Technische Universität München

TUM School of Engineering and Design

Development of Methods for Finite Element-Based Sensitivity Analysis and Goal-Directed Mesh Refinement Using the Adjoint Approach for Steady and Transient Flows

Suneth Warnakulasuriya

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

genehmigten Dissertation.

Vorsitzender:

Prof. Dr.-Ing. habil. Fabian Duddeck

Prüfer der Dissertation:

1. Prof. Dr.-Ing. Kai-Uwe Bletzinger
2. Prof. Dr.-Ing. habil. Roland Wüchner
3. Prof. Dr. Rainald Löhner

Die Dissertation wurde am 22.12.2021 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 29.09.2022 angenommen.

Schriftenreihe des Lehrstuhls für Statik
TU München

Band 53

Suneth Warnakulasuriya

DEVELOPMENT OF METHODS FOR FINITE ELEMENT-BASED SENSITIVITY ANALYSIS AND GOAL-DIRECTED MESH REFINEMENT USING THE ADJOINT APPROACH FOR STEADY AND TRANSIENT FLOWS

München 2022

Veröffentlicht durch

Kai-Uwe Bletzinger
Lehrstuhl für Statik
Technische Universität München
Arcisstr. 21
80333 München

Telefon: +49(0)89 289 22422
Telefax: +49(0)89 289 22421
E-Mail: kub@tum.de
Internet: www.cee.ed.tum.de/st/startseite/

ISBN: 978-3-943683-67-7

Kurzfassung

Wind ist ein komplexes Phänomen, das von lokalen Brisen bis zu Gewittern reichen kann und von komplexen Hindernissen in der Umgebung abhängt. Diese Arbeit konzentriert sich auf die Entwicklung von Methoden zur Analyse von Sensitivitäten und zur problemorientierten Netzverfeinerung für solche Strömungen.

Zwei-Gleichunge Reynolds Averaged Navier-Stokes (RANS) Turbulenzmodelle für $k-\epsilon$, $k-\omega$ und $k-\omega-sst$ mit neuartigen Stabilisierungsformulierungen werden so entwickelt, dass ihre Möglichkeiten zur Entwicklung von adjungierten Sensitivitätsanalysen und ihre Eignung für MPI-parallele-Frameworks für Hochleistungsberechnungen erhalten bleiben. Nach der Bewertung der Genauigkeit und der Möglichkeiten der neuartigen Stabilisierung wird diese verwendet, um die Anwendbarkeit der vorgenannten Turbulenzmodelle im Zusammenhang mit Fluid-Struktur-Interaktionsproblemen und grundlegenden Problemen des Windingenieurwesens zu bewerten.

Danach wird eine effiziente Methodik für eine Sensitivitätsanalyse im stationären Zustand entwickelt und in einem parallelen MPI-Framework realisiert. Darüber hinaus wird sie in einem Optimierungsproblem eingesetzt, um die Abhängigkeit von Turbulenzmodellen bei der Erzielung optimaler Formen zu ermitteln. Anschließend wird sie auf die Sensitivitätsanalyse für instationäre chaotische Probleme erweitert. Es werden zwei neue Stabilisierungsmethoden entwickelt, um das exponentielle Wachstum des adjungierten Lösungsfeldes aufgrund des Schmetterlingseffekts zu kontrollieren. Zuerst werden zeitlich gemittelte Größen des transienten Problems mit RANS-Turbulenzmodellierung verwendet. Danach wird eine kontrollierte künstliche Diffusion zur adjungierten Lösung hinzugefügt. Anschließend wird eine systematische Bewertung der Sensitivitäten mit Finite-Differenzen-Lösungen durchgeführt. Die Genauigkeit der ersten Stabilisierungsmethode hängt von der Genauigkeit des Turbulenzmodells ab. Die zweite Stabilisierungsmethode zeigt eine bessere Genauigkeit als die erste Methode. Dies wird durch die Anwendung auf ein Wirbelabwurf-Frequenz-Optimierungsproblem validiert. Schließlich wird eine Methodik zur Erlangung räumlicher Diskretisierungen entwickelt, getestet und für 2D und 3D validiert. Sie verwendet den neu entwickelten adjungierten Ansatz für stationäre und instationäre Probleme.

Abstract

Wind is a complex phenomenon that may consist of local breezes to thunderstorms and depends on complex surrounding obstacles. This thesis focuses on developing methodologies to analyze sensitivities and to perform goal oriented adaptive mesh refinement for such flows.

