition implies that $\boldsymbol{u}_{f\infty}$ is the prescribed state at the farfield boundary Γ_{ef}. The flow tangency boundary condition implies that the velocity vector $(v_x\hat{\imath} + v_y\hat{\jmath})$ has no component along the unit normal $\boldsymbol{n}_{fs}$ on the airfoil boundary Γ_{fs}. The farfield and wall boundaries for flow over an airfoil are shown in Figure 4.2. Note that the normal $\boldsymbol{n}_{fs}$ and the deformed airfoil boundary Γ_{fs} are functions of the structural response $\boldsymbol{u}_s$, as described in the next section.

4.2.4.3 Coupling of the Fluid and Structural Responses

Coupling of the aerodynamic and structural responses occurs because the loads on the structure come from the fluid response, whereas the fluid flow tangency boundary condition is obtained from the deformed structure. If a weak coupling strategy is followed, normally satisfactory for steady-state aeroelastic response, boundary loads for the airfoil structure are derived from the pressure distribution on the airfoil, and the pressure distribution depends on the deformed shape of the airfoil. This coupling can be represented as shown below.

$$\boldsymbol{\Phi}|_{\Gamma_{ns}} = \bar{\boldsymbol{\Phi}} = -p\boldsymbol{n}_{fs}|_{\Gamma_{fs}} \tag{4.34}$$

$$\Gamma_{fs} : \left\{ \begin{array}{cc} x_{fs} & y_{fs} \end{array} \right\}^T = \Gamma_{ns} : \left\{ \begin{array}{cc} x_s & y_s \end{array} \right\}^T \oplus \boldsymbol{u_s} : \left\{ \begin{array}{cc} u_x & u_y \end{array} \right\}^T, \text{ thus} \tag{4.35}$$

$$x_{fs} = x_s + u_x, \tag{4.36}$$

$$y_{fs} = y_s + u_y. \tag{4.37}$$

Eq. (4.34) defines the traction loads $\bar{\bar{\mathbf{\Phi}}}$ on the (undeformed) airfoil shape. A negative sign appears in Eq. (4.34) because the pressure acts in a direction normal to the airfoil surface from the fluid to the structural domain, whereas the normal vector $\boldsymbol{n}_{fs}$ is assumed to be in the opposite direction (i.e. from the structural domain to the fluid domain). Since pressure p is a fluid response, it depends on the flow analysis solution $\boldsymbol{u}_f$. Eq. (4.35) represents that the deformed airfoil shape Γ_{fs} is dependent on the structural response $\boldsymbol{u}_s$. Eqs. (4.36–4.37) are used to get the deformed airfoil shape, on which to impose the fluid no-penetration boundary condition (4.26).

In general, the normal $\boldsymbol{n} = \{n_x,\, n_y\}$ for a parameterized curve $\Gamma : \{x\,(b)\,,\, y\,(b)\}^T$ is given by

$$n_x\,(x,\, y) = -\frac{s}{\sqrt{1+s^2}}, \tag{4.38}$$

$$n_y\,(x,\, y) = \frac{1}{\sqrt{1+s^2}}, \tag{4.39}$$

where $s\,(x,\, y,\, b) = \frac{dy}{dx}\,(x,\, y,\, b)$ is the slope of Γ. Eqs. (4.38–4.39) can be used to get the normal $\boldsymbol{n}_{fs} = \{n_{xfs},\, n_{yfs}\}$ for the deformed fluid-structure interface $\Gamma_{fs} : \{x_{fs},\, y_{fs}\}^T$ or the normal $\boldsymbol{n}_s = \{n_{xs},\, n_{ys}\}$ for the undeformed structural boundary $\Gamma_s : \{x_s,\, y_s\}^T$. The dependence of Γ_{fs} and $\boldsymbol{n}_{fs}$ on $\boldsymbol{u}_s$ clearly couples the fluid and structural responses.

4.2.4.4 Results

The coupled analysis was done using SU2 (finite volume vertex based solver) for flow analysis and Nastran (finite element solver) for structural analysis. The flow and structural meshes used for the analysis are shown in Figures 4.3 and 4.4, respectively. The flow O-mesh was adopted from Vassberg and Jameson (2010), with 64×64 quadrilateral cells, and the

Table 4.1: Details of the aeroelasticity analysis

Parameter	Value
Mach number, M	0.3
Angle of attack,α	1.25°
Reference chord length, C_{ref}	1.0
Reference moment center, X_{ref}	0.25
Free stream pressure, p_∞	101325 N/m^2
Free stream temperature, T_∞	288.15 K
Flux scheme	Roe's II order upwind
Convergence criteria	$\log_{10}$ of L_2 norm of continuity residual less than -10
Thickness of airfoil (along Z)	0.1 m
Young's modulus, E	5.0E7 N/m^2
Poisson's ratio, ν	0.3

structural mesh, created using Triange tool (Shewchuk, 1996), had 138 CTRIAR triangular Nastran finite elements. The details of the aeroelasticity analysis are shown in Table 4.1. The weak coupling converges in about 10 iterations. Convergence of the coupled analysis is shown in the form of difference in the successive values of coefficient of lift, C_L (Figure 4.5) and coefficient of drag, C_D (Figure 4.6). At the steady-state $C_L = 0.17706$ and $C_D = 7.048E-3$. The steady-state flow and structural responses on the airfoil are shown in Figure 4.7. The movement of the flexible fluid-structure interface due to structural deformation is plotted in Figure 4.8. It can be observed that the boundary moves significantly only during the first two or three iterations. The structural deformation $\boldsymbol{u_s} = \{u_x, u_y\}^T$ at the steady-state response can be visualized in a vector plot as shown in Figure 4.9. This illustrates that the airfoil boundary is deformed mainly in the transverse direction during the weak coupling iterations.

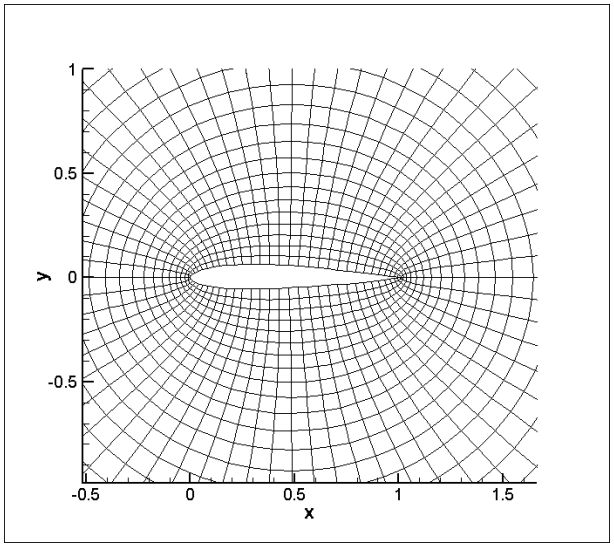

Figure 4.3: Close-up of O-mesh around the flexible airfoil with 64×64 cells (Vassberg and Jameson, 2010).

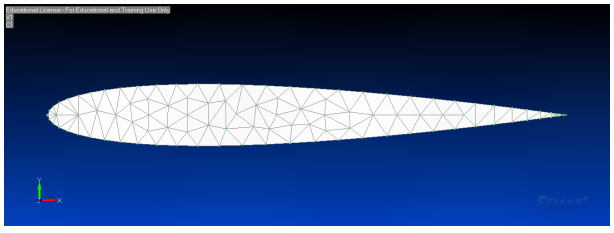

Figure 4.4: Structural mesh for the flexible airfoil with 138 triangular finite elements.

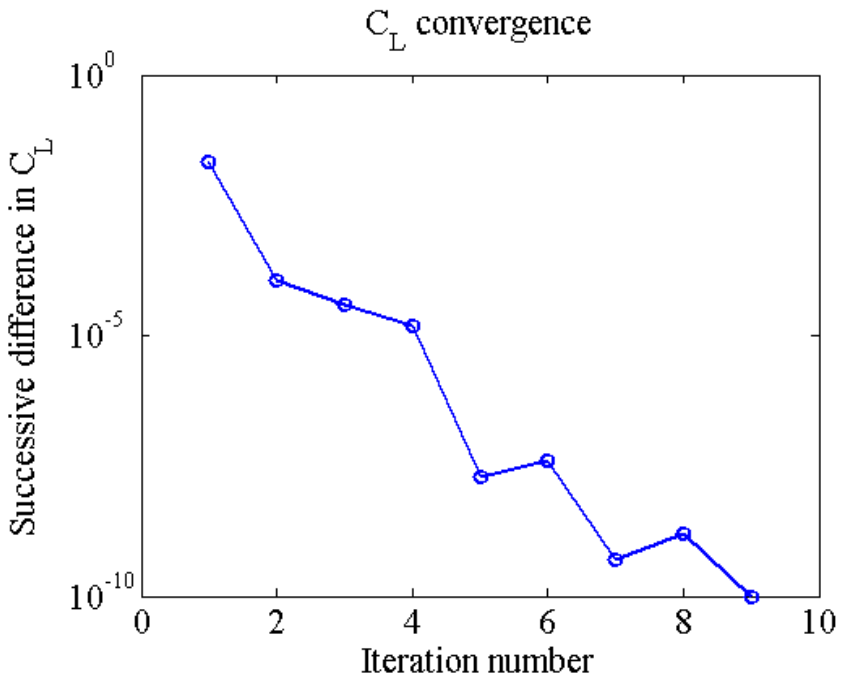

Figure 4.5: Difference in successive values of C_L for weak coupling iterations. The successive difference for the last six iterations is exactly zero, hence not shown on the log scale of the Y axis in the plot.

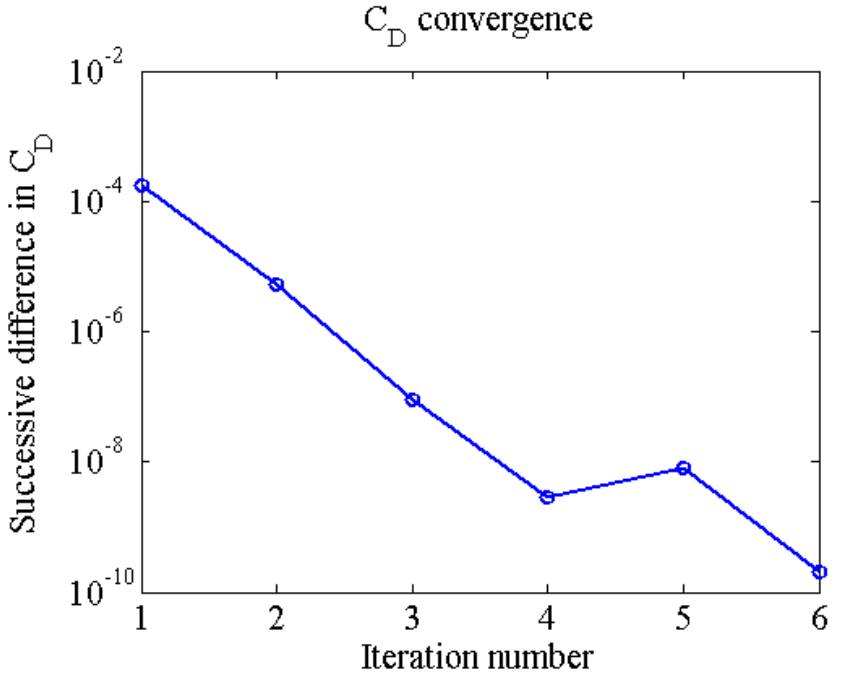

Figure 4.6: Difference in successive values of C_D for weak coupling iterations. The successive difference for the last nine iterations is exactly zero, hence not shown on the log scale of the Y axis in the plot.

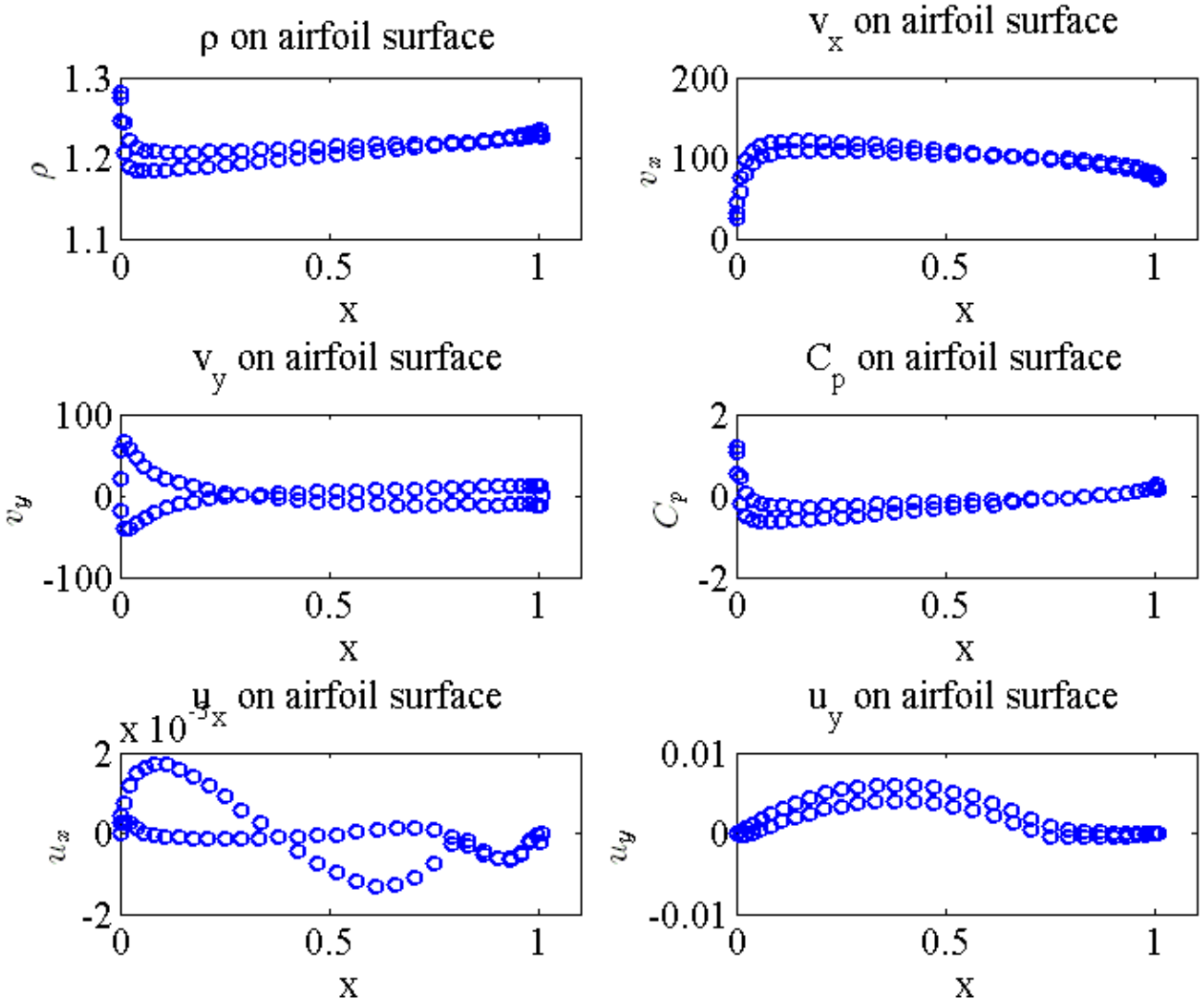

Figure 4.7: Aeroelastic response of the flexible airfoil.

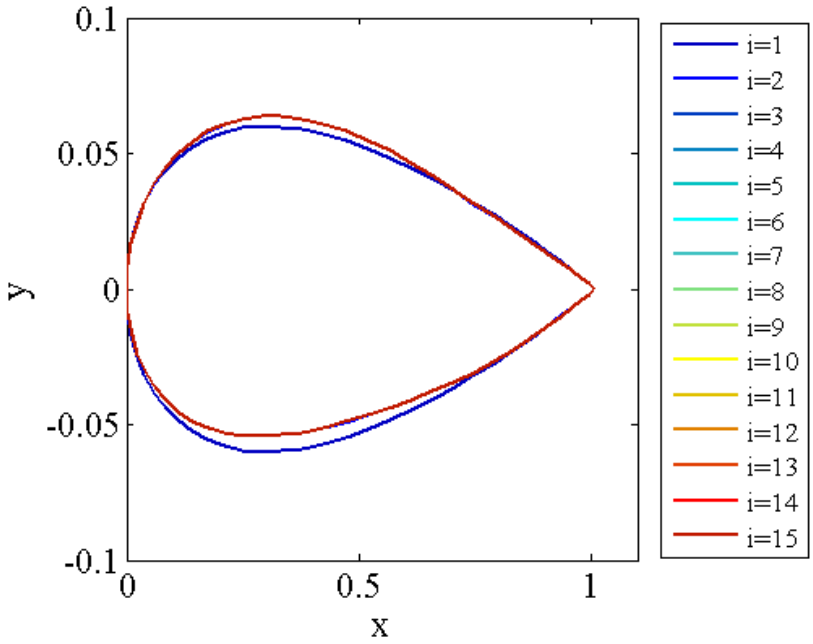

Figure 4.8: Airfoil boundary movement during weak coupling iterations.

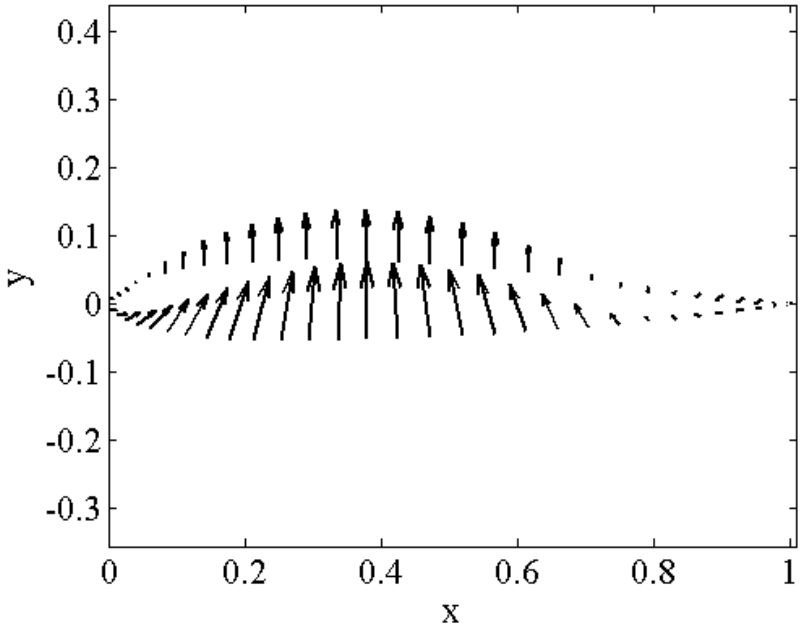

Figure 4.9: Vector plot to illustrate structural deformation of the flexible fluid-structure interface at steady-state. The deformation is with respect to the undeformed NACA 0012 shape. Vector lengths indicate magnitude of displacements and are scaled for better visualization.

4.2.5 Two-dimensional Coupled Fluid-Structure Sensitivity Analysis

4.2.5.1 Structural Response Sensitivity Equations

The CSEs for the 2D problem are derived by partial differentiation of Eq. (4.14)

$$[\boldsymbol{\partial}]\,\mathcal{D}\,[\boldsymbol{\partial}]^T\,\boldsymbol{u}_s' = \boldsymbol{F}'. \tag{4.40}$$

Comparing Eq. (4.40) to the general Eq. (4.8), we confirm that for the linear structural analysis considered here, the PDE operator for the structural local derivatives is same as the PDE operator for the structural response, i.e. $\boldsymbol{\mathcal{A}}_{sb} = \boldsymbol{\mathcal{A}}_s$.

According to (4.7), the structural CSE essential (geometric) boundary conditions would be

$$\boldsymbol{u}_s'|_{\Gamma_{es}} = \dot{\bar{\boldsymbol{u}}}_s - \boldsymbol{\nabla}_{\boldsymbol{x}}\left(\boldsymbol{u}_s\right).\boldsymbol{\mathcal{V}}|_{\Gamma_{es}}\,. \tag{4.41}$$

For the example of the flexible airfoil,

$$\boldsymbol{u}_{\boldsymbol{s}}'\Big|_{\Gamma_e^0} = 0, \tag{4.42}$$

$$\frac{1}{2}\left(\frac{\partial u_y'}{\partial x} - \frac{\partial u_x'}{\partial y}\right)\Bigg|_{\Gamma_e^0} = 0. \tag{4.43}$$

These BCs (4.42–4.43) are homogeneous because the shape design variable considered here changes only the shape of the airfoil, whereas the boundary Γ_{es} is unchanged, leading to zero design velocity there.

The structural CSE nonessential (natural) boundary conditions, will be

$$\boldsymbol{\Phi}'|_{\Gamma_{ns}} = [\boldsymbol{\sigma}']\,\boldsymbol{n}_s|_{\Gamma_{ns}} = \dot{\bar{\boldsymbol{\Phi}}} - \left(\mathcal{V}_{xs}\frac{\partial\,[\boldsymbol{\sigma}]}{\partial x} + \mathcal{V}_{ys}\frac{\partial\,[\boldsymbol{\sigma}]}{\partial y}\right)\boldsymbol{n}_s - [\boldsymbol{\sigma}]\,\dot{\boldsymbol{n}}_s\Big|_{\Gamma_{ns}}, \tag{4.44}$$

where

$$\begin{aligned} \dot{\bar{\Phi}} &= -\dot{p}\boldsymbol{n}_{fs} - p\dot{\boldsymbol{n}}_{fs}|_{\Gamma_{fs}}, \\ \frac{\partial[\boldsymbol{\sigma}]}{\partial x_i} &= \frac{\partial}{\partial x_i}\begin{bmatrix} \sigma_x & \tau_{xy} \\ \tau_{xy} & \sigma_y \end{bmatrix}, \end{aligned}$$

$\boldsymbol{\mathcal{V}}_s = \{\mathcal{V}_{xs}, \mathcal{V}_{ys}\}^T$ is the design velocity of the undeformed structure Γ_{ns}, and $\dot{\boldsymbol{n}}_s = \frac{D\boldsymbol{n}_s}{Db}$ and $\dot{\boldsymbol{n}}_{fs} = \frac{D\boldsymbol{n}_{fs}}{Db}$ are the material derivatives of the unit normals on Γ_{ns} and Γ_{fs}, respectively. Both these variables may be calculated based on the parameterization of the airfoil shape definition Γ_{ns}. The natural CSE BC (4.44) is discussed in detail in Section 4.2.5.3. For a given $\dot{p}$ and $\dot{\boldsymbol{n}}_{fs}$, solution of Eqs. (4.40–4.44) yields the local structural derivatives $\boldsymbol{u}'_s$. Addition of convective term to these yields the total structural derivatives $\dot{\boldsymbol{u}}_s$.

4.2.5.2 Fluid Response Sensitivity Equations

The CSEs for the flow problem are given by

$$\frac{\partial \mathscr{F}'}{\partial x} + \frac{\partial \mathscr{G}'}{\partial y} = 0, \tag{4.45}$$

where the state vector $\boldsymbol{u}'_f(x, y, t)$ and flux vectors $\mathscr{F}'(x, y, t)$ and $\mathscr{G}'(x, y, t)$ are

$$\boldsymbol{u}'_f = \begin{Bmatrix} \rho' \\ (\rho v_x)' \\ (\rho v_y)' \\ (\rho e_t)' \end{Bmatrix}, \quad \mathscr{F}' = \begin{Bmatrix} (\rho v_x)' \\ p' + (\rho v_x)' v_x + (\rho v_x) v'_x \\ (\rho v_x)' v_y + (\rho v_x) v'_y \\ (\rho h_t)' v_x + (\rho h_t) v'_x \end{Bmatrix}, \quad \mathscr{G}' = \begin{Bmatrix} (\rho v_y)' \\ (\rho v_x)' v_y + (\rho v_x) v'_y \\ p' + (\rho v_y)' v_y + (\rho v_y) v'_y \\ (\rho h_t)' v_y + (\rho h_t) v'_y \end{Bmatrix}. \tag{4.46}$$

Although the Euler Eqs. (4.23) are nonlinear, the CSEs (4.45) are linear in the sensitivity variables $\boldsymbol{u}'_f$. Comparing Eq. (4.45) to the general Eq. (4.8), we confirm that although the governing fluid Eqs. (4.23) are nonlinear with the nonlinear operator (4.28), the PDE

operator $\boldsymbol{\mathcal{A}}_{fb}$ for the CSEs is linear in the sensitivity variables $\boldsymbol{u}'_f$, and can be derived as

$$\boldsymbol{\mathcal{A}}_{fb}\left(\boldsymbol{u}_f,\, L_f\left(\boldsymbol{u}'_f\right)\right) = \left\{ \begin{array}{c} \boldsymbol{\mathcal{A}}_{fb1} \\ \boldsymbol{\mathcal{A}}_{fb2} \\ \boldsymbol{\mathcal{A}}_{fb3} \\ \boldsymbol{\mathcal{A}}_{fb4} \end{array} \right\}, \tag{4.47}$$

$$\begin{aligned}
\boldsymbol{\mathcal{A}}_{fb1} &= \frac{\partial}{\partial x}\left(\left(\rho v_x\right)'\right) + \frac{\partial}{\partial y}\left(\left(\rho v_y\right)'\right), && (4.48)\\
\boldsymbol{\mathcal{A}}_{fb2} &= \frac{\partial}{\partial x}\left(p' + \left(\rho v_x\right)' v_x + \left(\rho v_x\right) v'_x\right) + \frac{\partial}{\partial y}\left(\left(\rho v_x\right)' v_y + \left(\rho v_x\right) v'_y\right), && (4.49)\\
\boldsymbol{\mathcal{A}}_{fb3} &= \frac{\partial}{\partial x}\left(\left(\rho v_x\right)' v_y + \left(\rho v_x\right) v'_y\right) + \frac{\partial}{\partial y}\left(p' + \left(\rho v_y\right)' v_y + \left(\rho v_y\right) v'_y\right), && (4.50)\\
\boldsymbol{\mathcal{A}}_{fb4} &= \frac{\partial}{\partial x}\left(\left(\rho h_t\right)' v_x + \left(\rho h_t\right) v'_x\right) + \frac{\partial}{\partial y}\left(\left(\rho h_t\right)' v_y + \left(\rho h_t\right) v'_y\right), && (4.51)\\
\boldsymbol{f}'_f\left(\boldsymbol{x}; b\right) &= 0. && (4.52)
\end{aligned}$$

The local derivatives of flow velocities and pressure can be written in terms of $\boldsymbol{u}'_f$ as

$$\begin{aligned}
v'_x &= \frac{1}{\rho}\left(\left(\rho v_x\right)' - \frac{\left(\rho v_x\right)}{\rho}\rho'\right), && (4.53)\\
v'_y &= \frac{1}{\rho}\left(\left(\rho v_y\right)' - \frac{\left(\rho v_y\right)}{\rho}\rho'\right), && (4.54)
\end{aligned}$$

$$p' = \left(\gamma - 1\right)\left[\left(\rho e_t\right)' - \frac{1}{2}\rho'\left(\left(\frac{\rho v_x}{\rho}\right)^2 + \left(\frac{\rho v_y}{\rho}\right)^2\right) - \left(\left(\rho v_x\right) v'_x + \left(\rho v_y\right) v'_y\right)\right]. \tag{4.55}$$

The farfield CSE boundary condition is

$$\boldsymbol{u}'_f\Big|_{\Gamma_{ef}} = 0 - \frac{\partial \boldsymbol{u}_f}{\partial x}\mathcal{V}_{xf}\Big|_{\Gamma_{ef}} - \frac{\partial \boldsymbol{u}_f}{\partial y}\mathcal{V}_{yf}\Big|_{\Gamma_{ef}} = 0. \tag{4.56}$$

The shape design variable considered here only changes shape of the airfoil boundary which results in $\mathcal{V}_{xf}|_{\Gamma_{ef}} = \mathcal{V}_{yf}|_{\Gamma_{ef}} = 0$ and thus, the boundary condition (4.56) is homogeneous. Taking material derivative of the no-penetration (tangency) flow BC (4.26) yields the CSE BC at the airfoil boundary Γ_{fs}

$$\left(v'_x\hat{\imath} + v'_y\hat{\jmath}\right)\cdot\boldsymbol{n}_{fs}\Big|_{\Gamma_{fs}} = -\left(\left\{\frac{\partial v_x}{\partial x}\mathcal{V}_{xfs} + \frac{\partial v_x}{\partial y}\mathcal{V}_{yfs}\right\}\hat{\imath} + \left\{\frac{\partial v_y}{\partial x}\mathcal{V}_{xfs} + \frac{\partial v_y}{\partial y}\mathcal{V}_{yfs}\right\}\hat{\jmath}\right)\cdot\boldsymbol{n}_{fs} - (v_x\hat{\imath} + v_y\hat{\jmath})\cdot\dot{\boldsymbol{n}}_{fs}\Big|_{\Gamma_{fs}} \tag{4.57}$$

where $\boldsymbol{\mathcal{V}}_{fs} = \{\mathcal{V}_{xfs}, \mathcal{V}_{yfs}\}^T$ is the design velocity and $\dot{\boldsymbol{n}}_{fs} = \frac{D\boldsymbol{n}_{fs}}{Db}$ is the material derivative of the unit normal, both defined at the deformed fluid-structure interface Γ_{fs}. The CSE boundary condition (4.57) is called transpiration boundary condition, because the velocity local derivative vector $\left(v'_x\hat{\imath} + v'_y\hat{\jmath}\right)$ has a nonzero component along normal direction of the boundary Γ_{fs}. The terms $\dot{\boldsymbol{n}}_{fs}$ and $\boldsymbol{\mathcal{V}}_{fs}$ in Eq. (4.57) are sources of the coupling between the fluid and structural derivatives, and are derived in the following sub-section.

4.2.5.3 Coupling of the Fluid and Structural Derivatives

The coupling of the flow and structural derivatives occurs as follows:

- *Sensitivity loads on the structure depend on the fluid response and its derivative:* Substituting material derivative of Eq. (4.34) in Eq. (4.44), we get

$$\boldsymbol{\Phi}'|_{\Gamma_{ns}} = -\dot{p}\boldsymbol{n}_{fs} - p\dot{\boldsymbol{n}}_{fs}|_{\Gamma_{fs}} - \left(\mathcal{V}_{xs}\frac{\partial[\boldsymbol{\sigma}]}{\partial x} + \mathcal{V}_{ys}\frac{\partial[\boldsymbol{\sigma}]}{\partial y}\right)\boldsymbol{n}_s - [\boldsymbol{\sigma}]\,\dot{\boldsymbol{n}}_s\Big|_{\Gamma_{ns}}. \tag{4.58}$$

  Thus, the structural sensitivity boundary loads are dependent on the pressure $p(\boldsymbol{u}_f)$ and derivative $\dot{p}(\dot{\boldsymbol{u}}_{\boldsymbol{f}})$. Here, $\dot{\boldsymbol{n}}_s = \frac{D\boldsymbol{n}_s}{Db}$ is defined for the undeformed boundary Γ_{ns} whereas $\dot{\boldsymbol{n}}_{fs} = \frac{D\boldsymbol{n}_{fs}}{Db}$ is defined for the deformed boundary Γ_{fs} .

- *Fluid CSE BC depends on structural response and its derivative:* The CSE transpiration BC (4.57) is to be applied on the deformed boundary $\Gamma_{fs} : \{x_{fs}, y_{fs}\}^T$ with the corresponding normal $\boldsymbol{n}_{fs}$, both of which are dependent on the structural response $\boldsymbol{u}_s$

as given in Eqs. (4.35–4.39). Along with this, the material derivative of the normal $\dot{\boldsymbol{n}}_{fs} = \{\dot{n}_{xfs},\, \dot{n}_{yfs}\}^T$ also appears in the transpiration BC (4.57). This term can be derived as follows. Taking the material derivative of Eqs. (4.38–4.39), specifically for the boundary $\Gamma_{fs} : \{x_{fs},\, y_{fs}\}^T$, we get

$$\dot{n}_{xfs}\,(x_{fs},\, y_{fs}) = -\frac{\dot{s}_{fs}}{\sqrt{1+s_{fs}^2}} + \frac{s_{fs}^2 \dot{s}_{fs}}{\left(1+s_{fs}^2\right)^{3/2}}, \tag{4.59}$$

$$\dot{n}_{yfs}\,(x_{fs},\, y_{fs}) = -\frac{s_{fs}\dot{s}_{fs}}{\left(1+s_{fs}^2\right)^{3/2}}, \text{ where} \tag{4.60}$$

$$\dot{s}_{fs}\,(x_{fs},\, y_{fs}) = \frac{\partial s_{fs}}{\partial b} + \frac{\partial s_{fs}}{\partial x_{fs}}\frac{\partial x_{fs}}{\partial b} + \frac{\partial s}{\partial y_{fs}}\frac{\partial y_{fs}}{\partial b}. \tag{4.61}$$

Eq. (4.61) can also be written in terms of design velocity as

$$\dot{s}_{fs} = s'_{fs} + \mathcal{V}_{xfs}\frac{\partial s_{fs}}{\partial x_{fs}} + \mathcal{V}_{yfs}\frac{\partial s_{fs}}{\partial y_{fs}}. \tag{4.62}$$

Differentiation of Eqs. (4.36–4.37) with respect to the shape design variable b yields design velocity of the deformed fluid-structure interface

$$\mathcal{V}_{xfs} = \mathcal{V}_{xs} + \dot{u}_x, \tag{4.63}$$

$$\mathcal{V}_{yfs} = \mathcal{V}_{xs} + \dot{u}_y. \tag{4.64}$$

This design velocity appears in the fluid CSE BC (4.57), which clearly couples the structural and fluid derivatives.

4.2.5.4 Results

A finite difference study was carried out to with a step size of 0.0001 to obtain the flow and structural derivatives. These derivatives are plotted in Figure 4.10.

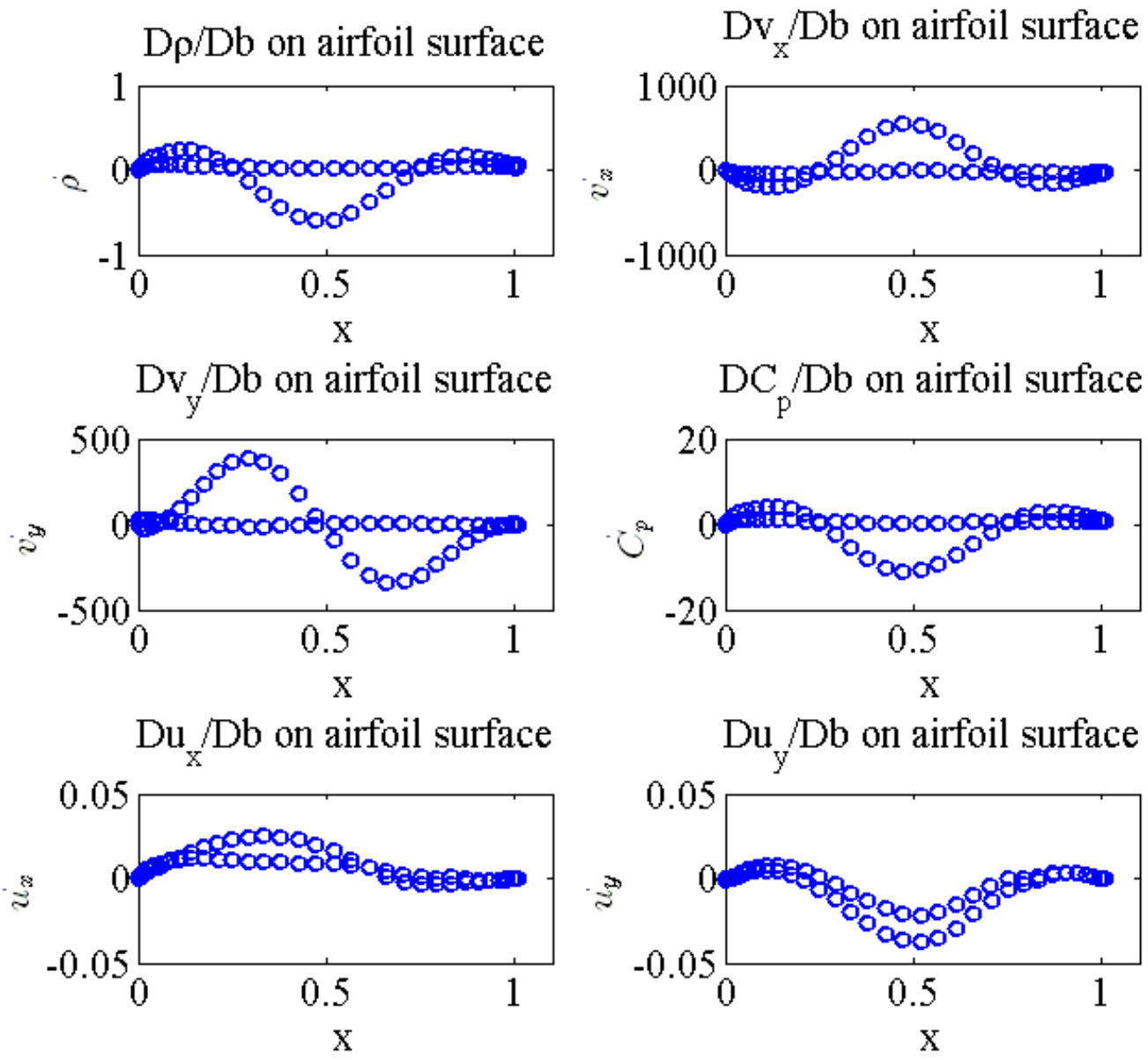

Figure 4.10: Derivatives of aeroelastic response of the flexible airfoil obtained using finite difference method.

4.3 Conclusion

We demonstrated a nonintrusive formulation of CSA for analyzing coupled fluid-structure systems, with linear structural analysis and Euler flow analysis. Similar to the coupling between the structural and fluid degrees of freedom, the structural and fluid variable derivatives are also coupled. We derived the sensitivity equations for the structural and fluid systems, and identified the sources of the coupling between the structural and fluid derivatives. This was demonstrated with an example of a flexible airfoil subjected to uniform subsonic flow. The interface that separates the fluid and structural domains was chosen to be flexible. This led to coupling terms in the sensitivity analysis which were highlighted. In particular, the structural sensitivity boundary loads are dependent on the pressure and its derivative on the flexible airfoil. Along with this, the fluid sensitivity transpiration boundary condition was applied on the flexible fluid-structure interface. This results in the fluid sensitivity boundary conditions being dependent on the structural deformation of the flexible interface and its derivatives. The coupling in the primary analysis and sensitivity analysis was achieved with a weak formulation, with data transfer between the structural and flow analysis codes during each iteration until convergence. The main contribution of this work, compared to the few aeroelastic CSA applications in the literature (Etienne et al., 2007; Wickert and Canfield, 2008; Liu and Canfield, 2013a, 2016; Cross and Canfield, 2014), is the nonintrusive formulation of local CSA with SGR for the aeroelasticity problem including finite volume CFD.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The three proposed research objectives for this PhD work were:

1. Formulate nonintrusive CSA with SGR for computation of high-fidelity flow derivatives, and demonstrate its application using CFD tools that are based on finite volume discretizations.

2. Develop hybrid adjoint formulation of CSA.

3. Formulate and demonstrate nonintrusive CSA with SGR for aeroelastic response derivatives involving high-fidelity flow and structural computation.

These goals have been accomplished as described in Chapters 2, 3 and 4, respectively. A summary of the contributions corresponding to each of these tasks is given next.

5.1.1 CSA for Fluid Applications

The method of CSA with SGR is used for calculating the material derivatives of 1-D and 2-D flows with respect to shape design parameters, with a focus on the flow solutions of compressible Euler equations and incompressible Navier-Stokes equations. One of the main

contributions of this work is the derivation of local CSA with SGR using finite volume formulation. An example of flow over a NACA0012 airfoil is presented that highlights the effect of the accuracy of the sensitivity boundary conditions on the derivatives of integrated flow quantities such as lift and drag. The spatial gradients of flow velocities, calculated using SGR, contribute significantly to the transpiration sensitivity boundary condition and thus affect the accuracy of total derivatives of the flow variables. Apart from this, the weak imposition of boundary conditions, which is typical in finite volume formulations, leads to errors in the solution to the sensitivity equations. It is believed that this may be one of the reasons affecting convergence of flow derivatives obtained using CSA. The CSA results are verified with the derivatives calculated using the automatic differentiation method and the finite difference method.

Another contribution of this work is the nonintrusive (without modifying the "black-box" analysis source code) implementation of CSA for analyzing fluid systems even in cases where consistent Jacobian may not be available. Particularly, we focus on the use of commonly used CFD codes, which use finite volume discretization, for solving the flow variables and their shape sensitivities nonintrusively. The main advantage of the nonintrusive approach is the savings in time and efforts required to implement a sensitivity analysis algorithm which otherwise requires programming, such as for the discrete (direct and adjoint) analytic methods. Apart from this it also offers a possibility to use software, such as Fluent or Nastran, that are not accessible for modification to implement a user specific (custom) sensitivity analysis method. However, many CFD and structural analysis software already have discrete analytic methods for performing sensitivity analysis. The advantage of the proposed CSA over such methods is three-fold, (a) since CSA is based on continuum equations, it can possibly yield a derivative that is close to the true continuum derivative, rather than the derivative that discrete methods may yield (assuming the two derivatives are different), (b) ability to use Jacobian different from the one being used for flow analysis, and (c) possible increase in efficiency of sensitivity analysis because CSEs are linear.