Two equations Reynolds Averaged Navier-Stokes (RANS) turbulence models for $k-\epsilon$, $k-\omega$ and $k-\omega-sst$ with novel stabilization formulations are developed such that their abilities to develop adjoint sensitivity analysis and suitability for MPI parallel framework for high performance computing are preserved. After assessing accuracy and capability of the novel stabilization, it is used to evaluate applicability of aforementioned turbulence models in the context of fluid-structure interaction problems and basic wind engineering problems.

Thereafter, an efficient methodology for steady state sensitivity analysis is developed and realized in MPI parallel framework. Furthermore, it is used in an optimization problem to identify the dependence on turbulence models in obtaining optimum shapes. Next, it is extended to sensitivity analysis on transient chaotic problems. Two novel stabilization methods are developed to control exponential growth of the adjoint solution field due to presence of butterfly effect. Firstly, time averaged quantities of the transient problem with RANS turbulence modelling is used. Secondly, controlled artificial diffusion is added to the adjoint solution. Afterwards, systematic assessment of sensitivities are carried out with finite difference solutions. Accuracy of the first stabilization method depends on the accuracy of the turbulence model. The second stabilization method shows improved accuracy compared to the first method. This is further validated by using it on a vortex shedding frequency optimization problem. Finally, a methodology to obtain spatial discretizations which can achieve given accuracy in a goal (i.e function) is developed, tested and validated for 2D and 3D using the novel adjoint approach for steady and transient problems.

Acknowledgements

In writing this document, which represents the culmination of the work that has kept me busy during the last few years, it becomes clear to me how much of it has been shaped by the influence of many people who have helped me reach this point. To all of them, I am in debt and, while I am sure that making a complete list would be impossible, I will try to at least mention as many as I can.

First of all, I would like to thank Prof. Dr.-Ing. Kai-Uwe Bletzinger for giving me the opportunity and freedom to work as a member of the Chair of Structural Analysis. I would like to thank Prof. Dr.-Ing. habil. Roland Wüchner for lending his precious time for long discussions and providing me guidance throughout my work. I would like to thank Dr.-Ing. Michael Andre for directing me in the right direction in the initial phase of my work. I would also like to take this opportunity to thank Dr. Jordi Cotela Dalmau for his constant guidance in my project.

Furthermore, I would like to thank all of my colleagues at the Chair of Structural Analysis for all the support given to me throughout my period at the chair, and also I would like to thank colleagues at International Centre for Numerical Methods in Engineering, Barcelona for assisting me with my methodological developments in the open source Kratos Multiphysics framework.

I would also like to express my gratitude towards the International Graduate School of Science and Engineering for funding a large part of my work. I would also like to thank the examiners for taking the time to review my thesis.

Finally, I would like to thank my family, who have supported and encouraged me every step of the way, and I would also like to give special thanks to my wife Eshani Warnakulasuriya who endured all the different flavours of times with me patiently and supportingly. I would like to take this opportunity to thank Nilami Weedagama Jayatunge and Athula Jayatunge as well who have been my extended family, welcoming and supporting me since the day I stepped into Germany.

Suneth Warnakulasuriya
München
October 2022

Contents

List of Abbreviations

ABL Atmospheric Boundary Layer.

AD Adjoint.

AFC Algebraic Flux Correction.

ALE Arbitrary Lagrangian-Eulerian.

AMR Adaptive Mesh Refinement.

AoA Angle of Attack.

BARC Benchmark on the Aerodynamics of a Rectangular 5:1 Cylinder.

CDR Convection-Diffusion-Reaction.

CFD Computational Fluid Dynamics.

CFL Courant–Friedrichs–Lewy.

CSM Computational Solid Mechanics.

CWD Crosswind Diffusion.

CWE Computational Wind Engineering.

DES Detached Eddy Simulation.

DFT Discrete Fourier Transform.

DMP Discrete Maximum Principle.

DNS Direct Numerical Simulation.

FCT Flux-Corrected Transport.

FD Finite Difference.

FEM Finite Element Method.

FSI Fluid-Structure Interaction.

FVM Finite Volume Method.

LBB Ladyzhenskaya-Babuška-Brezzi.

LES Large Eddy Simulation.

LSS Least-Squares-Shadowing.

NS Navier-Stokes.

PDE Paritial Differential Equation.

QOI Quantities of Interest.

RANS Reynolds Averaged Navier-Stokes.

RFC Residual-based Flux-Corrected.

RST Reynolds Stress Tensor.

SOLD Spurious Oscillations at Layers Diminishing.

SUPG Streamline Upwind Petrov-Galerkin.

TVD Total Variational Diminishing.

URANS Unsteady Reynolds Averaged Navier-Stokes.

VMS Variational Multi-Scale.