It was shown that closely matching derivatives are obtained for derivatives of compressible inviscid flow even with the use of inconsistent flow solution and Jacobian matrix. This differs from the cases of nonintrusive CSA reported in literature (Godfrey and Cliff, 2001; Borggaard and Burns, 1994, 1997). Godfrey and Cliff (2001) studied inconsistency of turbulence modeling in the flow and sensitivity equations. Also, they used finite difference method to compute the spatial gradients in the sensitivity analysis, which is reported to give poor results for unstructured meshes (Duvigneau and Pelletier, 2006; Becker and Ashcroft, 2014).

A variety of examples were used to illustrate the nonintrusive application of CSA. We presented the lid-driven cavity flow solved using Navier-Stokes (incompressible and viscous) equations with a structured grid, finite difference spatial discretization, and an explicit temporal discretization. Flow over a NACA0012 airfoil example was solved using Euler equations with a unstructured grid, finite volume spatial discretization, and an implicit temporal discretization. A value parameter sensitivity example was also presented which involved quasi-1-D flow in a convergent-divergent nozzle.

5.1.2 Hybrid Adjoint CSA Formulation

A new continuum-discrete hybrid adjoint formulation was introduced that has the following advantages: (a) it involves solving a linear discrete system only once for calculating derivatives of a single performance criterion with respect to many design variables, (b) it enjoys the benefits of local CSA with SGR, namely, the derivatives are accurate and the mesh sensitivity is avoided, and (c) unlike the continuous adjoint method, there are no complications associated with adjoint boundary conditions. This method was demonstrated with applications to 1-D, 2-D and 3-D structural problems. Application of this method to other disciplines, such as fluid analysis or coupled fluid-structure analysis is straightforward. The hybrid adjoint CSA method computes the same values for shape derivatives as direct CSA. Therefore accuracy and convergence properties are the same as for the direct local CSA. Finally, we show that for many design variables the hybrid adjoint formulation permits computing shape

derivatives with less effort as compared to the direct CSA.

5.1.3 CSA for Aeroelasticity Applications

We demonstrated a nonintrusive implementation of CSA for analyzing coupled fluid-structure systems, with linear structural analysis and Euler flow analysis. Similar to the coupling between the structural and fluid degrees of freedom, the structural and fluid variable derivatives are also coupled. We derived the sensitivity equations for the structural and fluid systems, identified the sources of the coupling between the structural and fluid derivatives, and implemented CSA nonintrusively to obtain the aeroelastic response derivatives. This was demonstrated with an example of a flexible airfoil subjected to uniform subsonic flow. Particularly, the interface that separates the fluid and structural domains was chosen to be flexible. This led to coupling terms in the sensitivity analysis which were highlighted. In particular, the structural sensitivity boundary loads were dependent on the pressure and its derivative on the flexible airfoil. Along with this, the fluid sensitivity transpiration boundary condition is applied on the flexible fluid-structure interface. This results in the fluid sensitivity boundary conditions being dependent on the structural deformation of the flexible interface and its derivatives. The coupling in the primary analysis and sensitivity analysis was achieved with a weak formulation, with data transfer between the structural and flow analysis codes during each iteration until convergence. The main contribution of this work, compared to the few aeroelastic CSA applications in the literature, (Etienne et al., 2007; Wickert and Canfield, 2008; Liu and Canfield, 2013a, 2016; Cross and Canfield, 2014) is the nonintrusive implementation of local CSA with SGR for the aeroelasticity problem that includes flow sensitivity calculations done with finite volume discretization. Thus, the contributions of solving the flow CSEs externally using consistent or inconsistent Jacobian, also apply to the aeroelasticity work.

5.2 Code Development

The following primary codes were developed as a part of the research conducted during this PhD work. There are also other supplementary codes, such as those for meshing and parsing data, which are not mentioned in this list.

- Complex step improvement to MstcGeom (MATLAB) and VT-CST (C++): With these improved codes, design velocities can be obtained for complex structures, with multiple interfaces and complex aerodynamic shapes (within the limitations of MstcGeom and VT-CST).

- 1-D, 2-D and 3-D SGR code for structured and unstructured meshes (MATLAB): This code does automatic n-layer patch detection and prescribes the known continuum values of the field variable at points in the patch wherever available. For example, if the the stress component σ_z is known (theoretically) to be zero at a surface, its value is taken to be zero at points in the patch on that surface, even if the values are nonzero due to discretization error. The spatial gradients are further used to prescribe CSA boundary conditions.

- SU2 code with CSA capability (C++): This code outputs the required Jacobian by running the SU2 solver for only one iteration using the restart feature. The analysis solution in the restart file could be either the one obtained from SU2 or from a different solver such as FLUENT.

- External solution of CSEs (MATLAB): This code sets up the CSE BCs and applies them using the weak formulation in the finite volume context. The CSEs are then solved exactly using the flow Jacobian (available a priori) with the MATLAB linear solver.

- Incompressible Navier-Stokes flow analysis and CSA solver (MATLAB)

- Direct and Hybrid Adjoint CSA solvers for 1-D, 2-D and 3-D structures (MATLAB): This suite of codes can be used to either do the sensitivity analysis inside MATLAB or externally using NASTRAN, with either the direct or the hybrid adjoint formulation.

- CSA for flexible airfoil problem (MATLAB): This code solves both the coupled primary analysis and coupled sensitivity analysis, with black-boxes being used for the structural (NASTRAN) and flow analysis (SU2) separately.

5.3 Publications and Presentations

Results from this research have been presented at the AIAA SciTech (Kulkarni et al., 2014b, 2015) and Aviation conferences (Kulkarni et al., 2014a). The conference papers for the years 2014 (SciTech), 2015 (Aviation), and 2016 (SciTech) were selected in the final rounds of the student paper competitions. The preliminary results of the hybrid adjoint research were published in the *AIAA Journal* as a technical note (Kulkarni et al., 2016). The research work on the reduced aerodynamics for flapping wing MAVs was published in the AIAA Structures, Structural Dynamics, and Materials Conference. Results were also presented during the 2012, 2013, 2014 and 2015 Technical Advisory Committee meetings of the Collaborative Center for Multidisciplinary Sciences at Virginia Tech. Apart from this, results were presented as posters, round-table discussions and talks during the Graduate Student Association Symposium at Virginia Tech (2015, 2016) and the Virginia Tech Interdisciplinary Research Society Symposium (2014, 2015).

5.4 Recommendations for Future Work

One of the lessons learned during this research was that the choice of SGR patch for CFD applications that use finite volume discretization may depend on flow and geometry specific information, and is critical for getting accurate spatial gradient predictions. For example,

in the case of reconstruction of spatial gradients at the trailing edge of an airfoil, if a vertex based finite volume discretization is used, a choice of data points in the patch downstream of the trailing edge can lead to unreasonable values of the spatial gradients. Hence it is necessary to avoid such data points. Similarly, for reconstruction of spatial gradients at points close to the trailing edge but on the upper surface of the airfoil, if the SGR patch has data points near the lower surface, erroneous results may be obtained. Hence at such expansion points on the top surface and close to the trailing edge, the SGR patch should have data points only near the top surface. As a result, improvements in the SGR code are one of the key recommendations for future work. Additionally, all the problems in this research are restricted to simplified problems with very few structural interfaces or material discontinuities. In a realistic high performance structure, such as an aircraft, there are many such interfaces that will complicate the CSA procedure due to the additional boundary conditions to be applied at these interfaces. Automation of the CSA process for many structural interfaces could be taken as a follow-up of the current work. Finally, all the problems solved using the hybrid adjoint method are restricted to steady-state or stationary analysis. Transient sensitivity analysis using the hybrid adjoint method could be another follow-up task.

Some specific recommendations are as follows:

- SGR improvements: Create a program for approximating spatial gradients in the direction normal to the boundary that changes shape. Another suggestion is to use a different method for approximating spatial gradients, such as the Green Gauss method, along the lines of Becker and Ashcroft (Becker and Ashcroft, 2014).

- Sensitivity of flow with discontinuities: The presented sensitivity analysis could be applied to a flow with discontinuities, such as flow with shocks. Some of the earlier work in this respect is done by Appel (Appel, 1997).

- Sensitivity for turbulent flow and viscous compressible flow: The current version of CSA

with SGR could be applied to turbulent flows with heat transfer. Results reported for this case (Colin et al., 2006) were shown to improve with adaptive re-meshing. Such re-meshing could potentially be avoided using the local CSA method with accurate spatial gradients found using SGR.

- Comparison with continuous adjoint: The presented hybrid adjoint method could be applied to fluid problems. This approach could be compared with the continuous adjoint approach reported by Economon et al. (Economon et al., 2015).

- Automation of CSA procedure for case with more interfaces: As described earlier, the implementation of the presented method becomes complicated if large number of interfaces are present. Thus, a way to automate the process for this case is suggested for future work.

- Transient sensitivity analysis with hybrid adjoint formulation: Such a formulation could be used to get time-accurate design derivatives for cases such as gust response of an aircraft.

- Monolithic formulation of the local CSA aeroelastic formulation: The present CSA formulation can be modified to obtain a monolithic (strongly coupled) formulation for solving sensitivity analysis for a coupled fluid-structure analysis problem, on the lines of Etienne and Pelletier (Etienne and Pelletier, 2005).

Bibliography

AIAA (2009). 4th aiaa cfd drag prediction workshop, san antonio, tx. San Antonio, TX. 54

Albring, T., Sagebaum, M., and Gauger, N. R. (2016). *A Consistent and Robust Discrete Adjoint Solver for the SU2 Framework - Validation and Application*, pages 77–86. Springer International Publishing, Cham. 13

Allison, D. L., Morris, C. C., Schetz, J. A., Kapania, R. K., Sultan, C., Watson, L. T., Deaton, J. D., and Grandhi, R. V. (2012). A multidisciplinary design optimization framework for design studies of an efficient supersonic air vehicle. In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSM*, number AIAA 2012-5492, Indianapolis, Indiana. AIAA. xiii, 6

Alyanak, E. J. (2012). Modeling for conceptual design: An aeroelastic approach. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, number AIAA 2012-1425, Honolulu, Hawaii. 8

Alyanak, E. J. and Kolonay, R. M. (2012). Efficient supersonic air vehicle structural modeling for conceptual design. In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSM*. 8

Anderson, J. D. (1995). *Computational Fluid Dynamics: The Basics with Applications*. Mcgraw Hill Series in Mechanical Engineering. McGraw-Hill, 1 edition. xx, 186

Anderson, W. K. and Bonhaus, D. L. (1994). An implicit upwind algorithm for computing turbulent flows on unstructured grids. *Computers and Fluids*, 23(1):1–21. 53

Ansari, S. A., Zbikowski, R., and Knowles, K. (2006a). Aerodynamic modelling of insect-like flapping flight for micro air vehicles. *Progress in Aerospace Sciences*, 42:129–172. 197, 199

Ansari, S. A., Zbikowski, R., and Knowles, K. (2006b). Non-linear unsteady aerodynamic model for insect-like flapping wings in the hover. part 1: Methodology and aanalysis. *Proceedings of the institute of mechanical engineering. Part G: J Aerospace Engineering*, 220(3):61–83. 198

Ansari, S. A., Zbikowski, R., and Knowles, K. (2006c). Non-linear unsteady aerodynamic model for insect-like flapping wings in the hover. part 2: Implementation and validation. *Proceedings of the institute of mechanical engineering. Part G: J Aerospace Engineering*, 220(3):169–186. 198

Appel, J. R. (1997). *Sensitivity Calculations For Conservation Laws With Application To Discontinuous Fluid Flows*. PhD thesis, Virginia Tech. 173

Arora, J. S. (1993). An exposition of the material derivatives approach for structural shape sensitivity analysis. *Computer Methods in Applied Mechanics and Engineering*, 105:41–62. 5, 78, 141

Arora, J. S. and Haug, E. J. (1979). Methods of design sensitivity analysis in structural optimization. *AIAA Journal*, 17(9):970–974. 86, 87, 88

Augarde, C. E. and Deeks, A. J. (2008). The use of timoshenko's exact solution for a cantilever beam in adaptive analysis. *Finite Elements in Analysis and Design*, 44(9Ű10):595–601. 106

Becker, K. and Ashcroft, G. (2014). A comparative study of gradient reconstruction meth-

ods for unstructured meshes with application to turbomachinery flows. In *52nd AIAA Aerospace Sciences Meeting, National Harbor, MD, USA*. 14, 169, 173

Berman, G. J. and Wang, Z. J. (2007). Energy-minimizing kinematics in hovering insect flight. *Journal of Fluid Mechanics*, 582:153–168. 197, 198

Borggaard, J. and Burns, J. (1994). A sensitivity equation approach to shape optimization in fluid flows. Nasa contractor report 191598, Langley Research Center. 5, 7, 12, 14, 20, 46, 84, 141, 148, 169, 191

Borggaard, J. and Burns, J. (1997). A pde sensitivity equation method for optimal aerodynamic design. *Journal of Computational Physics.*, 136:366–384. 5, 7, 12, 14, 20, 25, 46, 50, 78, 84, 141, 148, 169, 191

Charlot, L., Etienne, S., and Pelletier, D. (2012). A continuous lagrangian sensitivity equation method for incompressible flow. *Journal of Computational Physics*, 231:5989–6011. 11

Chen, P. C. and Zhang, Z. (2013). Store ejection loads analysis using an euler solver. In *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Boston, MA. AIAA. 50

Choi, K. K. and Kim, N.-H. (2005). *Structural sensitivity analysis and optimization.* Springer Science+Business media, New York. 5, 11, 78, 94, 141

Colin, E., Etienne, S., Pelletier, D., and Borggaard, J. (2006). A general sensitivity equation formulation for turbulent heat transfer. *Numerical Heat Transfer, Part B: Fundamentals*, 49(2):125–153. 174

Cross, D. M. and Canfield, R. A. (2012). Continuum shape sensitivity with spatial gradient reconstruction of nonlinear aeroelastic gust response. In *14th AIAA/ISSMO Multidiscip-*

linary Analysis and Optimization Conference, number AIAA 2012-55597, Indianapolis, Indiana. 25, 142

Cross, D. M. and Canfield, R. A. (2014). Local continuum shape sensitivity with spatial gradient reconstruction. *Structural and Multidisciplinary Optimization*, 50(6):975–1000. 7, 12, 18, 21, 78, 79, 84, 85, 88, 90, 92, 103, 141, 142, 146, 166, 170

Cross, D. M. and Canfield, R. A. (2015). Local continuum shape sensitivity with spatial gradient reconstruction for nonlinear analysis. *Structural and Multidisciplinary Optimization*, 51(4):849–865. 5, 90, 139

Cross, D. M. and Canfield, R. A. (2016). Convergence study of local continuum sensitivity method using spatial gradient reconstruction. *AIAA Journal*. 7, 22, 64, 66, 90

Dems, K. and Haftka, R. T. (1988). Two approaches to sensitivity analysis for shape variation of structures. *Mech. Struct. and Mach.*, 16(4):501. 5, 78, 141

Dems, K. and Mroz, Z. (1984). Variational approach by means of adjoint systems to structural optimization and sensitivity analysis. part ii - structure shape variation. *International Journal of Solids and Structures*, 20:527–552. 78

Dickinson, M. H., Lehmann, F.-O., and P, S. S. (1999). Wing rotation and the aerodynamic basis of insect flight. *Science*, 284:1954–1960. 197

Doman, D. B., Oppenheimer, M. W., and Sigthorsson, D. O. (2009). Dynamics and control of a minimally actuated biomimetic vehicle: Part i - aerodynamic model. In *AIAA Guidance, Navigation, and Control Conference*. 197

Dudley, R. (2000). *The Biomechanics of Insect Flight*. Princeton University Press. 198

Duivesteijn, G. F., Bijl, H., Koren, B., and van Brummelen, E. H. (2005). On the adjoint solution of the quasi-1d euler equations: The effect of boundary conditions and the numer-

ical flux function. *International Journal for Numerical Methods in Fluids*, 47(8-9):987–993. 8, 78, 139

Duvigneau, R. and Pelletier, D. (2006). On accurate boundary conditions for a shape sensitivity equation method. *International Journal for Numerical Methods in Fluids.*, 50(2):147–164. 3, 5, 7, 11, 12, 14, 21, 50, 78, 90, 141, 169

Economon, T. D., Palacios, F., and Alonso, J. J. (2015). Unsteady continuous adjoint approach for aerodynamic design on dynamic meshes. *AIAA Journal*, 53(9):2437–2453. 174

Ellington, C. P. (1984). The aerodynamics of hovering insect flight. v. a vortex theory. *Philosophical Transactions of the Royal Society of London. Series B, BiologicalSciences*, 305(1122):115–144. 198, 209

Etienne, S., Hay, A., and Garon, A. (2007). Sensitivity analysis of unsteady fluid-structure interaction problems. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, number AIAA-2007-332, Reno, Nevada. 166, 170

Etienne, S. and Pelletier, D. (2005). A general approach to sensitivity analysis of fluid-structure interactions. *Journal of Fluids and Structures*, 21(2):169–186. 5, 141, 174

Fritz, T. E. and Long, L. N. (2004). Object-oriented unsteady vortex lattice method for flapping flight. *Journal of Aircraft*, 41(6):1275–1290. 197, 205

Galinski, C. and Zbikowski, R. (2007). Some problems of micro air vehicles development. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 55(1):91–98. 197

Gay, D. M. (2005). Semiautomatic differentiation for efficient gradient computations. In Bücker, H. M., Corliss, G., Hovland, P., Naumann, U., and Norris, B., editors, *Automatic Differentiation: Applications, Theory, and Implementations*, Lecture Notes in Computational Science and Engineering, pages 147–158. Springer. 13

Ghommem, M. (2011). *Modeling and Analysis for Optimization of Unsteady Aeroelastic Systems*. PhD thesis, Virginia Tech. 198

Gobal, K., Grandhi, R. V., and Kolonay, R. M. (2015). Continuum sensitivity analysis for structural shape design variables using finite-volume method. *AIAA Journal*, 53(2):347–355. 12

Godfrey, A. G. and Cliff, E. M. (2001). Sensitivity equations for turbulent flows. In *39th AIAA Aerospace Sciences Meeting & Exhibit*, number 2001-1060, Reno, NV. 13, 14, 169

Haug, E. J., Choi, K. K., and Komkov, V. (1986). *Design sensitivity analysis of structural systems, Mathematics in science and engineering.*, volume 177. Academic Press, Orlando. 5, 141

Hicks, R. M. and Henne, P. A. (1978). Wing design by numerical optimization. *Journal of Aircraft*, 15(7):407–412. 44

Hirsch, C. (1990). *Numerical Computation of Internal and External Flows, Volume 2: Computational Methods for Inviscid and Viscous Flows*. John Wiley and Sons. 12, 14, 51, 68, 188, 194

Hogan, R. J. (2014). Fast reverse-mode automatic differentiation using expression templates in c++. *ACM Transactions on Mathematical Software*, 40(4):26:1–26:16. 13

Jameson, A. (1983). Solution of the euler equations for two dimensional transonic flow by a multigrid method. *Applied Mathematics and Computation*, 13(3):327 – 355. 53

Johnson, W. (1994). *Helicopter Theory*. Dover. 209

Keennon, M., Klingebiel, K., Won, H., and Andriukov, A. (2012). Development of the nano hummingbird: A tailless flapping wing micro air vehicle. In *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. 196, 197

Knight, D. (2006). *Elements of numerical methods for compressible flows*. Cambridge University Press. 194

Kulkarni, M. D., Canfield, R. A., and Patil, M. J. (2014a). Nonintrusive continuum sensitivity analysis for aerodynamic shape optimization. In *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number AIAA 2014-2043, Atlanta, GA. AIAA. 78, 172

Kulkarni, M. D., Canfield, R. A., and Patil, M. J. (2015). Discrete adjoint formulation for continuum sensitivity analysis. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, number AIAA 2015-0138, Kissimmee, FL. AIAA. 92, 172

Kulkarni, M. D., Canfield, R. A., Patil, M. J., and Alyahak, E. J. (2014b). Integration of geometric sensitivity and spatial gradient reconstruction for aeroelastic shape optimization. In *10th AIAA Multidisciplinary Design Optimization Conference*, number AIAA 2014-0470, National Harbor, MD. AIAA. 5, 24, 88, 172

Kulkarni, M. D., Cross, D. M., and Canfield, R. A. (2016). Discrete adjoint formulation for continuum sensitivity analysis. *AIAA Journal*. 17, 78, 92, 93, 145, 172

Kulkarni, M. D., Patil, M. J., and Snyder, R. D. (2013). Reduced order model for unsteady aerodynamics of flapping wing micro air vehicle in hover. In *54th AIAA Structures, Structural Dynamics and Materials Conference*, number AIAA 2014-1645, Boston, MA. AIAA. 196

Liu, S. and Canfield, R. A. (2011). Continuum shape sensitivity for nonlinear transient aeroelastic gust response. In *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, number AIAA 2011-1971, Denver, Colorado. 7

Liu, S. and Canfield, R. A. (2013a). Boundary velocity method for continuum shape sensit-

ivity of nonlinear fluid structure interaction problems. *Journal of Fluids and Structures*, 40:284 – 301. 7, 78, 90, 141, 166, 170

Liu, S. and Canfield, R. A. (2013b). Equivalence of continuum and discrete analytic sensitivity methods for nonlinear differential equations. *Structural and Multidisciplinary Optimization*, 48(6):1173–1188. 20, 46, 148, 191

Liu, S. and Canfield, R. A. (2016). Two forms of continuum shape sensitivity method for fluid structure interaction problems. *Journal of Fluids and Structures*, 62:46 – 64. 5, 7, 11, 18, 81, 141, 146, 166, 170

Lozano, C. and Ponsin, J. (2012). Remarks on the numerical solution of the adjoint quasi-one-dimensional euler equations. *International Journal for Numerical Methods in Fluids*, 69(5):966–982. 8, 78, 139

Nastran (2004). *MSC Nastran 2004, Reference Manual*. MSC Software. 151

Nastran (2010). *MD Nastran 2010, Design Sensitivity and Optimization User's Guide*. MSC Software. 106, 139

Noll, T. E., Brown, J. M., Perez-Davis, M. E., Ishmael, S. D., Tiffany, G. C., and Matthew, G. (2004). Investigation of the helios prototype aircraft mishap. Technical report, NASA. 2

Palacios, F., Michael R. Colonno, M. R., Aranake, A. C., Campos, A., Copeland, S. R., Economon, T. D., Lonkar, A. K., Lukaczyk, T. W., Taylor, T. W. R., and J., A. J. (2013). Stanford university unstructured (su2): An open-source integrated computational environment for multi-physics simulation and design. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, number AIAA 2013-0287, Grapevine, Texas. xiv, 12, 42

Pelletier, D., Hay, A., Etienne, S., and Borggaard, J. (2008). The sensitivity equation method in fluid mechanics. *European Journal of Computational Mechanics/Revue Europeenne de Mecanique Numerique*, 17(1-2):31–61. 11

Petricca, L., Ohlckers, P., and Grinde, C. (2011). Micro- and nano-air vehicles: State of the art. *International Journal of Aerospace Engineering*, 2011. 196

Pines, D. J. and Bohorquez, F. (2006). Challenges facing future micro-air-vehicle development. *Journal of Aircraft*, 43(2):290–305. 196, 197

Roy, C. J. and Hopkins, M. M. (2003). Discretization error estimates using exact solutions to nearby problems. In *41st Aerospace Sciences Meeting and Exhibit*, number AIAA 2003-0629, Reno, Nevada. 26

Sane, S. P. and Dickinson, M. H. (2002). The aerodynamic effects of wing rotation and a revised quasi-steady model of flapping flight. *The Journal of Experimental Biology*, 205:1087–1096. 197, 199

Sane, S. P. and Dickinson, M. H. (2003). The aerodynamics of insect flight. *The Journal of Experimental Biology*, 206:4191–4208. 199

Schafroth, D. M. (2010). *Aerodynamics, Modeling and Control of an Autonomous Micro Helicopter*. PhD thesis, ETH Zurich. 197

Shewchuk, J. R. (1996). *Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator*, pages 203–222. Springer Berlin Heidelberg, Berlin, Heidelberg. 155

Shyy, W., Lian, Y., Tan, J., Viieru, D., and Liu, H. (2008). *Aerodynamics of Low Reynolds Number Flyers*. Cambridge University Press. 198

Stanford, B. K. and Beran, P. S. (2010). Analytical sensitivity analysis of an unsteady vortex-lattice method for flapping-wing optimization. *Journal of Aircraft*, 47(2):647–662. 197, 198, 199, 205

Stanley, L. G. D. and Stewart, D. (2002). *Design sensitivity analysis : computational issues of sensitivity equation methods.* Academic Press, Philadelphia. 5, 7, 78, 141

Sun, M., Wang, J., and Xiong, Y. (2007). Dynamic flight stability of hovering insects. *Acta Mechanica Sinica*, 23:231–246. 198

Sun, M. and Xiong, Y. (2005). Dynamic flight stability of a hovering bumblebee. *The Journal of Experimental Biology*, 208:447–459. 198

The-Boeing-Company (2016). http://www.boeing.com/company/about-bca/. 1

Turgeon, E., Pelletier, D., and Borggaard, J. (2005). A general continuous sensitivity equation formulation for complex flows. *Numerical Heat Transfer Part B-Fundamentals*, 42(6):485–498. 5, 7, 141

Vassberg, J. C. and Jameson, A. (2010). In pursuit of grid convergence for two-dimensional euler solutions. *Journal of Aircraft*, 47(4):1152–1166. xiv, xx, 53, 55, 58, 60, 125, 154, 156

Villa, A. (2009). Convergence of weakly imposed boundary conditions: The one-dimensional hyperbolic case. *SIAM Journal on Scientific Computing*, 31(4):3116–3127. 12

Villa, A., Barbieri, L., and Malgesini, R. (2012). Ghost cell boundary conditions for the euler equations and their relationships with feedback control. *Communications in Applied and Industrial Mathematics*, 3(1). 12

Weis-Fogh, T. (1972). Energetics of hovering flight in hummingbirds and in drosophila. *The Journal of Experimental Biology*, 56:79–104. 197, 209

Wickert, D. P. and Canfield, R. A. (2008). Least-squares continuous sensitivity analysis of an example fluid-structure interaction problem. In *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, number AIAA 2008-1896, Schaumburg, IL. 166, 170

Wickert, D. P., Canfield, R. A., and Reddy, J. N. (2010). Least-squares continuous sensitivity shape optimization for structural elasticity applications. *AIAA Journal*, 48(12):2752–2762. 20, 46, 92, 93, 96, 100, 102, 148, 191

Woods, M. I., Henderson, J. F., and Lock, G. D. (2001). Energy requirements for the flight of micro air vehicles. *Aeronautical Journal*, 105(1043):135–149. 196

Yang, R. and Botkin, M. (1986). Comparison between the variational and implicit differentiation approaches to shape design sensitivities. *AIAA Journal*, 24(6):1027–1032. 88

Zhang, Z., Liuy, F., and Schusterz, D. M. (2006). An effcient euler method on non-moving cartesian grids with boundary-layer correction for wing flutter simulations. In *Aerospace Sciences Meeting*, Reno, Nevada. 50

Zienkiewicz, O. C. and Zhu, J. Z. (1992). The superconvergent patch recovery (spr) and adaptive finite element refinement. *Computer Methods in Applied Mechanics and Engineering*, 101(1):207–224. 21

Appendix A

Sensitivity of Flow in a Convergent-Divergent Nozzle

A.1 Flow Analysis

This problem involves calculating the steady-state flow quantities through the duct of a convergent divergent nozzle as shown in Figure A.1.

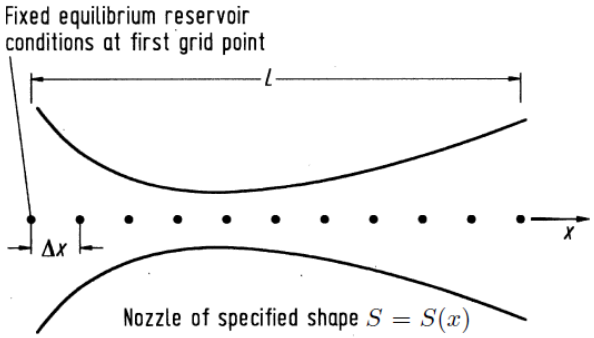

Figure A.1: A convergent divergent nozzle (Anderson, 1995)

The flow is assumed to be isentropic with subsonic inflow condition and a supersonic outflow condition resulting from a shock at the throat. The flow is also assumed to be quasi-one-dimensional. This means that the flow properties are uniform across any given cross section of the nozzle (Anderson, 1995). In other words, although the cross sectional area A

varies as a function of distance x, we treat the flow variables to be invariant in the the cross section. Unlike the Euler equations for flow over airfoil, the quasi-1-D approximation gives rise to a nonzero source term $\boldsymbol{H}$ in the Euler equations

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} - \boldsymbol{H} = 0, \tag{A.1}$$

where

$$\boldsymbol{u} = \begin{Bmatrix} \rho \\ \rho u \\ \rho e_t \end{Bmatrix}, \quad \boldsymbol{F} = \begin{Bmatrix} \rho u \\ p + \rho u^2 \\ u\left(p + \rho e_t\right) \end{Bmatrix}, \quad \boldsymbol{H} = \frac{\partial A}{\partial x} \begin{Bmatrix} 0 \\ p \\ 0 \end{Bmatrix}. \tag{A.2}$$

When a cell centered finite volume discretization is used the following semi-discrete system of equations is obtained.

$$V_j \frac{\partial}{\partial t}\left(\boldsymbol{Q}\right) + \boldsymbol{F}_{j+1/2} A_{j+1/2} - \boldsymbol{F}_{j-1/2} A_{j-1/2} - \boldsymbol{H}_j \left(\Delta x\right)_j = 0 \tag{A.3}$$

Here, ρ, p, u and e_t denote density, pressure, horizontal velocity and total energy in the convergent divergent nozzle, respectively. The pressure and density can be related to the temperature T by the equation of state $p = \rho \mathcal{R} T$, where $\mathcal{R}$ is the specific gas constant. Although these equation can be used for time-accurate solutions to the nozzle flow problem, the variable t in the present context only represents pseudo-time and is used for the purpose of time marching to reach the steady-state. The state vector of conserved variables is $\boldsymbol{u}$, $\boldsymbol{F}$ is the flux vector, and $\boldsymbol{H}$ is the source term. V_j denotes the volume of the j^{th} cell while $\left(\Delta x\right)_j$ denotes the dimension of the j^{th} cell in the X direction. The area of the left and right faces of the j^{th} cell are denoted by $A_{j-1/2}$ and $A_{j+1/2}$ respectively. The system is closed by the equation of state for ideal gas

$$p = (\gamma - 1)\rho \left[e_t - \frac{u^2}{2}\right], \quad e_t = \frac{1}{\gamma - 1}\frac{p}{\rho} + \frac{u^2}{2}. \tag{A.4}$$

In the present case, we have assumed that the nozzle flow is isentropic, with subsonic flow at the left boundary and supersonic flow at the right boundary. Hence, using the method of characteristics (Hirsch, 1990), we can conclude that (a) at the subsonic inflow boundary, we must stipulate the values of two dependent flow variables (typically pressure and density), whereas the value of one other variable (typically velocity) must be determined based on interior values, and (b) at the supersonic outflow boundary all variables (pressure, density and velocity) must be determined based on interior values. When value of a certain variable at the boundary is not fixed, that value is determined based on (linear) extrapolation from the values in the interior of the domain. Thus, the boundary conditions for this problem are that the pressure p_{in} and density ρ_{in} at the inlet are prescribed based on fixed values of stagnation temperature T_0 and stagnation pressure P_0. Value of u_{in} is extrapolated based on values in the interior or the domain. The following isentropic relations are used to set these boundary conditions.

$$T_{in} = T_0 - \frac{\gamma - 1}{2\gamma R} u_{in}^2 \tag{A.5}$$

$$\psi = \frac{T_0}{T_{in}} \tag{A.6}$$

$$p_{in} = \frac{P_0}{\psi^{\gamma/(\gamma-1)}} \tag{A.7}$$

$$\rho_{in} = \frac{P_0}{RT_0\psi^{1/(\gamma-1)}} \tag{A.8}$$

The Euler equations (A.3) can be solved with an implicit scheme as shown below. Since the problem involves calculation of steady-state quantities, the residual at cell j is given by

$$\boldsymbol{R}_j = \boldsymbol{F}_{j+1/2} A_{j+1/2} - \boldsymbol{F}_{j-1/2} A_{j-1/2} - \boldsymbol{H}_j \left(\Delta x\right)_j . \tag{A.9}$$

For the Euler implicit scheme, the residual is evaluated at the $(n+1)^{th}$ time step and so the

semi-discrete system can be written compactly as

$$\frac{\Delta \boldsymbol{u}^{(n)}}{\Delta t} + \boldsymbol{R}^{(n+1)}(\boldsymbol{u}) = 0, \tag{A.10}$$

where the superscript denotes the pseudo-time step, and $\Delta \boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n+1)} - \boldsymbol{u}^{(n)}$. The residual at the time step $(n+1)$ can be approximated to first-order as

$$\boldsymbol{R}^{(n+1)}(\boldsymbol{u}) \approx \boldsymbol{R}^{(n)}(\boldsymbol{u}) + \left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\right)^{(n)} \Delta \boldsymbol{u}^{(n)}. \tag{A.11}$$

Substituting this approximation (A.11) into Eq. (A.10), we get the implicit system

$$\left[\frac{I}{\Delta t} + \left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\right)^{(n)}\right] \Delta \boldsymbol{u}^{(n)} = -\boldsymbol{R}^{(n)}(\boldsymbol{u}), \tag{A.12}$$

where I denotes an identity matrix. Here, the quantity $\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\right)^{(n)}$ is known as the Jacobian matrix at time step n, and is same as the tangent matrix $[T(\{\boldsymbol{u}\}_h^n)]$ in Section 2.2.3. The time term $(I/\Delta t)$ is used for pseudo-time stepping, but can be avoided to get the Newton's method. This system of equations can be solved at each time step for the update of the flow variables, $\Delta \boldsymbol{u}^{(n)}$. Thus the solution at the $(n+1)^{th}$ iteration is determined from

$$\boldsymbol{u}^{(n+1)} = \boldsymbol{u}^{(n)} + \Delta \boldsymbol{u}^{(n)}.$$

A.2 Derivatives of the Flow in a quasi-1-D Convergent Divergent Nozzle

In this section we propose a nonintrusive formulation of CSA to find derivatives of the flow in a convergent divergent nozzle. The design variables are parameters that change the shape of the nozzle. Hence, if the shape of the nozzle is given by

$$A(x) = b_0 + b_1 x + b_2 x^2,$$

we are interested in finding the values of $\dot{\boldsymbol{u}} = D\boldsymbol{u}/Db$ where $b = b_0$, b_1 or b_2. An analytical solution is available for this problem. Since the analysis is concerning flow only in one dimension, the parameters b_i are value parameters and they do not physically change the shape of the domain. Hence the sensitivity problem defined here is for obtaining value sensitivity and $\dot{\boldsymbol{u}} = \boldsymbol{u}'$. As explained in Section 2.2, we will evaluate the local derivatives of the flow variables, $\boldsymbol{u}'$, by solving the local form of continuum sensitivity equations (CSEs). The CSEs are obtained by partial differentiation of the state Equations (A.3) as follows.

$$\frac{\partial}{\partial t}\left(\boldsymbol{u}'\right) + \frac{\partial}{\partial x}\left(\boldsymbol{F}'\right) - \boldsymbol{H}' = 0, \tag{A.13}$$

where

$$\boldsymbol{u}' = \frac{\partial \boldsymbol{u}}{\partial b} = \left\{ \begin{array}{c} \frac{\partial}{\partial b}(\rho) \\ \frac{\partial}{\partial b}(\rho u) \\ \frac{\partial}{\partial b}(\rho e_t) \end{array} \right\},$$

$$\boldsymbol{F}' = \frac{\partial \boldsymbol{F}}{\partial b} = \left\{ \begin{array}{c} \frac{\partial}{\partial b}(\rho u) \\ \frac{\partial}{\partial b}(p) + u\frac{\partial}{\partial b}(\rho u) + \rho u \frac{\partial}{\partial b}(u) \\ p\frac{\partial}{\partial b}(u) + u\frac{\partial}{\partial b}(p) + u\frac{\partial}{\partial b}(\rho e_t) + \rho e_t \frac{\partial}{\partial b}(u) \end{array} \right\},$$

$$\boldsymbol{H}' = \frac{\partial \boldsymbol{H}}{\partial b} = \frac{\partial A}{\partial x}\left\{ \begin{array}{c} 0 \\ \frac{\partial}{\partial b}(p) \\ 0 \end{array} \right\} + \frac{\partial}{\partial b}\left(\frac{\partial A}{\partial x}\right)\left\{ \begin{array}{c} 0 \\ p \\ 0 \end{array} \right\},$$

and where

$$\frac{\partial}{\partial b}(u) = \left[\frac{\partial}{\partial b}(\rho u) - \frac{\rho u}{\rho}\frac{\partial}{\partial b}(\rho)\right] \Big/ \rho,$$

$$\frac{\partial}{\partial b}(p) = (\gamma - 1)\left[\frac{\partial}{\partial b}(\rho e_t) - \frac{u}{2}\frac{\partial}{\partial b}(\rho u) - \frac{\rho u}{2}\frac{\partial}{\partial b}(u)\right].$$

The sensitivity boundary conditions are obtained by differentiation of the corresponding flow boundary conditions (A.5–A.8). In the present problem the boundaries of the domain do not change with the design variable. Hence, the design velocity is zero. This leads to the following sensitivity boundary conditions to be imposed at the inlet on the sensitivity variables p'_{in} and ρ'_{in}.

$$T'_{in} = -\frac{\gamma - 1}{\gamma R} u_{in} u'_{in} \tag{A.14}$$

$$\psi' = -\frac{T_0}{T_{in}^2} T'_{in} \tag{A.15}$$

$$p'_{in} = -\frac{P_0}{\psi^{1+\gamma/(\gamma-1)}} \psi' \tag{A.16}$$

$$\rho'_{in} = -\frac{P_0}{R T_0 \psi^{1+1/(\gamma-1)}} \psi' \tag{A.17}$$

The CSEs can be solved with the same implicit scheme and discretization that was used to solve the Euler equations. This leads to the following linear system to be solved for calculating the update, $\Delta \boldsymbol{u}'^{(n)}$, of the derivatives at the n^{th} time step:

$$\left[\frac{I}{\Delta t} + \left(\frac{\partial \boldsymbol{R}'}{\partial \boldsymbol{u}'}\right)\right] \Delta \boldsymbol{u}'^{(n)} = -\boldsymbol{R}'^{(n)}(\boldsymbol{u}'), \tag{A.18}$$

where $\boldsymbol{R}'^{(n)}(\boldsymbol{u}') = \frac{\partial}{\partial x}\left(\boldsymbol{F}'^{(n)}\right) - \boldsymbol{H}'^{(n)}$ is the residual for the CSEs. Note that since the CSEs are linear, the tangent matrix $\left(\frac{\partial \boldsymbol{R}'}{\partial \boldsymbol{u}'}\right)$ will be a constant matrix and pseudo-time stepping is not required for solving the CSEs, as mentioned in Section 2.2.3. Borggaard and Burns (Borggaard and Burns, 1994, 1997), Wickert (Wickert et al., 2010) and Liu and Canfield (Liu and Canfield, 2013b) showed that, if the same discretization used for the analysis is used to discretize the CSEs, then the tangent matrix for the sensitivity system is same as

the tangent matrix at the converged steady-state solution,

$$\left(\frac{\partial \boldsymbol{R}'}{\partial \boldsymbol{u}'}\right) = \left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{Q}}\right)^{(N)}, \tag{A.19}$$

where N is the last pseudo-time step of the flow analysis. So, the same coefficient matrix can be used for obtaining the local derivatives. Thus, the discretized update equation has the form

$$\left[\frac{I}{\Delta t} + \left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{Q}}\right)^{(N)}\right] \Delta \boldsymbol{u}'^{(n)} = -\boldsymbol{R}'^{(n)}\left(\boldsymbol{u}'\right). \tag{A.20}$$

A few comments worth mentioning about Eq. (A.20) and the associated CSE BCs (A.14–A.17):

- Pseudo-time stepping is optional, but if done, at each time step the derivatives can be obtained based on the converged tangent matrix used for the calculating the state variables. This avoids the expensive step of calculating Jacobian matrix for the CSEs, provided it is available from the analysis.

- The only expense in solving sensitivities at each time step is involved in calculating the residual $\boldsymbol{R}'^{(n)}\left(\boldsymbol{u}'\right)$ based on the expressions for the flux vector $\boldsymbol{F}'$ and source term $\boldsymbol{H}'$.