List of Symbols

Material properties

ρ	mass density
ν	fluid kinematic viscosity
ν_t	fluid turbulent kinematic viscosity

Flow properties

$\underline{u}$	velocity
u_τ	friction velocity defined in equation (2.21)
p	fluid pressure
k	turbulent kinetic energy
ϵ	turbulent energy dissipation rate
ω	turbulent specific energy dissipation rate
$\underline{x}$	position
y	wall distance
t	Time
τ	dimensionless time defined in equation (2.7)
h	Characteristic element length

Dimensionless numbers

Re	Reynolds number
Re_τ	friction Reynolds number
κ	Von Kármán constant
y^+	dimensionless wall distance
k^+	dimensionless turbulent kinetic energy defined in equation (2.28)
u^+	dimensionless velocity defined in equation (2.26)
τ^+	dimensionless friction shear stress defined in equation (2.28)
β_{exact}	dimensionless scalar transport equation stabilization controlling factor defined in equation (2.13)
C_D	drag coefficient
C_L	lift coefficient

Mathematical symbols

$\underline{(\cdot)}$	Vector
$(\cdot)_i$	i^{th} component of vector
$\underline{(\cdot)}$	Matrix
$(\cdot)_{ij}$	i^{th} row j^{th} column component of matrix
$(\cdot)^T$	Transpose
$(\cdot)^H$	Finite element coarse mesh discretization
$(\cdot)^h$	Finite element refined mesh discretization
$(\cdot)^{\tilde{H}}$	Finite element coarse mesh linearly interpolated on refined mesh
$(\cdot)^a$	a^{th} nodal index
$(\cdot)^{ab}$	a^{th} nodal index residual and b^{th} nodal index degree of freedom
$\lvert(\cdot)\rvert$	Absolute value of scalar
$\lvert\underline{(\cdot)}\rvert$	Magnitude of vector
∇	gradient operator
$\nabla\cdot$	divergence operator
$\tilde{(\cdot)}$	corresponding adjoint for $(\cdot)$
$\overline{(\cdot)}$	Mean value
$\dot{(\cdot)}$	time derivative
$(\cdot)^n$	n^{th} time step
$(\cdot)_{\phi,\psi}$	$(\cdot)_\phi$ partial differentiated w.r.t. ψ
δ^{ab}	Kronecker notation

Chapter 1

Introduction

Wind is a complex phenomenon which can span in the range from local breezes which last for hours to thunderstorms which last for minutes. Spatially it can scale from a few meters to a few kilometers. The complex surroundings on which wind is acted upon, turbulence characteristics, wind speed, wind speed distribution contribute towards increasing this complexity. Most of the complexity stems from the Atmospheric Boundary Layer (ABL) which is the layer of wind attached to the surface. It may span up-to a few 100 meters or to a few kilometers. This consists of multiple temporal and spatial scales and it is highly influenced by complex topographies such as urban environments. Understanding the wind behavior in ABL is of great importance for many fields such as thermal comfort (Fröhlich et al. [47]), renewable energy generation (Mathew et al. [91]).

In the context of Computational Wind Engineering (CWE), it is required to determine the wind loads acting on structures and structural responses. This involves understanding the wind effects in the ABL for specified use cases such as in wind flow in urban topographies, wind turbine design. Mainly there are two different approaches to determine the wind effects in

ABL. First one is wind tunnel testing where a scaled down version of a urban city or a wind turbine model is constructed using scaling laws (Molina et al. [98], Ricci et al. [112], and Uehara et al. [130]). One major restriction in this method is, the scaled down version is not able to replicate all non-dimensional parameters the same as of the actual problem. Apart from this, wind tunnel tests are also expensive because they need to construct physical scaled down models and they need heavy initial investments. They also have difficulties in analyzing the aerodynamic effects of scaled down structures due to their higher stiffness. Second approach is performing numerical wind tunnel tests. Numerical wind tunnels use computational resources to mimic the actual wind effects. Since this numerical wind tunnel is only bounded by computational capacity without any physical bounds such as physical size of structures, it allows replicating actual problem to actual scale with actual flow and structural properties. This allows to carry out the aerodynamic effects analysis of a given structure as well. In addition, cost of a numerical wind tunnel test is less expensive than its actual wind tunnel counterpart.