- The design variables considered for this problem are value parameters, because they lead to zero design velocity. Hence the CSE boundary conditions are homogeneous (similar to those of the Euler equations). There are no convective terms involving spatial gradients in these boundary conditions. This avoids calculation of the spatial gradients at the boundaries, which are otherwise required for assembling the sensitivity boundary conditions as seen in Equation 2.5. However, the quasi-1-D assumption leads to a nonzero source term $\boldsymbol{f}\left(\boldsymbol{x}, t; b\right)$ for the flow equations (2.1) and the term $\boldsymbol{f'}\left(\boldsymbol{x}, t; b\right)$ sensitivity equations (2.6). This is a typical feature of CSEs for value parameters.

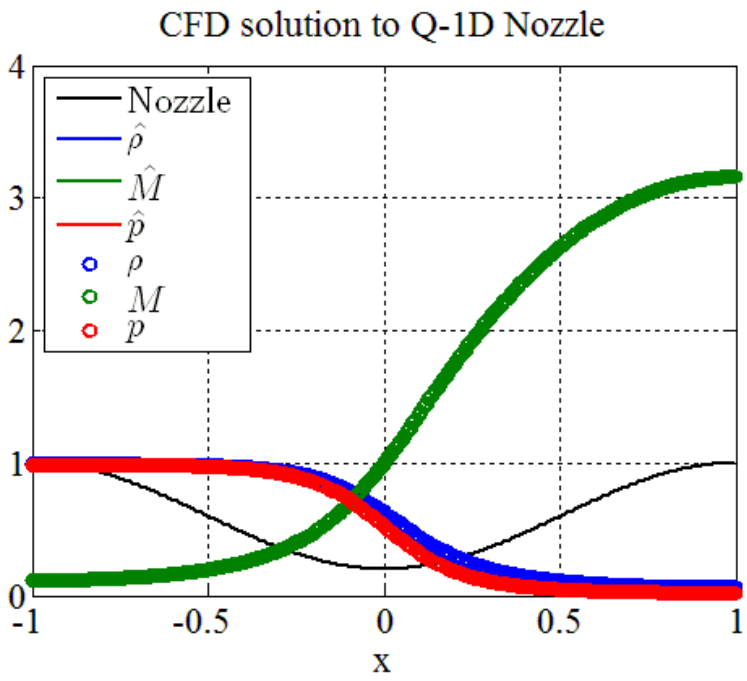

Figure A.2: Density, Mach number and pressure in a convergent divergent nozzle. Solid lines indicate analytical solution, whereas the circles indicate numerical (Euler) solution.

A.3 Results

The Euler solution results for isentropic flow in the convergent divergent nozzle, compared to the analytical solutions are shown in Figure A.2. The solutions were converged such that the L_2 norm of the conservation equations were less than 10^{-10}.

Although an analytic solution is available for the derivatives of the nozzle flow, to highlight the disadvantages of the finite difference method, a step-size study was performed as illustrated in Figure A.3. It is not intuitive to find out the correct step-size of 1×10^{-5}. As expected, the central finite difference method yields better than the forward finite difference for larger step-sizes, but is doubly expensive.

The derivatives of the nozzle flow obtained using CSA are plotted in Figure A.4 along with the analytical derivatives and the finite difference derivatives. The CSA results are closer to the analytical results than the best finite difference results.

A grid convergence study was done to find out the rate of convergence of the flow deriv-

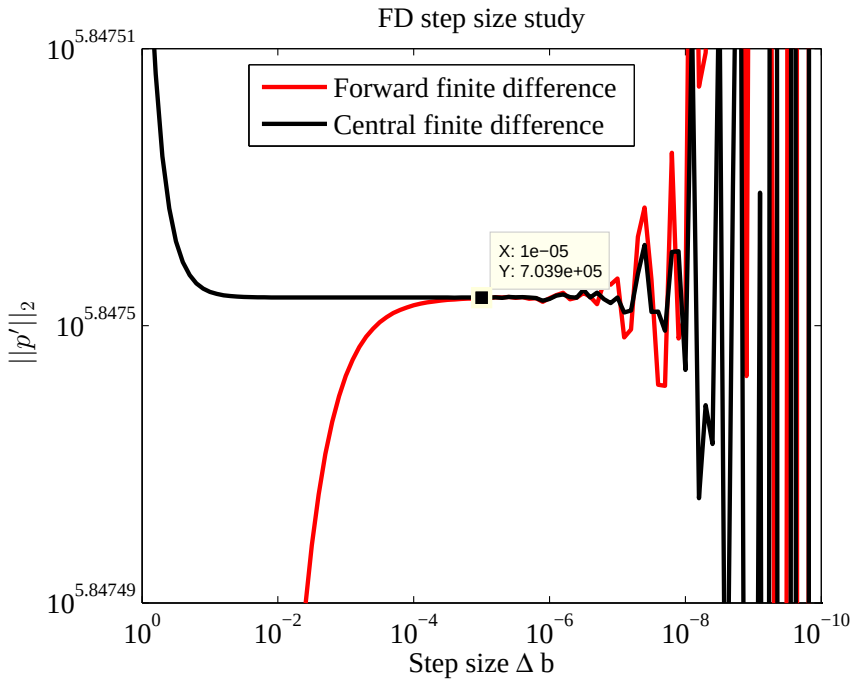

Figure A.3: Step size convergence of Euler solution for the convergent-divergent nozzle.

atives. The flow solution was based on a first-order accurate Euler code, so the derivatives are expected to converge at a rate not more than one. This is seen in the convergence results plotted in Figure A.5. These results were obtained with the Roe's flux difference scheme (Hirsch, 1990). To make sure that a change in the flux scheme does not alter the results, the same problem was solved using the Van Leer flux vector splitting scheme (Knight, 2006). The derivative solutions obtained with Van Leer scheme match those obtained with Roe's scheme. The rate of convergence of one was obtained with the Van Leer scheme as well as shown in Figure A.6. This verifies that the CSA gives consistent result with both the flux schemes. Implementation wise, there was no difference in CSA procedure other than changing the flux scheme used for flow and sensitivity analysis.

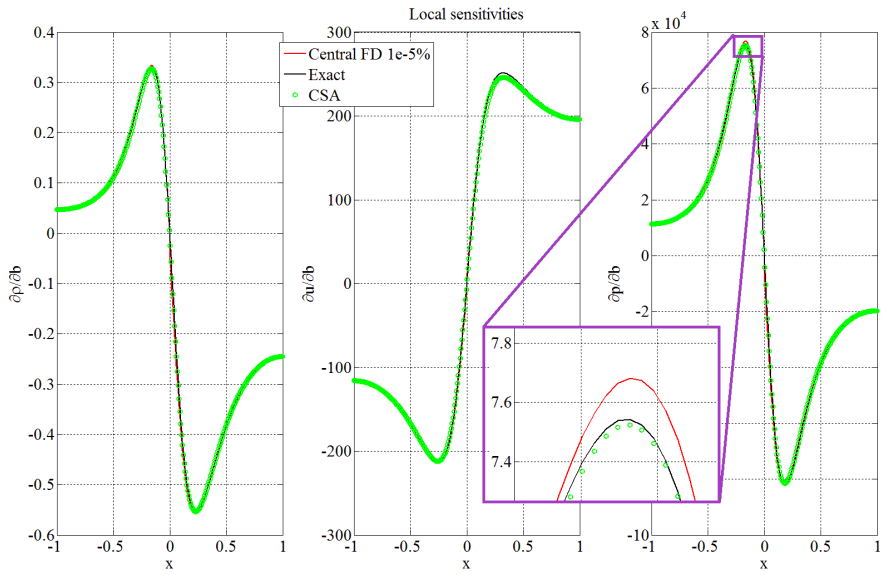

Figure A.4: Derivatives of the nozzle flow.

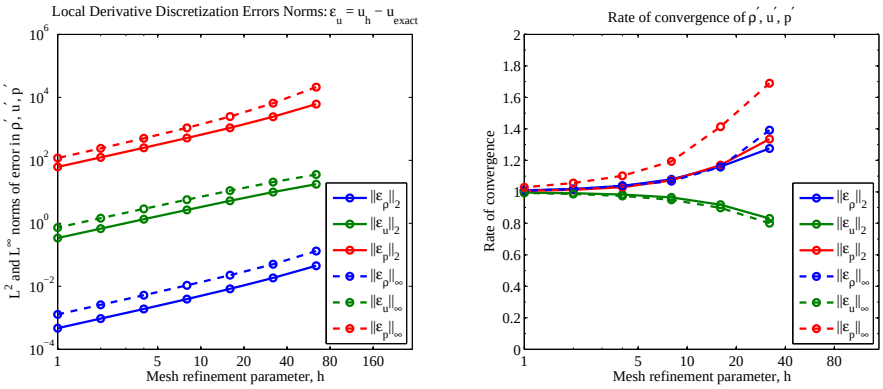

Figure A.5: Rate of convergence of derivatives of the nozzle flow, with Roe's flux difference scheme.

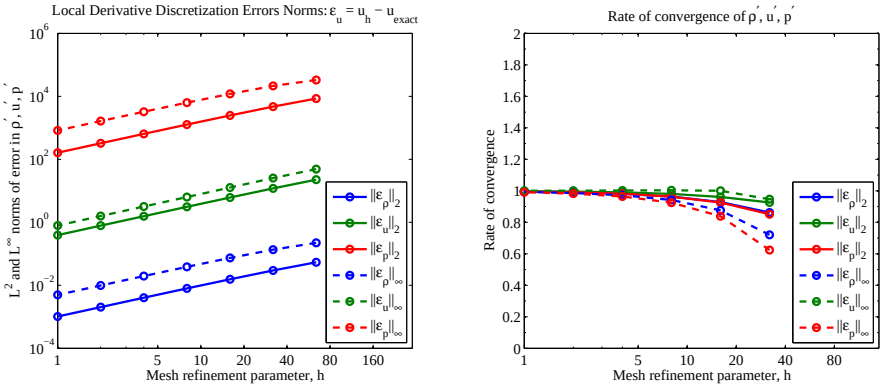

Figure A.6: Rate of convergence of derivatives of the nozzle flow, with Van Leer's flux vector splitting scheme.

Appendix B

Reduced Order Model for Unsteady Aerodynamics of Flapping Wing Micro Air Vehicle in Hover

This work was presented as paper AIAA 2013-1645 (Kulkarni et al., 2013) at the 2013 AIAA SDM conference.

B.1 Introduction and motivation

Micro Air Vehicles (MAVs) have gained large popularity (Petricca et al., 2011; Keennon et al., 2012; Pines and Bohorquez, 2006) for surveillance and reconnaissance applications owing to their small size and ability to fly in indoor environments. MAVs can be classified into three main categories: fixed wing, rotary wing (including quadrotors) and flapping wing. Fixed wing MAVs depend on forward speed for lift generation hence cannot be used for hovering. Rotary wing MAVs are capable of hovering and vertical take-off and landing. However, such maneuvers are energy expensive and decrease the range and endurance of rotary wing MAVs (Petricca et al., 2011). Flapping wing MAVs can be more efficient than fixed and rotary wing MAVs in terms of specific power requirement (Woods et al., 2001). As a result, a lot

of research is being directed towards design and development of flapping wing MAVs.

Present day flapping wing MAVs, such as the Nano-Hummingbird (Keennon et al., 2012) developed by Aerovironment Inc., are designed mainly based on quasi-steady aerodynamics and blade-element theory based approaches. However, it has been found that such theories are incapable of doing justice to the actual physics of the problem, which is much complex. Flapping flight is associated with laminar, incompressible, unsteady flow in low Reynolds number regime (Galinski and Zbikowski, 2007). Lift and thrust forces generated during flapping are a result of several unsteady aerodynamic phenomenon such as delayed stall, rotational circulation, wake capture (Dickinson et al., 1999), leading edge vortex, added mass effect (Ansari et al., 2006a), etc.

Some researchers have suggested the use of low fidelity, quasi-steady aerodynamic models (Sane and Dickinson, 2002) for designing flapping MAVs (Doman et al., 2009; Berman and Wang, 2007). These models rely on assumptions of low angle of attack and low reduced frequency flight. Such models give reasonably good estimate of lift and thrust forces for high and moderate forward speeds. At low forward speeds and high reduced frequencies, wake shed by the flapping MAV is in close vicinity of the lifting surface. In case of hovering flapping wing, reduced frequency may even go to infinity. The wake pattern for such cases is very complicated and unsteady effects become more prominent (Stanford and Beran, 2010; Fritz and Long, 2004). Researchers have proved that steady-state and quasi-steady-state aerodynamic theories are insufficient to estimate the high lifting forces produced by birds and insects (Pines and Bohorquez, 2006). Hence, quasi-steady aerodynamic models are unreliable for designing flapping MAVs capable of hovering or performing low velocity forward flight.

Momentum Disc Theory (MDT) is often used to model aerodynamics for micro helicopters (Schafroth, 2010). MDT was used by Weiss-Fogh to derive induced downwash for hovering insects (Weis-Fogh, 1972). His theory was based on the assumption that the high frequency flapping wings of the insects behave like an actuator disc. Lift which is produced

by such a disc can be used to calculate inflow in the plane of the disc. This induced downwash incorporated the effect of the shed wake. This approach for modeling inflow can be grouped in the category of low fidelity models. Later Ellington (Ellington, 1984) proposed an expression based on partial-actuator disc which was more representative of actual hovering flight. However, such approaches are not directly suitable to account for changes in flapping kinematics or wing geometry and thus are not suitable for the design of flapping wing MAVs.

Several high fidelity studies such as CFD analysis (Sun and Xiong, 2005; Sun et al., 2007) have been done to accurately capture the physics and to obtain lift and thrust forces produced by birds and insects. Such studies are often very computationally expensive. Hence, it is not feasible to carry out optimization and sensitivity analysis, or control system design studies while relying on such high fidelity methods (Berman and Wang, 2007).

There are some studies which have used medium-fidelity aerodynamic tools such as Unsteady Vortex Lattice Method (UVLM) for finding optimum design of flapping wing MAVs (Ghommem, 2011). A model proposed by Ansari and co-workers (Ansari et al., 2006b,c) is another medium-fidelity tool which accounts for the unsteady aerodynamic phenomenon. These models are essentially two dimensional models which are extended to three dimensions by using blade element theory. This means that there is no interaction between two neighboring chord-wise strips of elements on the wing.

In reality, wake shed by the wing consists of tubes of vorticity. In case of a flapping wing, these tubes would criss-cross each other and the vortex tubes would get severely twisted and distorted (Stanford and Beran, 2010). It would be very complicated to model such a wake using UVLM beyond a certain number of oscillations. To simplify this, wake is often assumed to be fixed instead of being free to convect. Authors have not found any studies which use medium fidelity three dimensional aerodynamics methods with free wake to model hovering flapping flight. The current work is a step in that direction.

Viscous effects become prominent for flow at low Reynolds number. Insect-like flapping flight (Dudley, 2000; Shyy et al., 2008) is categorized into flow at Reynolds numbers 10

through 10^5. Flow structures for insect flight would certainly depend on viscosity of the fluid and size of the insect. Still there is remarkable similarity in the aerodynamic forces on airfoils predicted by CFD simulations and those predicted by inviscid methods (Sane and Dickinson, 2002, 2003). Although inviscid aerodynamic models cannot capture flow separation, transition and reattachment, viscosity is taken in consideration indirectly by the Kutta-Joukowski condition (Ansari et al., 2006a). It is thus reasonable to use an inviscid aerodynamic model for doing optimization studies and design of flapping wing MAVs (Stanford and Beran, 2010).

In the present paper, we propose a reduced order aerodynamic model for a flapping airfoil in hover. This model is a combination of low fidelity MDT model and a medium fidelity UVLM. The flapping airfoil sheds vortices at each time step. These vortices induce velocity in the vicinity of the airfoil which is also called inflow. This inflow consists of two parts: (a) A 'global' velocity, and (b) A 'local' velocity. The global velocity is a result of the counter rotating vortices shed by the airfoil during each half stroke and has the effect of pushing the flow downwards. The local velocity is the inflow due to vortices which are close to the airfoil. Velocity induced by the vortices decreases with square of the distance from the airfoil, and therefore, vortices far away do not contribute much to the local velocity. In the current work, global inflow is approximated using the MDT while the local inflow is approximated using UVLM and considering only the 'near' wake. Results are compared with UVLM computation for the case when influence of the entire wake is retained. It is observed that significant accuracy is obtained even when the recent one oscillation is retained. (By 'retaining recent one oscillation', it is meant that 'retaining vortices shed in the recent one oscillation'.) Retaining only a fraction of the wake simplifies computation significantly. Such approach can thus be extended to the 3D case.

UVLM computation for calculating lift and drag is explained in the next section. Similar computation for a quasi-steady case is explained in Section III. The proposed reduced order scheme is explained in Section IV. Error metrics used to judge the effect of approximations are

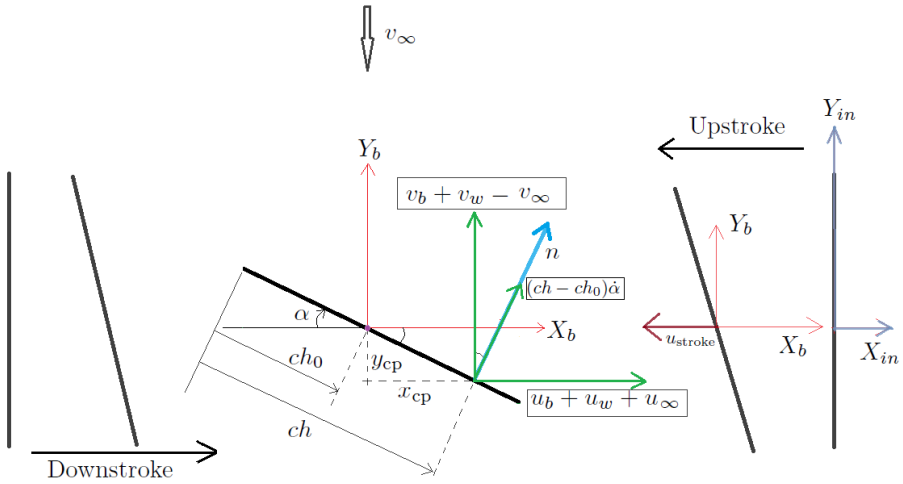

Figure B.1: Airfoil flapping motion

defined in Section V. Results are discussed in Section VI. It is found that there is uncertainty in lift depending on the starting location. This is explained in Section VII.

B.2 Unsteady Vortex Lattice Method

The hovering flapping motion of an airfoil is shown in Fig. B.1. $X_{in} - Y_{in}$ define the inertial reference frame while $X_b - Y_b$ define the body-fixed reference frame. The body-fixed reference frame translates with the airfoil hinge and remains parallel to the inertial reference frame. In Fig. B.1, the downstroke motion of the airfoil is shown below the upstroke motion. However, in all the simulations, there is no deviation (up-down movement of the airfoil) and hence, during upstroke and downstroke, mid chord of the airfoil moves on the stroke plane defined by $y_{in} = 0$. At any time t, position of the mid-chord in the inertial frame s_{in}, and airfoil angle of attack α are given by:

$$\begin{aligned} s_{in} &= s_0 \cos(\omega t + s_\phi) \\ \alpha &= \frac{\pi}{2} - \alpha_0 \sin(\omega t + \alpha_\phi) \end{aligned} \tag{B.1}$$

In Eq. B.1, α_0 is the amplitude of rotation measured with respect to the vertical axis Y_b. It is assumed that the airfoil is flapping in still air, hence velocities experienced by the airfoil due to this flapping motion are:

$$\begin{aligned} u_\infty &= -u_{stroke} = -\frac{\partial s_{in}}{\partial t} = \underbrace{(s_0\omega)}_{u_0} \sin(\omega t + s_\phi) = u_0 \sin(\omega t + s_\phi) \\ v_\infty &= 0... \text{ because no deviation} \\ \dot{\alpha} &= \frac{\partial \alpha}{\partial t} = -\alpha_0 \cos(\omega t + \alpha_\phi) \end{aligned} \tag{B.2}$$

The airfoil has semi-chord length b and the chord is divided into n_b segments. It is assumed that a bound vortex is located at quarter segment location and a control point is located at three-quarter segment location of each segment. The hinge about which the airfoil rotates while flapping, is located at a distance h_0 from the leading edge as shown in figure B.1.

Position of a control point (x_{cp}, y_{cp}) at a distance h_{cp} from leading edge in the body reference frame $X_b - Y_b$ is given by:

$$\begin{aligned} x_{cp} &= (h_{cp} - h_0)\cos\alpha \\ y_{cp} &= -(h_{cp} - h_0)\sin\alpha \end{aligned} \tag{B.3}$$

Similarly, the position of a bound vortex (x_{bv}, y_{bv}) at distance h_{bv} from leading edge in the body reference frame $X_b - Y_b$ is given by:

$$\begin{aligned} x_{bv} &= (h_{bv} - h_0)\cos\alpha \\ y_{bv} &= -(h_{bv} - h_0)\sin\alpha \end{aligned} \tag{B.4}$$

The recently shed vortex is placed at a distance of quarter segment location from the trailing edge along the extended airfoil.

As the airfoil flaps, it sheds one vortex at each time step. At each time step, the unknowns are the strengths of the n_b bound vortices (Γ_{b_i}) and the recently shed wake vortex (Γ_{w_1}). To get these values, we have n_b number of equations which satisfy the no-penetration boundary condition at each of the n_b control points, and the equation of conservation of circulation (Kelvin's circulation theorem). Following algorithm is used for the computation:

1. Define kinematics (Eq. B.1)
2. Get positions of bound vortices, control points and recently shed vortex at each time step (Eqs. B.3 and B.4)
3. Calculate Aerodynamic Influence Coefficients (AICs): Effect of bound vortices at each control point and at each time step
4. Start iteration for each time step:
   (a) Calculate AICs due to wake vortices: Effect of wake vortices at each control point
   (b) Solve $(n_b + 1)$ equations simultaneously to get values of Γ_{b_i} and Γ_{w_1}
   (c) Plot airfoil, bound vortices, control points, shed vortices
   (d) Calculate and plot inflow at the stroke plane
   (e) Calculate lift and drag
   (f) Free wake: Calculate velocity induced by bound and shed vortices at each shed vortex and shift the shed vortex to its new position (in body-fixed frame $X_b - Y_b$)
   (g) Iterate steps (a)-(f) for $t = 0 : dt : (n_t - 1)dt$

B.2.1 No-penetration boundary condition

No-penetration boundary condition is imposed at each control point on the airfoil. Velocity at each control point is a combination of (i) Velocity induced by bound vortices (u_b, v_b), (ii) Velocity induced by shed vortices (u_w, v_w), and (iii) Structural velocities given by Eq.

B.2. Note that the structural velocity due to $\dot{\alpha}$ acts perpendicular to the airfoil. Hence, the no-penetration boundary condition at each control point is given by:

$$(u_b + u_w + u_\infty)\sin\alpha + (v_b + v_w - v_\infty)\cos\alpha + (h_{cp} - h_0)\dot{\alpha} = 0$$

This equation can be simplified by splitting the velocity due to $\dot{\alpha}$ into components ($u_{\dot{\alpha}}$, $v_{\dot{\alpha}}$) as:

$$(u_b + u_w)\sin\alpha + (v_b + v_w)\cos\alpha = -(u_\infty - u_{\dot{\alpha}})\sin\alpha - (-v_\infty + v_{\dot{\alpha}})\cos\alpha \tag{B.5}$$

Vortex strengths are considered to be positive in clockwise direction (i.e. positive Γ is in $-Z_{in}$ direction). Applying Biot-Savart law, velocities induced by the n_b bound vortices and n_w wake vortices at control point j, can be found, and Eq. B.5 can be written as:

- Left-Hand-Side:

$$(u_b + u_w)\sin\alpha + (v_b + v_w)\cos\alpha\Big|_j = \sum_{i=1}^{n_b} A_{B_{j,i}}\Gamma_{b_i} + A_{W_{j,1}}\Gamma_{w_1} + \sum_{i=2}^{n_w} A_{W_{j,i}}\Gamma_{w_i}$$

- Right-Hand-Side:

$$-(u_\infty - u_{\dot{\alpha}})\sin\alpha - (-v_\infty + v_{\dot{\alpha}})\cos\alpha\Big|_j = \underbrace{\{-u_\infty\sin\alpha + v_\infty\cos\alpha\}}_{\text{same for all } j} + \underbrace{\left\{y_{cp_j}\dot{\alpha}\sin\alpha - x_{cp_j}\dot{\alpha}\cos\alpha\right\}}_{\text{changes with } j}$$

Thus, the no-penetration boundary is:

$$\sum_{i=1}^{n_b} A_{B_{j,i}}\Gamma_{b_i} + A_{W_{j,1}}\Gamma_{w_1} + \sum_{i=2}^{n_w} A_{W_{j,i}}\Gamma_{w_i} = \{-u_\infty\sin\alpha + v_\infty\cos\alpha\} + \left\{y_{cp_j}\dot{\alpha}\sin\alpha - x_{cp_j}\dot{\alpha}\cos\alpha\right\} \tag{B.6}$$

Here, A_B and A_W are AICs.

B.2.2 Calculation of Γ_{b_i} and Γ_{w_1}

No-penetration boundary condition at control point j (Eq. B.6) can be reorganized as:

$$\sum_{i=1}^{n_b} A_{B_{j,\,i}}\Gamma_{b_i} + A_{W_{j,\,1}}\Gamma_{w1} = \underbrace{\{-u_\infty \sin\alpha + v_\infty \cos\alpha\} + \left\{y_{cp_j}\dot{\alpha}\sin\alpha - x_{cp_j}\dot{\alpha}\cos\alpha\right\}}_{\text{Structural velocity, } v_{str}} - \underbrace{\sum_{i=2}^{n_w} A_{W_{j,\,i}}\Gamma_{w_i}}_{\text{Effect of shed wake, } v_{wake}} \tag{B.7}$$

In Eq. B.7, AIC $A_{B_{j,\,i}}$ relates the effect of bound vortex Γ_{b_i} on control point j and AIC $A_{W_{j,\,1}}$ relates the effect of recently shed wake vortex Γ_{w_1} on the control point j. These coefficients can be pre-calculated since they depend only on the positions of the bound vortices, control points and the recently shed wake vortex. This calculation can be done once the kinematics are defined. The right hand side of Eq. B.7 consists of structural velocity, v_{str} and, velocity induced by the shed wake vortices (other than the recently shed vortex), v_{wake}. v_{str} is also known and can be pre-calculated once the kinematics are defined. v_{wake} depends on the positions of the wake vortices. Since the wake is free, these wake vortices move depending on the velocity induced at their position by other wake vortices. Thus, calculation of v_{wake} is a computationally expensive step. Aim of the current work is to simplify computation of this term by making reasonable approximations to the wake shed by the hovering airfoil.

The only unknowns in Eq. B.7 are Γ_{b_i}, $i \subseteq [1, n_b]$ and Γ_{w_1}. Equation B.7 can be written at each of the n_b control points to get a set of n_b equations. To solve for the (n_b+1) unknowns, we need an additional equation, which is the equation for conservation of circulation:

$$\sum_{i=1}^{n_b}\Gamma_{b_i} + \sum_{i=1}^{n_w}\Gamma_{w_i} = 0$$

$$\implies \sum_{i=1}^{n_b}\Gamma_{b_i} + \Gamma_{w_1} = -\sum_{i=2}^{n_w}\Gamma_{w_i} \tag{B.8}$$

Combining Eq. B.7 $\forall j \subseteq [1, n_b]$ and Eq. B.8, we get:

$$\underbrace{\begin{bmatrix} A_{B_{j,\,i}} & A_{W_{j,\,1}} \\ 1 & 1 \end{bmatrix}_{(n_b+1)\times(n_b+1)}}_{\text{AIC are known}} \begin{Bmatrix} \Gamma_{b_i} \\ \Gamma_{w_1} \end{Bmatrix} = \underbrace{v_{str} + v_{wake}}_{\text{known at each time step}} \tag{B.9}$$

$$v_{wake} = -\begin{bmatrix} A_{W_{j,\,i}} \\ 1 \end{bmatrix}_{(n_b+1)\times(n_w-1)} \{\Gamma_{w_i}\}_{(n_w-1)\times 1} \tag{B.10}$$

$$v_{str} = (-u_\infty \sin\alpha + v_\infty \cos\alpha) \begin{Bmatrix} \mathbf{1} \\ 0 \end{Bmatrix}_{(n_b+1)\times 1} + \begin{Bmatrix} y_{cp_j} \\ 0 \end{Bmatrix} \dot{\alpha}\sin\alpha - \begin{Bmatrix} x_{cp_j} \\ 0 \end{Bmatrix} \dot{\alpha}\cos\alpha \tag{B.11}$$

Equation B.9 can be solved at each time step to get values Γ_{b_i} and Γ_{w_1}. Expression for v_{wake} (Eq. B.10) and v_{str} (Eq. B.11) are the same as in Eq. B.7 except for an additional row to account for the equation of conservation of circulation (Eq. B.8).

Computation of v_{wake} involves computation of a $(n_b + 1) \times (n_w - 1)$ matrix of AICs. Thus, computation of the influence of wake becomes increasingly complicated as the number of wake vortices n_w increases. It would be even more complicated to get the influence of wake in the case of 3D flapping wing (Stanford and Beran, 2010; Fritz and Long, 2004). In the current work we focus on getting the value of v_{wake} using a combination of MDT and by retaining only a part of the wake, instead of all the n_w vortices.

B.2.3 Calculation of Lift and Drag

Circulation (positive in $-Z_{in}$ direction) produced on the airfoil is given by:

$$\hat{\Gamma}_b = \sum_{j=1}^{n_b} \Gamma_{b_j} \tag{B.12}$$

Circulatory force generated by the airfoil can be resolved into components along X_{in} and

Y_{in} as:

- Component along Y_{in} or circulatory lift:

$$L_c = \sum_{j=1}^{n_b} \rho \left\{ (u_{w_j} + u_\infty - u_{\dot{\alpha}_j}) \right\} \Gamma_{b_j} \tag{B.13}$$

- Component along X_{in} or circulatory drag:

$$D_c = -\sum_{j=1}^{n_b} \rho \left\{ (v_{w_j} - v_\infty + v_{\dot{\alpha}_j}) \right\} \Gamma_{b_j} \tag{B.14}$$

Here, u_{w_j}, $u_{\dot{\alpha}_j}$ etc. are calculated at the vortex point j and not at the control point. This is because Γ_b value is available at the vortex point and not at the control point.

The pressure difference ΔP_{nc} between the lower and upper surfaces of the airfoil due to non-circulatory effects, at location x^* (measured from leading edge) is given by:

$$\begin{aligned}
\Delta P_{nc}(x^*) &= \rho \int_0^{x^*} \frac{\partial \gamma_b}{\partial t}\, d\zeta \\
&= \rho \frac{\partial}{\partial t} \int_0^{x^*} \gamma_b\, d\zeta \\
\hat{\Gamma}_b(x^*) &= \int_0^{x^*} \gamma_b\, d\zeta \\
\therefore \Delta P_{nc}(x^*) &= \rho \dot{\hat{\Gamma}}_b(x^*)
\end{aligned}$$

Here $\hat{\Gamma}_b$ is the circulation which is the summation of Γ_{b_j} values obtained in the code. Thus, pressure difference at segment j is given by:

$$\Delta P_{nc}\Big|_j = \rho \sum_{i=1}^{j} \dot{\Gamma}_{b_i}$$

Using this pressure difference, we get the non-circulatory force produced by the airfoil

as:

$$\begin{aligned} F_{nc} &= \int_0^c \Delta P_{nc}(\zeta)\, d\zeta \\ &= \sum_{j=1}^{n_b} \Delta P_{nc_j} \Delta l_j \\ &= \Delta l \sum_{j=1}^{n_b} \rho \sum_{i=1}^{j} \dot{\Gamma}_{b_i} \end{aligned} \tag{B.15}$$

Since F_{nc} always acts perpendicular to the airfoil (from the lower surface to the upper surface), total lift and drag forces, and sectional lift and drag coefficients are obtained as:

$$L = L_c + F_{nc}\cos\alpha, \qquad C_L = \frac{L}{\frac{1}{2}\rho u_0^2 (2b)} \tag{B.16}$$

$$D = D_c + F_{nc}\sin\alpha, \qquad C_D = \frac{D}{\frac{1}{2}\rho u_0^2 (2b)} \tag{B.17}$$

B.3 Quasi-steady calculation

In case of quasi-steady calculation, no wake is shed and the no-penetration boundary condition (Eq. B.9) is modified to:

$$\underbrace{\left[A_{B_{j,\,i}}\right]_{n_b \times n_b}}_{\text{AIC are known}} \{\Gamma_{b_i}\}_{QS} = \underbrace{v_{str}}_{\text{known at each time step}} \tag{B.18}$$

$$v_{str} = (-u_\infty \sin\alpha + v_\infty \cos\alpha)\,\{\mathbf{1}\}_{n_b \times 1} + \left\{y_{cp_j}\right\}\dot{\alpha}\sin\alpha - \left\{x_{cp_j}\right\}\dot{\alpha}\cos\alpha$$

Since there are only n_b unknowns ($\Gamma_{b_{i_{QS}}}$), and there are no vortices shed in the wake, the equation of conservation of circulation is not used in the calculation.

Quasi-steady circulation produced on the airfoil is given by:

$$\hat{\Gamma}_{b_{QS}} = \sum_{j=1}^{n_b} \Gamma_{b_{j_{QS}}}$$

Circulatory lift (along Y_{in}) and circulatory drag (along X_{in}) are given by:

$$\begin{aligned} L_{c_{QS}} &= \sum_{j=1}^{n_b} \rho \left\{ (u_\infty - u_{\dot{\alpha}_j}) \right\} \Gamma_{b_{j_{QS}}} \\ D_{c_{QS}} &= -\sum_{j=1}^{n_b} \rho \left\{ (-v_\infty + v_{\dot{\alpha}_j}) \right\} \Gamma_{b_{j_{QS}}} \end{aligned}$$

It should be noted that, as opposed to Eqs. B.13 and B.14, there are no terms corresponding to velocity induced by wake vortices in these equations.

Non-circulatory quasi-steady force is given by:

$$F_{nc_{QS}} = \Delta l \sum_{j=1}^{n_b} \rho \sum_{i=1}^{j} \dot{\Gamma}_{b_{i_{QS}}}$$

Thus, total quasi-steady lift and quasi-steady drag are given by:

$$\begin{aligned} L_{QS} &= L_{c_{QS}} + L_{nc_{QS}} \cos\alpha \\ D_{QS} &= D_{c_{QS}} + L_{nc_{QS}} \sin\alpha \end{aligned}$$

This total lift L_{QS} is used to obtain value of error metrics as shown in the Section V.

B.4 Proposed reduced order method

B.4.1 Momentum Disc Theory (MDT) calculation

According to MDT (Johnson, 1994), inflow generated by an actuator disc which produces a lift L is given by:

$$v_{wake,\,\text{Momentum Disc Theory}} = \sqrt{\frac{L_{previous\,cycle}}{2\rho A}} \tag{B.19}$$

Here, A is the area of the actuator disc and ρ is the density of air. This inflow is a result of the lift generated by the actuator disc. Weis-Fogh (1972) and Ellington (1984) used similar expressions for approximating the effect of wake for a flapping airfoil.

As a first approximation, for a flapping airfoil, we can assume that A is the area swept by the airfoil at the stroke plane, $A = 2s_0$. Span of the airfoil is assumed to be unity. We can assume that L is the lift generated by the airfoil in the recent one oscillation.

Inflow in the vertical direction, v_{wake} calculated on the stroke plane during each time step of the last flapping oscillation, is shown in Fig. B.2. Flapping kinematics are defined by: frequency of 1 Hz ($\omega = 2\pi$ rad.), stroke amplitude $s_0 = 2$ (i.e. flapping in the region $-2 \leq x_{in} \leq 2$ and $y_{in} = 0$) and, rotation amplitude $\alpha_0 = 30°$. The thick magenta line shows inflow for the case of quasi-steady calculation. As expected, this value is zero, because no wake is shed in quasi-steady calculation. Inflow obtained from MDT (Eq. B.19) is shown with a thick black line. Thin blue lines represent inflow obtained using UVLM calculation which has influence of entire wake, as explained in Section II. Red circle represents inflow at the location of the mid-chord at each time step for which the blue thin lines are plotted. Thick red line represents mean (time-average) of the values of inflow at the mid-chord (red circles). The black line is quite close to the red line, thus, it can be seen that MDT gives a close estimate of the full-wake UVLM inflow. However, MDT inflow is the same at all locations on the stroke plane. We can thus assume that the value of inflow obtained from MDT (Eq. B.19) represents a 'global' downward velocity.

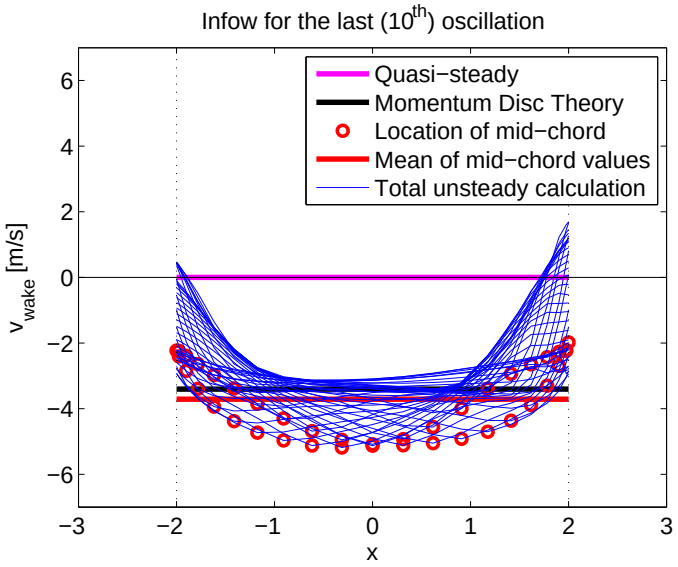

Figure B.2: Comparison of inflow obtained from MDT with inflow from total unsteady calculation

Quasi-steady calculation assumes that there is no wake shed. Hence, it can be said to be one end of the spectrum of approximating v_{wake}, which gives the most inaccurate result. Whereas, the case when the entire wake is retained (which is referred to as 'total unsteady calculation') is the other end of this spectrum, which gives the most accurate result in the current realm of UVLM calculation. We can improve the quasi-steady calculation by incorporating the inflow calculated using MDT. With this assumption, the no-penetration boundary condition will be:

$$\underbrace{\left[A_{B_{j,\,i}}\right]_{n_b \times n_b}}_{AIC \text{ are known}} \{\Gamma_{b_i}\} \quad = \quad \underbrace{v_{str} + v_{wake,\,MDT}}_{\text{known at each time step}} \tag{B.20}$$

$$v_{wake,\,MDT} = \sqrt{\frac{L_{mean,\,previous\,cycle}}{2\rho A}} \cos\alpha \, \{\mathbf{1}\}_{n_b \times 1} \tag{B.21}$$

$$v_{str} \quad = \quad (-u_\infty \sin\alpha + v_\infty \cos\alpha) \, \{\mathbf{1}\}_{n_b \times 1} + \left\{y_{cp_j}\right\} \dot{\alpha} \sin\alpha - \left\{x_{cp_j}\right\} \dot{\alpha} \cos\alpha$$

As for quasi-steady calculation, no wake is shed in this case, however, a constant inflow $v_{wake,\,MDT}$ is added to approximate the effect of shed wake as shown in Eq. B.20. Computation required to evaluate Eq. B.21 is minimal.

B.4.2 Calculation using Reduced Wake and Momentum Disc Theory (RW-MDT)

Velocity induced by shed vortices decreases as $(1/r^2)$, where r is the distance of vortex from the point of interest. Thus, the inflow generated by vortices which are close to the airfoil is more than that generated by far away vortices. During hover, most of the vortices shed by the airfoil are close to it, however, these vortices slowly move downwards, away from the airfoil, if stroke averaged lift is generated. One way of approximating the wake is to consider only the recent vortices shed by the airfoil instead of all the shed vortices. Along these lines, the no-penetration boundary condition for total unsteady calculation (Eqs. B.9, B.10) can be modified as follows:

$$\underbrace{\begin{bmatrix} A_{B_{j,\,i}} & A_{W_{j,\,1}} \\ 1 & 1 \end{bmatrix}_{(n_b+1)\times(n_b+1)}}_{\text{AIC are known}} \begin{Bmatrix} \Gamma_{b_i} \\ \Gamma_{w_1} \end{Bmatrix} = \underbrace{v_{str} + v_{wake,\,APPROX.} + \Gamma_{corr}}_{\text{known at each time step}} \tag{B.22}$$

$$v_{wake,\,APPROX.} = v_{Near\,wake} + (v_{wake,\,MDT} - v_{correction}) \tag{B.23}$$

$$v_{Near\,wake} = -\begin{bmatrix} A_{W_{j,\,i}} \\ 1 \end{bmatrix}_{(n_b+1)\times(m_w-1)} \{\Gamma_{w_i}\}_{(m_w-1)\times 1} \tag{B.24}$$

$$v_{wake,\,MDT} = \sqrt{\frac{L_{mean,\,previous\,cycle}}{2\rho A}}\cos\alpha \left\{ \begin{array}{c} \mathbf{1} \\ 0 \end{array} \right\}_{(n_b+1)\times 1} \tag{B.25}$$

$$v_{correction} = v_{avg\,inflow}\cos\alpha \left\{ \begin{array}{c} \mathbf{1} \\ 0 \end{array} \right\} \tag{B.26}$$

$v_{avg\,inflow} \rightarrow$ Space-average (over the stroke-plane $-2 \leq x \leq +2$) and time-average inflow (over previous cycle) calculated at the position of airfoil

$$v_{str} = (-u_\infty \sin\alpha + v_\infty \cos\alpha) \left\{ \begin{array}{c} \mathbf{1} \\ 0 \end{array} \right\} + \left\{ \begin{array}{c} y_{cp_j} \\ 0 \end{array} \right\} \dot{\alpha}\sin\alpha - \left\{ \begin{array}{c} x_{cp_j} \\ 0 \end{array} \right\} \dot{\alpha}\cos\alpha$$

$$\Gamma_{corr} = -\left\{ \begin{array}{c} \mathbf{0} \\ 1 \end{array} \right\} (\sum \Gamma_{w,\,neglected}) \tag{B.27}$$

The main difference between Eq. B.9 and Eq. B.22 is brought by changing the term v_{wake}. v_{wake} is split into two parts as shown in Eq. B.23:

- $v_{Near,\,wake}$ represents the effect of retaining only the recent m_w $(2 \leq m_w < n_w)$ vortices instead of all the n_w vortices shed in the wake. Thus, it can be seen that Eq. B.24 which is used to evaluate this term consists of a matrix of dimension $(n_b+1)\times(m_w-1)$ which is smaller than the corresponding matrix in Eq. B.10. This would reduce the complexity of solution significantly.