1.1 Motivation

Although, numerical wind tunnels are less expensive than actual wind tunnels, they also have some associated costs related to labor hours. Figure 1.1 illustrates the break down of relative time used for each step in conducting a numerical wind tunnel test which includes human time requirements as well as computational time requirements. It is of interest in many engineering fields to reduce total time required for a numerical wind tunnel test as in work done by Bouhal et al. [20], Gan [50], and Spence et al. [120]. Therefore one part of this thesis focuses on investigating methodologies to reduce the overall time consumption of the numerical wind tunnel test. It can be seen that, considerable amount of time in a numerical wind tunnel test is spent on steps from "Design of model" through "Mesh manipulation" and "Run simulation" which sums to approx. 80% of the total time (refer orange regions in figure 1.1). All of these steps involve handling a mesh. It is also evident that, if a mesh is coarsened, then the overall time consumption can be reduced at the expense of the accuracy of its solution. This suggests that, it is important to have an efficient and effective mesh generation process which can produce meshes to achieve required accuracy.

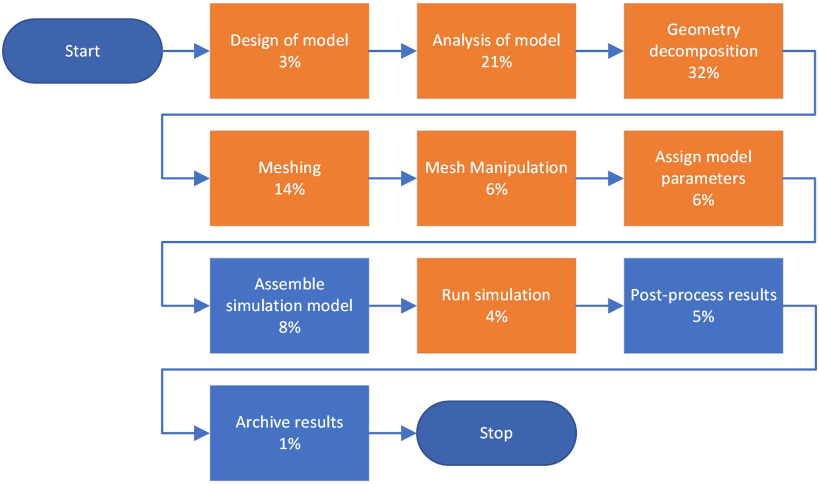


Figure 1.1: Time spent on each step of a numerical wind tunnel test (Boggs et al. [19]).

The required accuracy of a numerical wind tunnel test is also subjective in CWE. This mainly depends on problem being solved and the context in which it is being solved (such as manufacturing, safety, etc). As CWE problems are most of the time transient and chaotic, it is challenging to replicate the accurate time instantaneous values of a numerical wind tunnel test as in the actual problem. This is because, chaotic problems are highly sensitive to initial conditions which is also known as "Butterfly effect" (Lorenz [83]). Butterfly effect causes small perturbations in initial conditions to grow at exponential rate and this growth is propagated through time. This results in a different state of the solution after some time. However, statistical quantities of the flow are bounded. Therefore, only statistical Quantities of Interest (QOI) (such as mean drag, mean lift) are of interest for many applications in CWE.

1.2 State of the Art

In CWE, one of the main governing equations for wind is Navier-Stokes (NS) equation which determines fluid motion, consequently evaluated QOIs. Turbulence modelling is a crucial part in solving problems governed by NS which determines required complexity of a mesh and accuracy of its solution. Direct Numerical Simulation (DNS) is considered to be the most accurate turbulence resolving computational method. It resolves all spatial and temporal scales present in turbulence, including Kolmogorov microscales, without using any turbulence model (Orszag [104]). However, it can be prohibitively expensive for most practical engineering problems in CWE because it requires highly refined mesh. Therefore, the economical Large Eddy Simulation (LES) and Reynolds Averaged Navier-Stokes (RANS) methods have gained attention among the CWE community. Even though LES is economical compared to DNS (Smagorinsky [118]), it is still very expensive to use for many problems of practical interest, since it requires a relatively fine mesh and implies the solution to the temporal evolution of the problem. Due to the butterfly effect, most of the practical problems have interest in their statistical nature in Computational Fluid Dynamics (CFD) such as mean flow. This is advantageous for RANS, where it is developed by time-averaging fluid flow governing NS equations (Reynolds [111]). This also helps to circumvent the high initial condition sensitivity exhibited by NS problems. This time-averaging makes it more appropriate and least expensive method for obtaining mean flow characteristics, and statistical quantities. Unsteady Reynolds Averaged Navier-Stokes (URANS) is the transient extension of the RANS, which is used to solve transient problems. One way to achieve reduced computational cost is to have a less refined mesh than LES in RANS. Use of wall functions which model near wall effects rather than resolving them also allows to have a less refined mesh in RANS. Another way is to reduce the dimensionality of the problem such as simplifying a given problem from 3D to 2D which allows to have a less computationally expensive mesh. This is used extensively in solving optimization problems (Lyu et al. [87]) or flutter derivative analysis (Nieto et al. [101]).

Another approach to reduce time consumption associated with mesh complexity is to automate the mesh generation process as much as possible; consequently, reducing the human time consumption for wind tunnel tests. Using mesh-free methods (Idelsohn et al. [65], Wang et al. [137], and Zhang et al. [149]) can be advantageous in modelling complex urban cities at the expense of higher computational costs at each solving step, because this requires re-identifying neighbor connectivities after each solution. For an urban city scaling up to a few kilometers may lead to infeasible computational cost if this method is used. On the other hand, Adaptive Mesh Refinement (AMR) methods can be useful in this case where mesh is adapted and optimized based on given criteria which can reduce human time consumption and computational time consumption in wind tunnel tests. Two of the most common AMR methods are based on flow field error minimization (Apel et al. [7], Gourma et al. [53], and Verfürth [133]) and goal oriented error minimization.

In the case of the goal oriented AMR method, it is important to obtain the sensitivity of a goal with respect to discretization points in the domain. This can be achieved mainly through two different methods: The first one is direct sensitivity analysis, where state derivatives are computed for each state variable by evaluating the primal problem, next followed by goal sensitivity calculation, thus making this method expensive. The second approach is adjoint sensitivity analysis, which solves the primal problem and its dual problem to obtain goal sensitivity, thus making it more economical in the cases in which a large number of parameters needs optimization (such as AMR for CWE problems).

The first two stages of the numerical wind tunnel pipeline (refer figure 1.1) involve designing of a model and analysis of a model which sum up to approx. 30% of the overall time. Sensitivity analysis of QOIs can be used to reduce time consumed in these stages (Arriola et al. [8]). Therefore, having information about QOIs in a CWE problem is also important in reducing total evaluation time in a numerical wind tunnel test. Sensitivities can be calculated either by direct method or adjoint method as explained previously. Since most of the problems in CWE are transient and chaotic in nature, they pose a challenge to compute the sensitivities of a statistical QOI. This is because, presence of butterfly effect makes computed sensitivities useless if computed with the adjoint approach, because they grow exponentially with time. Therefore, stabilization of these sensitivities are also a focus in this thesis. One other advantage of having sensitivities to a given QOI is, they can be used to perform optimization of the given QOI while changing the structural shape (Dhert et al. [38] and Yu et al. [148]).

This provides insights to design and analysis stages which enhance the ability to produce more appealing and efficient structures in CWE

1.3 Outline

In summary, all of these aspects in CWE motivate to investigate different approaches in turbulence modelling, sensitivity analysis, different AMR methods, and optimization procedures. Chapter 2 implements, validates and then investigates applicability of RANS and URANS turbulence models in the context of CWE. The same developments are then extended for sensitivity analysis. In the case of steady state sensitivity analysis, a proper framework is designed and introduced in this thesis in chapter 3. It is then further extended for transient sensitivity analysis in chapter 4, where a novel stabilization method is proposed for the adjoint solution of chaotic problems. Finally goal oriented AMR is introduced in chapter 5 for steady and transient problems. These developments are carried out in "Kratos-Multiphysics", the open source Finite Element Method (FEM) framework (Dadvand et al. [37]). It is a highly modular platform which is shared memory / distributed memory parallelizable. It provides basic building blocks for developments in FEM. It is also equipped with optimization procedures, meshing procedures which are useful in aforementioned developments.

CHAPTER 2

TURBULENCE MODELLING AND FINITE ELEMENT METHOD

Turbulence is an important phenomenon in everyday life. To date, a large number of studies have focused on the turbulence in fluid flow. Wind engineering (Prospathopoulos et al. [110]), chemical engineering processes (Harris et al. [56]), and aerospace engineering (Catalano et al. [25]) are a few of them. The recent rapid developments in computational power encouraged most of the researchers in those fields to investigate the use of computational methods such as DNS, LES, and RANS. Since DNS, LES methods are very expensive for most of the CWE applications, this chapter focuses on using RANS and URANS turbulence models to solve CWE problems.