- $(v_{wake,\,MDT} - v_{correction})$ represents the global inflow as obtained from MDT, which is not already calculated in the near wake inflow. The cosine factor in Eq. B.25 accounts for the component of inflow perpendicular to the chord.

The idea behind this approximation is that $v_{wake,\,MDT}$ will represent the average inflow which is the 'global' downward velocity induced by the shed vortices, while the local effect which is not captured by the MDT, is accounted for by the term $v_{Near,\,wake}$. In this way, a total of $(n_w - m_w)$ number of vortices are neglected. If these vortices are at a significantly large distance from the airfoil, the effect of neglecting them will be minimal. However, if they are

close, then neglecting them would lead to some error in lift and drag, as compared to the total unsteady case, where all vortices are retained.

Structural velocity v_{str} is the same as in Eq. B.11. The term Γ_{corr} is used to correct the conservation of circulation equation, since now some vortices are neglected. The term $v_{avg\,inflow}$ is the contribution of the retained m_w vortices to the global downward velocity. $v_{avg\,inflow}$ is thus subtracted from MDT inflow to avoid double counting. $v_{avg\,inflow}$ is approximated by taking a space-average (over the stroke-plane $-2 \leq x \leq +2$) and time-average (over previous cycle) of the inflow, due to the retained m_w vortices. This calculation is referred to as Reduced Wake calculation including Momentum Disk Theory inflow (RW-MDT).

B.5 Error metric

To judge the effect of approximating v_{wake}, following two error metrics are proposed. These metrics give the error in lift as compared with the quasi-steady case and the UVLM.

B.5.1 Average error metric (non-dimensionalized with RMS quasi-steady error)

This error metric gives the RMS difference between lift in 'current' calculation and lift obtained from 'total unsteady' calculation. Here 'total unsteady' calculation is when the entire wake is retained and calculation is done according to UVLM as described in Section II. 'Current' calculation refers to the case at hand. The RMS difference thus obtained is normalized by the RMS difference when the 'current' case is quasi-steady approximation. Thus, the expression for e_1 is summarized as:

$$Error\ e_1 \quad = \quad \frac{RMS\left\{Lift_{Current} - Lift_{Unsteady}\right\}}{RMS\left\{Lift_{QS} - Lift_{Unsteady}\right\}} \tag{B.28}$$

Consider a case when the approximation in the calculation starts at time t_0. Thus, if

the approximation is started after 2 flapping oscillations, then t_0 corresponds to the time required to complete 2 oscillations. Then at any time t, value of error metric e_1 is given by:

$$e_1(t) = \sqrt{\frac{\frac{1}{t-t_0}\int_{t_0}^{t}\left(L_{Current}(\tau) - L_U(\tau)\right)^2\, d\tau}{\frac{1}{t}\int_{0}^{t}\left(L_{QS}(\tau) - L_U(\tau)\right)^2\, d\tau}} \tag{B.29}$$

From time 0 through t_0, the analysis would be the same as total unsteady calculation. So for this period, value of e_1 would be 0.

e_1 gives an average or RMS error estimate since the instantaneous error $(L(\tau) - L_U(\tau))$ is integrated with respect to time. With the above definition of e_1, $e_1 = 1$ corresponds to quasi-steady calculation, while $e_1 = 0$ means total unsteady calculation.

B.5.2 Instantaneous error metric (non-dimensionalized with RMS quasi-steady error)

The instantaneous error in calculation is estimated using error metric e_2 as:

$$e_2(t) = \frac{L_{Current}(t) - L_U(t)}{\sqrt{\frac{1}{t}\int_{0}^{t}\left(L_{QS}(\tau) - L_U(\tau)\right)^2\, d\tau}} \tag{B.30}$$

Since $(L_{Current}(t) - L_U(t))$ fluctuates with time, so does $e_2(t)$. $e_{2_{QS}}$, i.e. e_2 for quasi-steady case, fluctuates about an approximate mean value of 1, and with an approximate magnitude of 0.7. Magnitude of $e_{2_{QS}}$ can be taken as a reference to judge the values of e_2 during the 'current' calculation.

B.6 Results

For the current analysis, $s_\phi = 0$, $\alpha_\phi = 0$, i.e. airfoil start from an upright position at the right end of the figure. It is assumed that $h_0 = b$, i.e. hinge is located at mid-chord. Frequency of flapping is 1 Hz, amplitude of rotation is 30 degrees, stroke amplitude is 2 units, chord

length is 1 unit, there are 10 bound vortices and 40 vortices are shed per flapping oscillation.

B.6.1 Improvement of quasi-steady result using Momentum Disc Theory

Results of using MDT inflow in the quasi-steady calculation are presented in this section. Value of MDT inflow is obtained as explained in Section V (A).

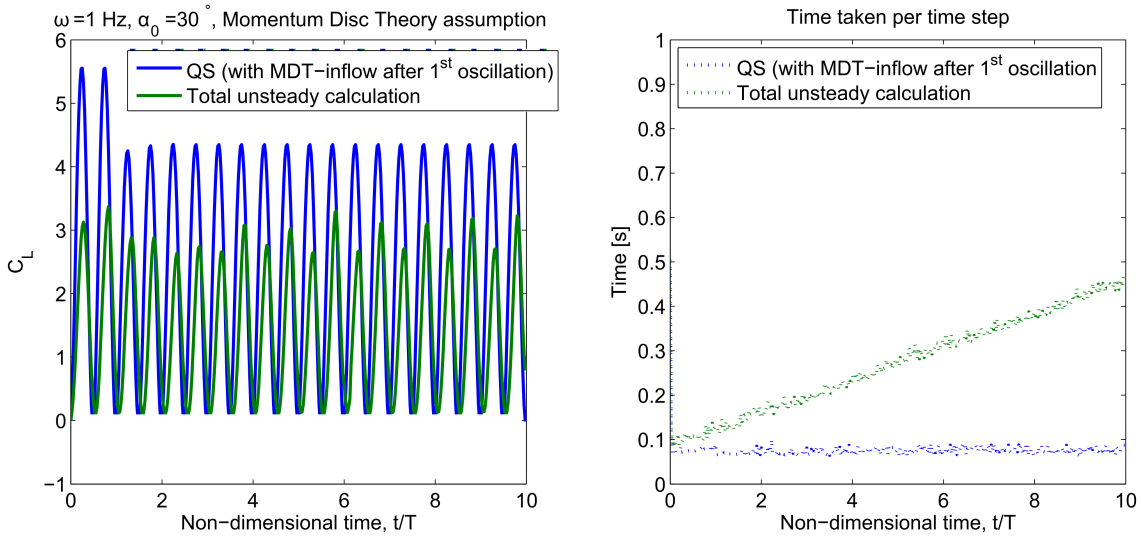

Figure B.3: Improvement of quasi-steady lift using MDT

Results of computation proposed in Section V (A) are shown graphically in Figure B.3. Blue line represents quasi-steady without MDT inflow for the first oscillation, since the approximation starts after the first oscillation. As expected, amplitude of lift for quasi-steady calculation is more than the lift obtained with total unsteady calculation (green line). After the approximation starts, the amplitude of lift decreases and settles to a value closer to the amplitude of unsteady lift as compared to the quasi-steady result. The time required for computation of each time step during the calculation is also shown in Fig. B.3. For the total unsteady calculation, one new vortex is shed at each step, thus time required

for computation of each time step increases during the simulation. Whereas, for quasi-steady calculation time required for computation of each time step is small and approximately the same for all time steps.

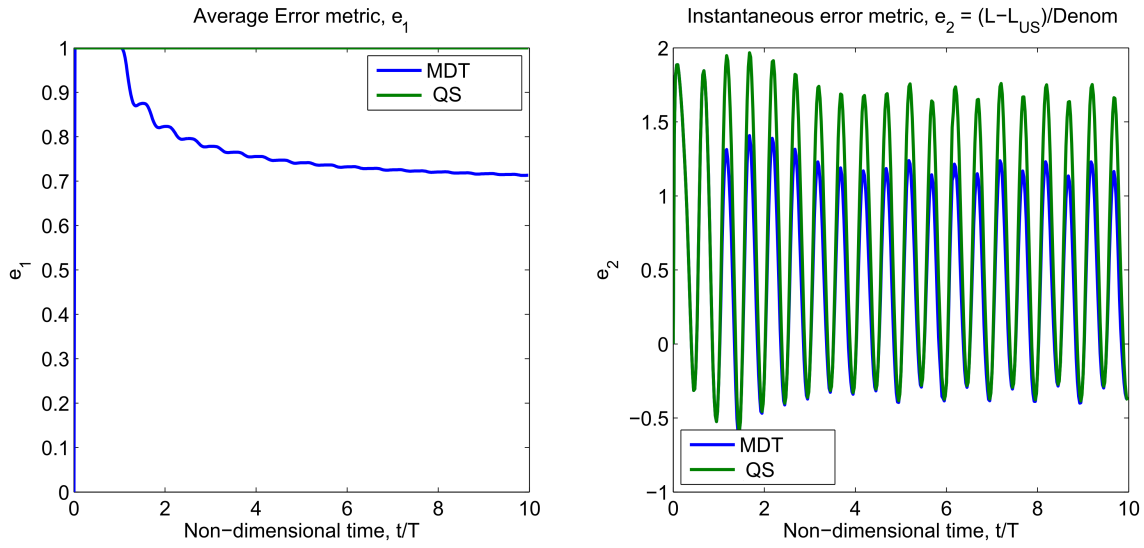

Figure B.4: Effect of considering MDT inflow in a quasi-steady calculation

In Figure B.4, values of error metrics for the current case are plotted against non-dimensional time. Value of e_1 is 1 for the first oscillation, since this oscillation represents quasi-steady calculation. After the approximation starts, e_1 starts decreasing and settles to a value of about 0.7 at the end of 10 oscillations. This can be interpreted as a 30% improvement over the quasi-steady calculation. This is a significant improvement by a minor change in the computation.

Inflow approximated using MDT is constant throughout the stroke plane. However, in reality, the inflow values change significantly over the stroke plane (as seen by the blue lines in Fig. B.2). Thus, MDT alone cannot be used to get the effect of wake. MDT inflow can be understood as the 'global' effect of the shed vortices, which pushes the flow down.

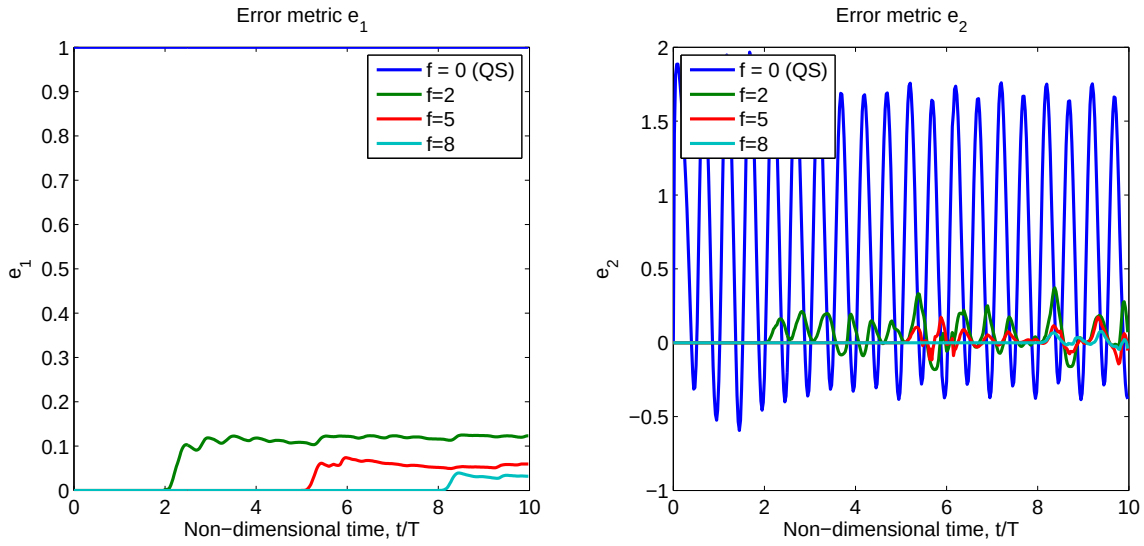

Figure B.5: Effect of retaining a fraction of the shed wake

B.6.2 Results for retaining only a fraction of the wake (RW calculation)

Error metrics e_1 and e_2 for the cases when only a fraction of the wake is retained are shown in Fig. B.5. Thus for results in this subsection, expression for v_{wake} consists of only the term $v_{Near,\,wake}$. Thus these are the results for Reduced Wake (RW) calculation.

Blue line represents quasi-steady calculation, thus e_1 is one throughout. Green line represents the case $f = 2$, which means that vortices in the recent 2 oscillations are retained, while the remaining vortices are neglected. For this case, the result is exact (or matches the total unsteady calculation) till 2 oscillations ($e_1 = 0$ for the first 2 oscillations), and thereafter, e_1 builds up and reaches a value of 0.12 at the end of 10 oscillations. Similarly, for $f = 5$, i.e. retaining vortices shed in the recent 5 oscillations, the result is exact for the first 5 oscillations and reaches a value of 0.05 at the end of 10 oscillations. For $f = 8$, the final value of e_1 at the end of 10 oscillations is 0.03. Thus, at the end of 10 oscillations when we can assume that a steady-state has been reached, the error has been reduced by 88% for the case $f = 2$, by 95% for the case $f = 5$, and by 97% for the case $f = 8$. It is quite

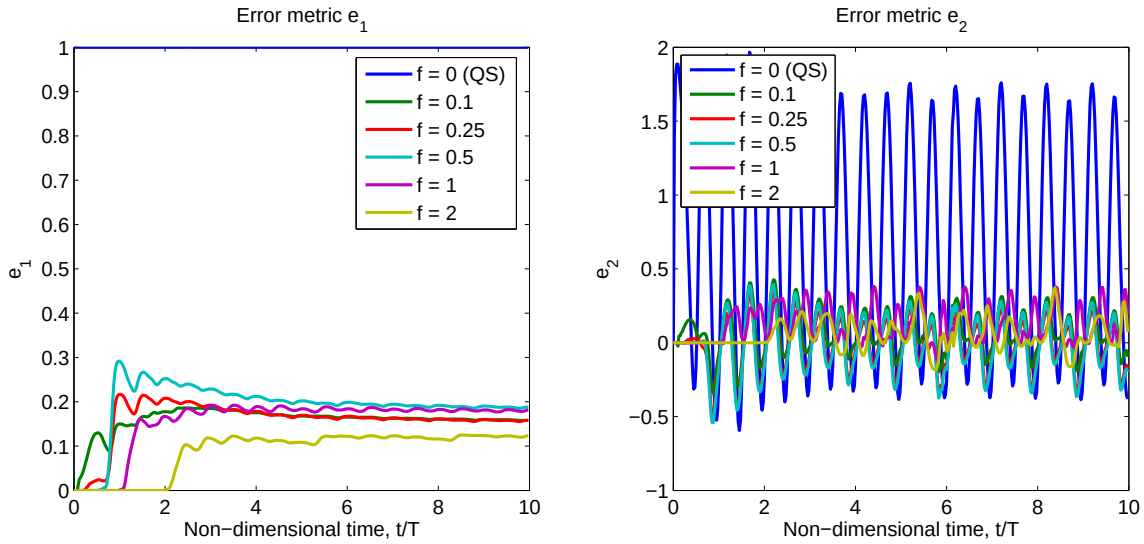

Figure B.6: Effect of retaining a very small fraction of the shed wake

surprising that even with retaining only the recent 2 oscillations, we can capture as much as 88% of the unsteady effects. Instantaneous error metric e_2 also reveals that by retaining only recent 2 oscillations yields significant improvement over quasi-steady case and is quite close to result obtained using total unsteady calculation.

The effect of retaining a very small fraction of the wake is of more significance, for example, retaining only the recent $(1/10)^{th}$ of the wake. This is illustrated in Fig. B.6. For the above simulation, 40 vortices were shed per oscillation. The average error e_1 for retaining just the recent 4 vortices (f=0.1) is shown in the above figure by the green line. Similarly, red line stands for retaining the recent 10 vortices (f=0.25), teal line represents retaining the recent 20 vortices (f=0.5) and so on. It is observed that the average error at the end of 10 oscillations is less than 20%, even for retaining as less as 4 vortices. Thus almost 80% of the unsteadiness is captures by retaining just 4 recently shed vortices.

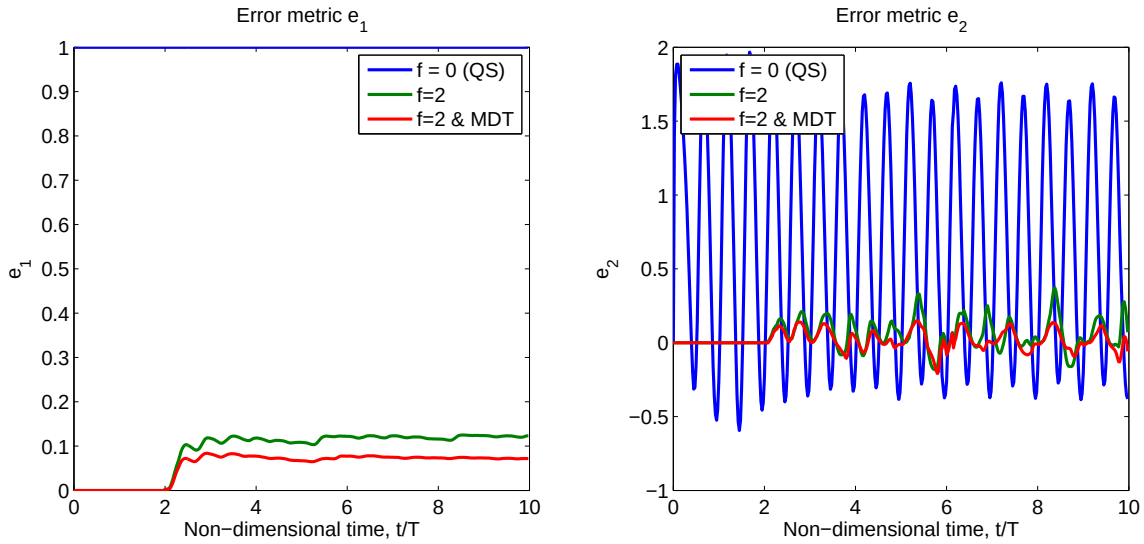

Figure B.7: Effect of retaining a fraction of the shed wake and also using MDT inflow

B.6.3 Results using the proposed reduced order scheme (RW-MDT calculation)

Error metrics e_1 and e_2 for the calculation done as illustrated in Section V (B), i.e. as per RW-MDT calculation, are shown in Fig. B.7. Blue line represents quasi-steady case, green line represents calculation for retaining 2 oscillations without addition of MDT inflow, and red line represents the case for retaining 2 oscillations with addition of MDT inflow. It is seen that there is an improvement in the result from $e_1 = 0.12$ to $e_1 = 0.08$ by addition of MDT inflow. This improvement can also be seen in values of e_2, as fluctuations of the red line are closer to zero than those of the green line.

Similar analysis can also be done to find error metric values with respect to drag. Values of e_1 (calculated for lift and drag separately) at the end of 10 oscillations is plotted against the number of vortices retained, in Fig. B.8. Blue line represents calculation without MDT inflow (RW calculation) while green line represents calculation with MDT inflow (RW-MDT

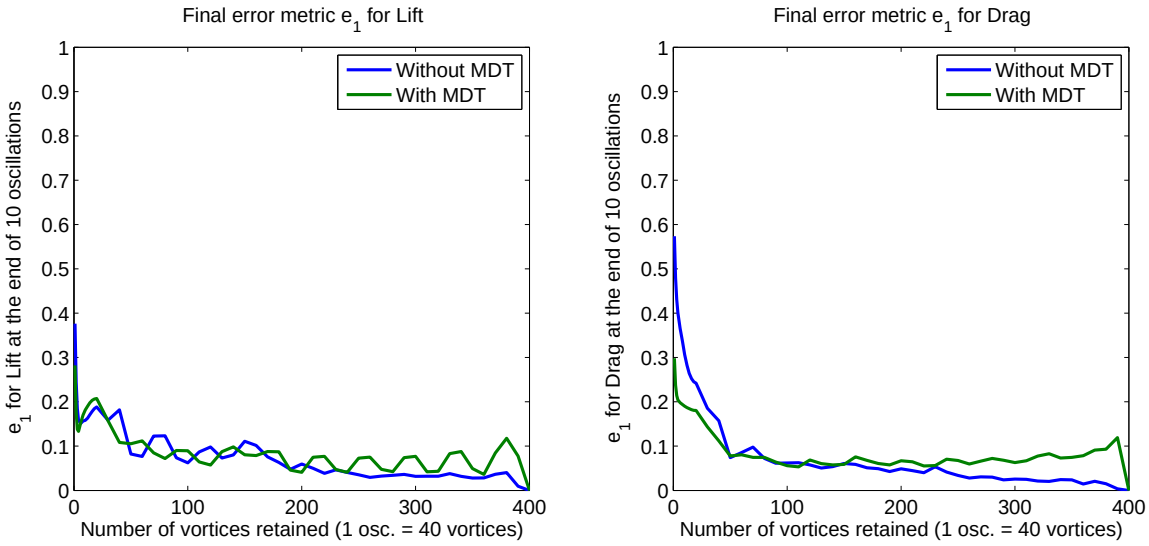

Figure B.8: Effect of retaining only a fraction of the shed vortices, judged from the value of e_1 at the end of 10 oscillations

calculation). For all the simulations, 40 vortices were shed per oscillation. Thus, the figure represents effect or retaining 1 through 400 vortices. As expected, the value of e_1 for retaining all 400 vortices is zero. It is observed that at some (but not all) points, addition of MDT inflow yields a better result. As seen in Fig. B.7, $f = 2$ (or retaining 80 vortices) is one such case. Adding MDT inflow always yields a better drag value, for the case of retaining up to 50 vortices.

The effect of individually changing different parameters in the current simulation on error in lift and drag is presented graphically in Figs. B.9, B.10, B.11, B.12. In these figures value of e_1 at the end of 10 oscillations is plotted against the number of vortices retained.

From this parametric study, it is seen that individually changing parameters (while keeping other parameters constant) such as amplitude of rotation (α_0), semi-chord (b), number of

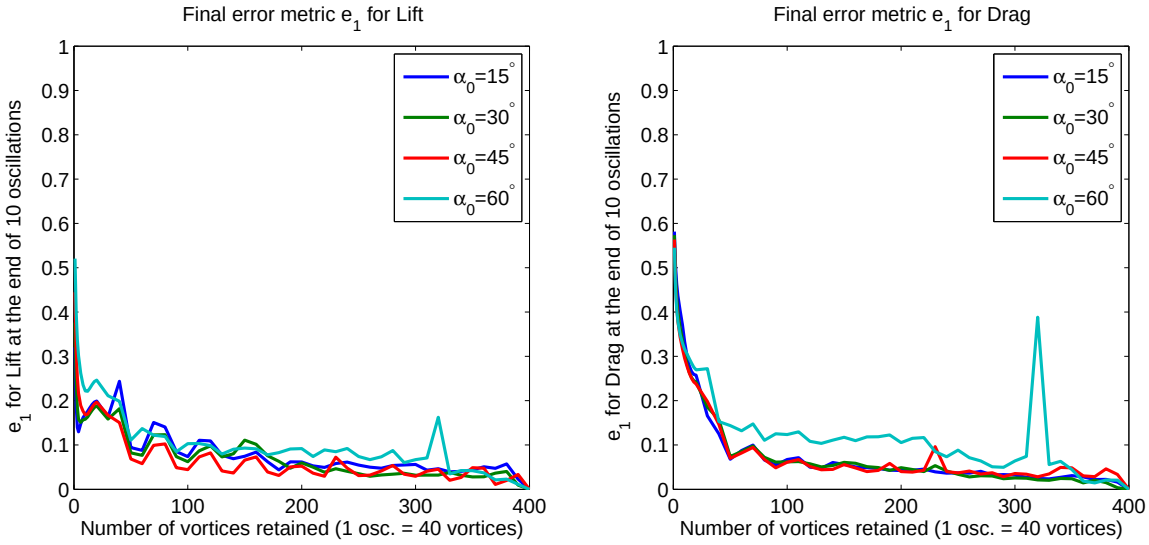

Figure B.9: Parametric study; effect of changing rotation amplitude, α_0

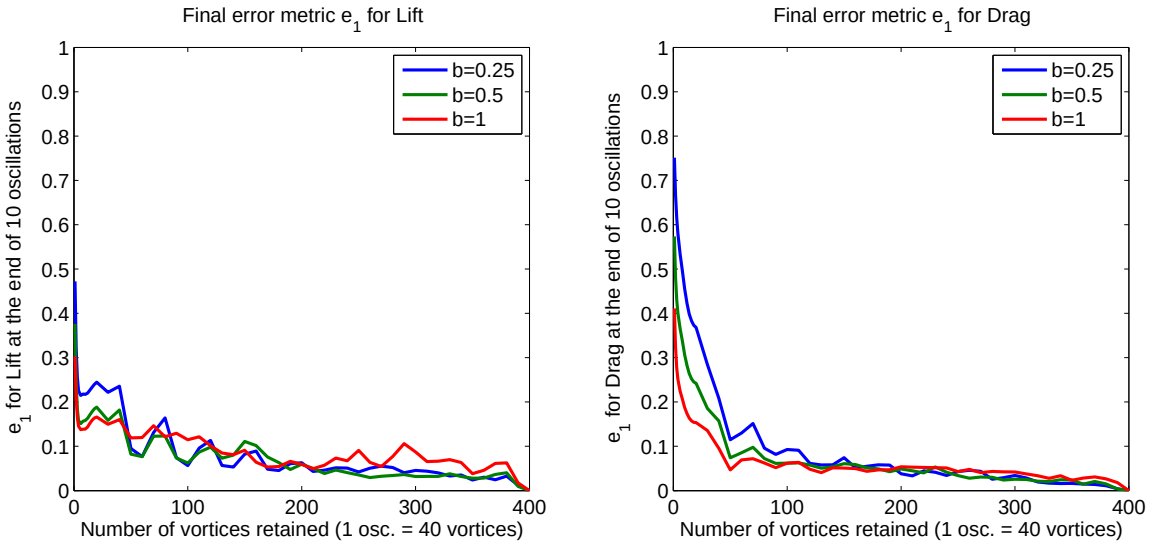

Figure B.10: Parametric study; effect of changing semi-chord, b

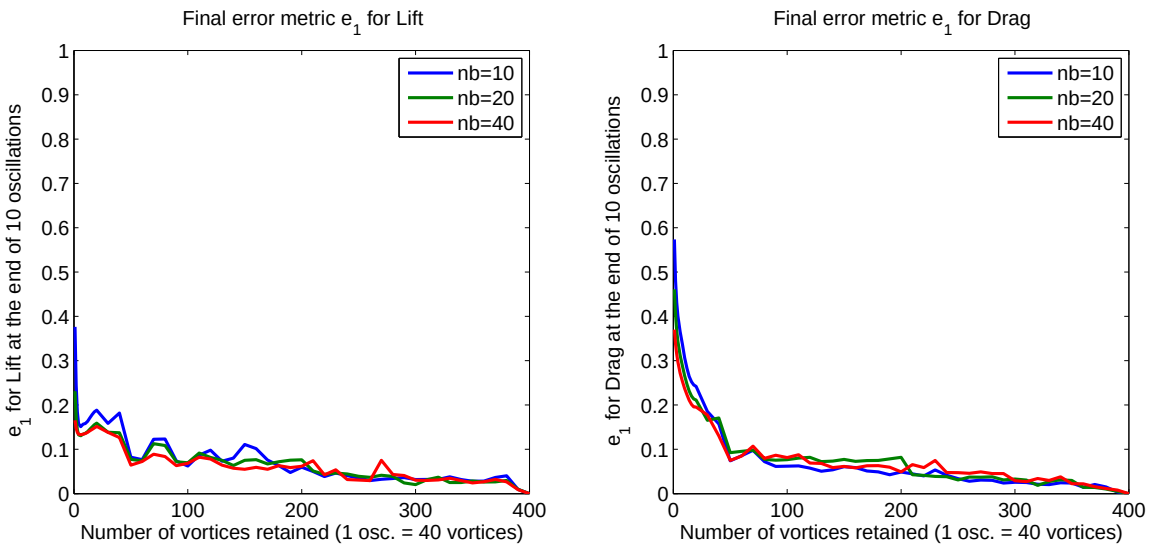

Figure B.11: Parametric study, effect of changing number of bound vortices, n_b

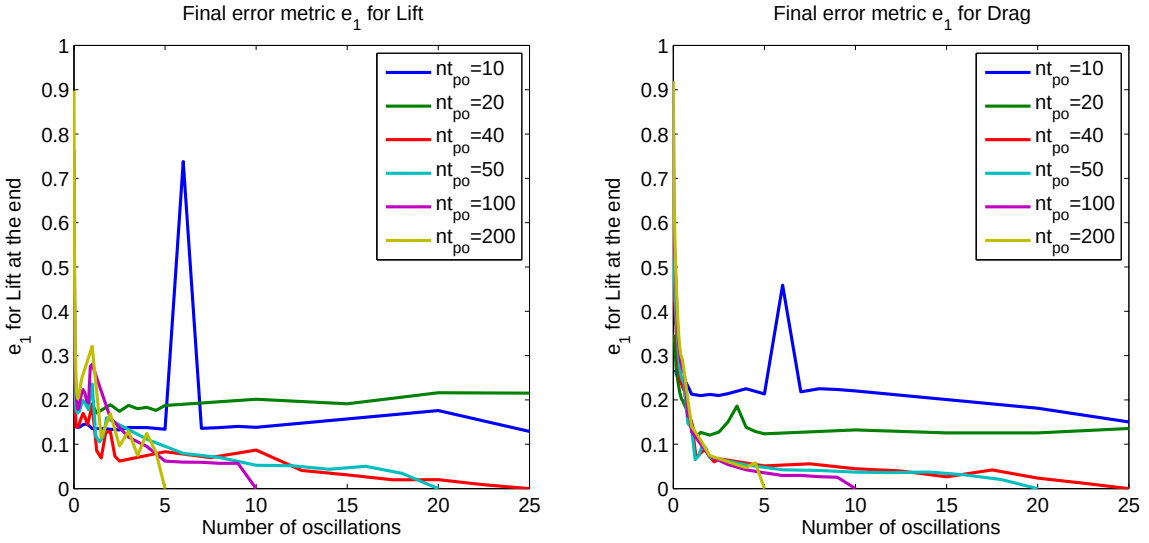

Figure B.12: Parametric study, effect of changing number of time steps per oscillation, nt_{po}

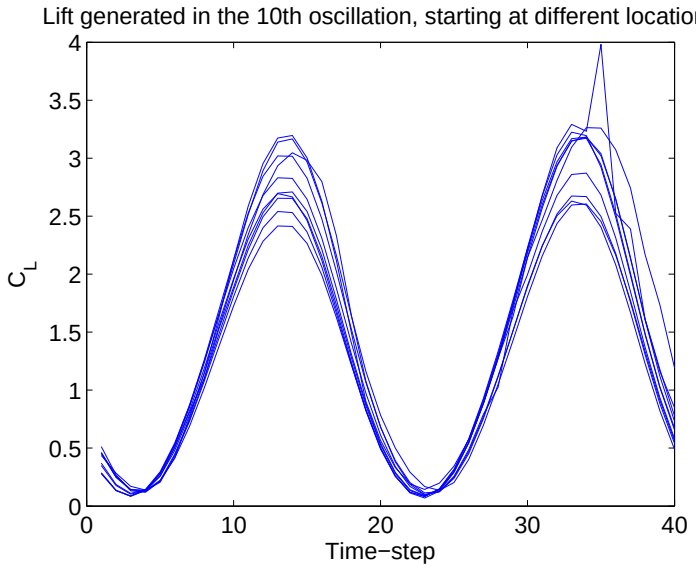

Figure B.13: Lift generated in the 10th oscillation, starting at ten different location on the stroke plane. The calculation is done according to the UVLM and all vortices are retained.

bound vortices (n_b), or number of time steps per oscillation (nt_{po}) does not have a significant effect on the error in lift and drag.

The following section illustrates uncertainty in lift observed due to different starting locations.

B.7 Uncertainty in Lift

Flow field around the flapping airfoil and the lift generated by it depends on the position at which it starts. To analyze this, simulation was run with 10 different starting locations. The first starting location is at the right end of the stroke. The second starting location is at (1/10)th distance from the right end of the stroke. The third starting location is at (2/10)th distance from the right end of the stroke, and so on. Lift generated by the airfoil in the 10th oscillation in each of these cases is plotted in Figure B.13. Lift plotted in this figure is for the 'Total Unsteady' case, i.e. all vortices are retained. There is a large spread in the lift. This could be due to the difference in the wake shed by the airfoil in each of these

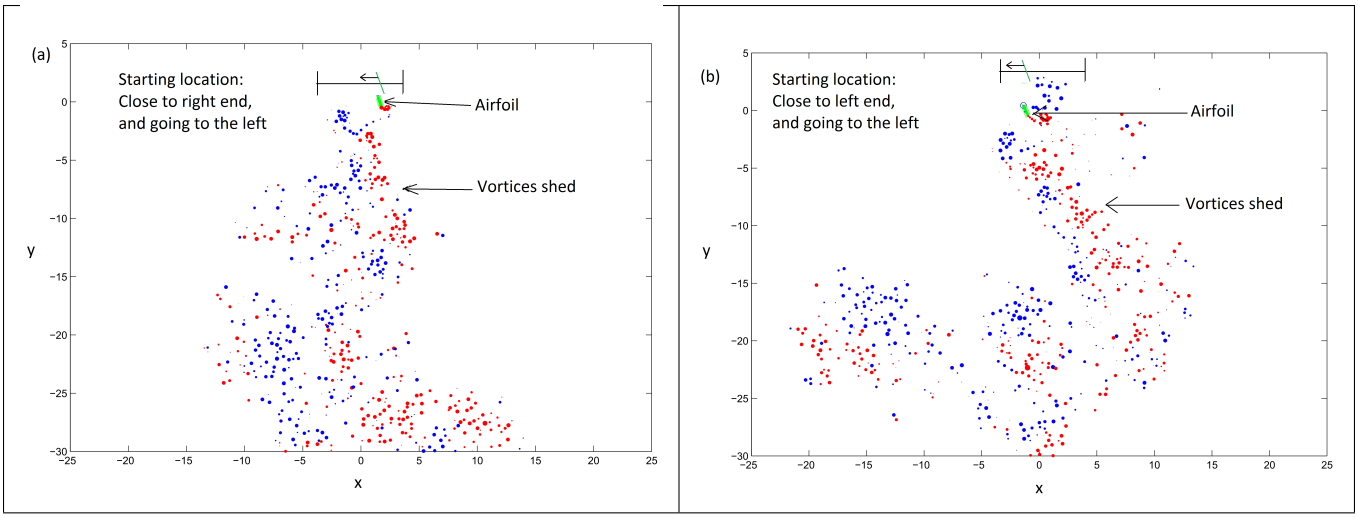

Figure B.14: Comparison of wake shed by airfoil at the end of 15 oscillations, starting at different locations

ten cases. Location and magnitude of the shed vortices influences inflow at the stroke plane which in turn influences the lift generated by the airfoil. The wake pattern (location and magnitude of vortices) shed by the airfoil at the end of 15 oscillations is shown in Figure B.14. The airfoil is represented by a small green line. Blue circles indicate clockwise vortices while red circles indicate counter-clockwise vortices. Radius of the circles is proportional to the magnitude of the vortices. Figure B.14 (a), the airfoil starts at a location (1/4)th stroke length from the right end while in, Figure B.14 (b), the airfoil starts at a location (1/4)th stroke length from the left end. The location and magnitude of the vortices at the end of 15 oscillations is significantly different in the two cases. This confirms that there is uncertainty in the lift generated even for the 'Total Unsteady' case.

The effect of starting at different locations on the lift generated varies according to the method used to calculate lift. Lift generated in the 10th oscillation, using different calculation methods, is shown in Figure B.15. These methods are represented as: 'TU' = Total Unsteady calculation, 'QS' = Quasi-steady calculation, 'MDT' = calculation with Momentum Disc

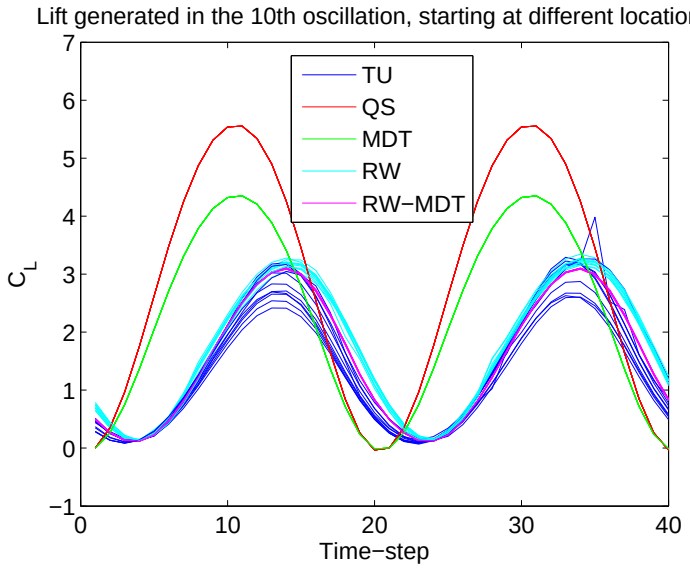

Figure B.15: Lift generated in the 10th oscillation, starting at ten different location on the stroke plane. 'TU' = Total Unsteady calculation, 'QS' = Quasi-steady calculation, 'MDT' = calculation with Momentum Disc Theory inflow, 'RW' = Reduced Wake calculation with retention of 1 oscillation, 'RW-MDT' = Reduced Wake calculation with retention of 1 oscillation and MDT inflow

Theory inflow, 'RW' = Reduced Wake calculation with retention of 1 oscillation, 'RW-MDT' = Reduced Wake calculation with retention of 1 oscillation and MDT inflow. Ten curves are plotted for each of these calculation methods, one for each starting location as described in the previous paragraph. It is observed that the spread in the lift is largest for the TU case. This seems appropriate because the TU calculation incorporates effect of all vortices shed till the end of 10 oscillations. There is almost no uncertainty in the lift produced for the QS and MDT calculations since no vortices are shed in these cases. There is some spread in the lift for the RW and RW-MDT cases, however it is less that the spread in the TU case. This is also expected since only the vortices shed in the recent one oscillation (i.e. 40 vortices) are retained in the RW and RW-MDT cases.

B.8 Summary and conclusion

Present day aerodynamic models used for designing flapping wing MAVs are not accommodative enough to account for all the unsteady phenomenon responsible for generating lift and drag due to flapping motion. This is due to lack of reduced order unsteady aerodynamic models for simulating 3D flapping flight. Present work is the first step to create such a model.

A UVLM calculation with retention of all the shed vortices is not possible for a hovering flapping flight. This is because as time passes, the vortex sheets shed by the wing create singularities in the flow. Thus there is a need of modifying the conventional UVLM to make it applicable for design of 3D flapping flight. In the present work, a reduced order scheme is proposed which uses MDT and UVLM for modeling inflow of a flapping airfoil. Retaining only the recent 2 (out of 10) oscillations helps in capturing 88% of the unsteadiness, as compared to the case of retaining all vortices. Addition of inflow calculated using MDT for this particular case helps in capturing 92% of the unsteadiness. Thus this study shows how the complications of UVLM can be reduced while ensuring that unsteady effects are captured to the desired extent. The reduced model proposed in this work yields significant understanding about the inflow generated by the vortices of flapping airfoil in hover. It is found that retaining only a small fraction of these vortices can lead to significant improvement over quasi-steady results. This 2D scheme can be the basis for creating a 3D unsteady aerodynamic model for flapping flight.

Finally it is shown that there is some uncertainty in the lift generated by the airfoil. This is due to the variation of wake patterns resulting from different starting positions.