Most of the practical problems in CWE have interest in their statistical nature such as mean flow. Therefore, RANS has the advantage which is developed by time-averaging fluid flow governing NS equations (Reynolds [111]). This time-averaging makes it the more appropriate and the least expensive method for obtaining mean flow characteristics. However, the time-averaging of NS equations introduces Reynolds Stress Tensor (RST) due to non-linearity in NS equations. This RST leads to new unknowns in fluid flow description, which is known as the closure problem. A common approach is to use the Boussinesq hypothesis to reduce the number of unknowns, providing a model of the RST that depends on a single free parameter: the turbulent eddy viscosity. Zero equation models (algebraic models) (Baldwin et al. [11]), one equation models (Spalart et al. [119]) and two equation models (i.e. $k-\epsilon$, $k-\omega$) are a few examples developed based on this hypothesis. Research conducted by Mahmood et al. [88] and Sanderse et al. [115] include comprehensive reviews on the different methods of turbulence models. Among these, the two equation turbulence models are of more interest for practical applications due to their robustness, efficiency, and quality of solutions in different fields of applications (Ahmadi [1], Menter [93], and Svensson et al. [123]).

There is a large number of published studies describing the two previously mentioned two equation turbulence models using the Finite Volume Method (FVM) (Jasak [69] and Peric [107]). The Finite Element Method (FEM) is another emerging fluid flow solving methodology which has made its success originally in Computational Solid Mechanics (CSM). Recently, researchers have shown an increased interest in FEM for fluid flow due to various reasons, one of them being its performance in the context of Fluid-Structure Interaction (FSI) problems. They involve both fluid and solid mechanics and interaction between them (Winterstein et al. [145]). FSI is a crucial part in CWE. Therefore this chapter focuses on developing robust and efficient RANS modeling methodology to solve CWE problems. The same methodology serves as a solid basis for economical adjoint sensitivity analysis; which is used to reduce the overall time consumption of a numerical wind tunnel test with goal oriented AMR as explained in chapter 1.

Solving RANS equations involve solving PDEs for conservation of mass (i.e. continuity equation), conservation of momentum and accompanying turbulence eddy-viscosity Convection-Diffusion-Reaction (CDR) transport equations. In this case, incompressibility is assumed with RANS in most of the applications in wind engineering (Winterstein et al. [145]), chemical engineering (Harris et al. [56]) and many other fields. As described previously, FEM is an emerging technique to solve fluid flows which can be used to solve incompressible RANS PDEs. In FEM, standard Galerkin formulation with linear elements has been one of the most popular methods for solving PDEs due to its simplicity, robustness, and ability to handle complex geometries with unstructured meshes. Despite its success, the standard Galerkin FEM formulation with linear elements is not numerically stable for solving convection or reaction dominated flows. Especially in the case of incompressible NS, it violates the $inf-sup$ or Ladyzhenskaya-Babuška-Brezzi (LBB) condition in solving continuity and momentum conservation equations. In order to satisfy the LBB condition, an option would be to use mixed element formulation where higher order interpolation is used for velocity as in work done by Taylor et al. [126] or by having a weak formulation which is not restricted by the LBB condition, such as in work done by Cotela Dalmau et al. [35].

Apart from solving for momentum and continuity equations, RANS has to solve additional CDR transport equations to determine the RST as discussed before. Especially in two equation models, this involves solving for turbulent characteristic quantities such as turbulent kinetic energy (i.e. k), turbulent energy dissipation rate (i.e. ϵ), and turbulent specific energy dissipation rate (i.e. ω). These CDR transport equations need to be positivity preserving, since negative turbulence quantities represented by them do not have a physical meaning. In practical situations, these turbulent quantities may be dominated by convection or reaction effects. Convection dominated flows are known to produce numerically unstable node-to-node spurious oscillations when standard Galerkin formulation is used (Brooks et al. [21]). Consequently, in addition to challenges arising due to flow dominance from convection or reaction terms, the RANS equations also have to overcome challenges in achieving positivity preserving solutions. Stabilization methods are used to overcome these difficulties: To avoid spurious node-to-node oscillations in convection dominated flows, Hughes et al. [63, 64] developed the generalized Streamline Upwind Petrov-Galerkin (SUPG) stabilization method focusing on advection diffusive transport equations. Another study by Galeão et al. [49] focused on discontinuity capturing methods to add diffusion to overcome convection dominated instabilities. This method produces excessive diffusion in the regions of regular solution, which hinders its solution quality. This drawback was improved by Carmo et al. [24]. Such expositions are unsatisfactory for RANS because they are unable to assure positivity of solution field. Ilinca et al. [66] was successful in implementing the $k-\epsilon$ turbulence model by solving for natural logarithm of k and ϵ quantities while preserving positivity. However, this method produced an ill-conditioned system matrix, thus making it not viable to solve problems involving large complex domains in CWE.

The next class of positivity preserving stabilization methods were developed based on Discrete Maximum Principle (DMP). Burman et al. [23] developed a method to satisfy DMP and was successful in achieving positivity preserving solutions. However, this method requires an acute mesh which is not feasible in practical complex geometries discretized with unstructured meshes. This DMP was further studied by Turek et al. [129], where a Flux-Corrected Transport (FCT) method was developed which adds diffusion based on the discretized system matrix. Furthermore, Strehl et al. [121] was also able to extend FCT schemes for transient CDR transport equations while preserving positivity of solutions. The Total Variational Diminishing (TVD) method introduced by Harten [57] was also successful in attaining positivity preserving solutions. This method was further extended by Kuzmin et al. [79] to obtain $k-\epsilon$ turbulence model flow solution while satisfying positivity of the solution field. John et al. [72] conducted a comparative study on CDR transport equation stabilization methods of the following types; SUPG, Spurious Oscillations at Layers Diminishing (SOLD), and FEM-FCT with non-homogeneous Dirichlet boundary conditions and homogeneous Neumann boundary conditions. In this study, it was shown that the SUPG method and SOLD methods produced overly diffusive solution fields, whereas the FEM-FCT method produced better results.

Furthermore, Joshi et al. [73] presented multi-dimensional CDR transport equation stabilization focusing on preserving positivity. Despite the success in achieving positivity preserving solutions, the discretized system of equations produced highly ill-conditioned matrices thus limiting the applicability of this method for 3D large domain cases. Kuzmin [76] presented the Algebraic Flux Correction (AFC) method for scalar CDR transport equations. This method is successful in achieving positivity preserving solution fields with the least amount of artificial diffusion. However, it requires keeping track of additional variables for each discretized point, thereby increasing memory consumption. Even though the author identifies methodologies to overcome these challenges, developing adjoint sensitivity analysis based on AFC method could be very challenging, and memory intensive.

Although some research has been carried out on having positivity preserving solution techniques in FEM with linear element unstructured meshes for CDR transport equations, they present shortcomings in either robustness, efficiency or in their ability to perform adjoint sensitivity analysis. Therefore, this chapter focuses on introducing a novel positivity preserving stabilization method for FEM used with unstructured grids, while paying attention to the method's ability to develop discrete adjoint sensitivity analysis while minimizing computational cost of evaluating an adjoint problem. Two equation turbulent eddy-viscosity based turbulence models are implemented as use cases for the novel stabilization method.

The chapter is organized as follows: Introduction of the novel Residual-based Flux-Corrected (RFC) stabilization method is in section 2.1. It is compared against the AFC method developed by Kuzmin [76] and the CrossWind Diffusion (Crosswind Diffusion (CWD)) method developed by Joshi et al. [73]. The stabilization methods are then validated using analytical solutions obtained for scalar transport problems as explained in section 2.2. Afterwards, the novel RFC method is extended for RANS turbulence modelling and validated using DNS data and experimental data for benchmarks in section 2.3. All of these stabilization methods are studied against three different turbulent eddy-viscosity models, namely $k-\epsilon$, $k-\omega$ and $k-\omega-sst$. RANS is further extended for URANS in section 2.4, where implementation is validated against data from literature. Finally RANS and URANS are used to solve CWE problems in section 2.5, and results are compared against results from LES.

2.1 Residual-based Flux-Corrected Stabilization Method

This section introduces a novel stabilization method named Residual-based Flux-Corrected (RFC) for scalar transport equations. Following that, turbulence models are introduced, which will be evaluated in subsequent sections.

2.1.1 Problem definition

The problem of interest in this study is the scalar CDR transport equation where ϕ is the unknown transport function. The general strong form is shown in equation (2.1).

$$\frac{\partial \phi}{\partial t} + \underline{u} \cdot \frac{\partial \phi}{\partial \underline{x}} - \frac{\partial}{\partial \underline{x}} \left(\nu \frac{\partial \phi}{\partial \underline{x}} \right) + s\phi = f \tag{2.1}$$

Where $\underline{u} \in \mathbb{R}^d$ is convective velocity, d is dimensionality of domain, $\nu \in \mathbb{R}^+$ is the effective kinematic viscosity, $s \in \mathbb{R}$ is the reactive coefficient, and $f \in \mathbb{R}$ is the source term. Temporal discretization is performed using the Bossak method (Wood et al. [147]). In the case of "steady-state" problems, the time derivative of equation (2.1) is assumed to be zero.

A finite element partition of the computational domain $\Omega \subset \mathbb{R}^d$ is denoted by $\{\Omega^e\}$, the index e ranging from 1 to the total number of elements n_{el}. The boundary of the computational domain Ω is given by Γ and is partitioned as a boundary on which Dirichlet conditions are satisfied Γ_D and those where Neumann conditions are specified Γ_N satisfying equation (2.2).

$$\Gamma = \Gamma_D \cup \Gamma_N \quad and \quad \emptyset = \Gamma_D \cap \Gamma_N \tag{2.2}$$

The space of test functions is denoted by $\Psi = H_0^1(\Omega)$ and the space of trial solutions is denoted by equation (2.3) where $\underline{\eta}$ is the surface outward unit normal acting on Γ boundary.

$$\Phi = \left\{ \phi \in H^1(\Omega) : \phi = g_D \; on\, \Gamma_D, \; and \frac{\partial \phi}{\partial \underline{x}} \cdot \underline{\eta} = g_N \; on\, \Gamma_N \right\} \tag{2.3}$$

A subscript h is introduced to identify the discrete finite element problem derived from equation (2.1). Therefore the finite element formulation for solving ϕ is: Find $\phi^h \in \Phi^h$ for all $\psi^h \in \Psi^h$ such that,

$$\begin{aligned} \int_\Omega \psi^h \left(\frac{\partial \phi^h}{\partial t} + u_i \frac{\partial \phi^h}{\partial x_i} + s\phi^h \right) d\Omega + \int_\Omega \nu \frac{\partial \psi^h}{\partial x_i} \frac{\partial \phi^h}{\partial x_i} d\Omega \\ - \int_\Omega \psi^h f \, d\Omega - \int_{\Gamma_N} \nu \psi^h g_N \, d\Gamma_N = 0 \end{aligned} \tag{2.4}$$

2.1.2 Novel stabilization method

As was pointed out in the introduction to this chapter, the standard Galerkin method produces spurious node-to-node oscillations in numerical solution when convection effects are dominant (Brooks et al. [21]).

This method is also unable to produce positivity preserving solutions for CDR transport equations. A variety of stabilization methods are successful in avoiding node-to-node spurious oscillations and having positivity preserving solutions. The CWD method proposed by Joshi et al. [73], and the AFC method proposed by Kuzmin [76] are two of those chosen stabilization methods to compare against the novel RFC stabilization method. It is important to re-emphasize the fact that the novel RFC stabilization is developed and validated paying attention to its capability in developing discrete adjoint sensitivity analysis and usefulness in solving 2D and 3D complex domain problems.

The evidence presented in chapter 2 introduction suggests that the SUPG stabilization method is able to successfully minimize spurious node-to-node oscillations, making the solution strategy stable. Therefore, SUPG stabilization is used in the novel RFC stabilization method as depicted in equation (2.5).

$$\begin{aligned}\int_{\Omega} N^a \left(\frac{\partial \phi}{\partial t} + u_i \frac{\partial \phi}{\partial x_i} + s\phi \right) d\Omega + \int_{\Omega} \nu \frac{\partial N^a}{\partial x_i} \frac{\partial \phi}{\partial x_i} d\Omega \\ + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau \left(u_i \frac{\partial N^a}{\partial x_i} + N^a |s| \right) R \, d\Omega^e \\ = \int_{\Omega} N^a f \, d\Omega + \int_{\Gamma_N} \nu N^a g_N \, d\Gamma_N\end{aligned} \tag{2.5}$$

Where the superscript a denotes a^{th} node, N is the shape function, R is the residual of the strong form (depicted in equation (2.6)) to the weak form depicted in equation (2.4). The second derivative term in this residual vanishes because linear finite elements are used for the spatial discretization.

$$R = \frac{\partial \phi}{\partial t} + u_j \frac{\partial \phi}{\partial x_j} + s\phi - f \tag{2.6}$$

τ is the SUPG stabilization term calculated using equation (2.7), where $\tau_{dyn} = 0.0$ for steady state problems, and $\tau_{dyn} = 1.0$ for dynamic simulations discretized using the Bossak method. α_b and γ_b are Bossak time discretization constants, and h is element characteristic length (Shakib et al. [117]).

$$\tau = \left[\left(\tau_{dyn} \frac{1-\alpha_b}{\gamma_b \Delta t} \right)^2 + \left(\frac{2|\underline{u}|}{h} \right)^2 + \left(\frac{12\nu}{h^2} \right)^2 + s^2 \right]^{-1/2} \tag{2.7}$$

Furthermore, the weak formulation presented in equation (2.5) is linearized and illustrated in matrix form as shown in equation (2.8) where superscripts a and b denote the a^{th} and b^{th} point in the FEM spatial discretization. Subscript i represents the i^{th} direction.

$$\begin{aligned} &\underbrace{\int_\Omega \left[N^a (1 + \tau |s|) + \tau u_i \frac{\partial N^a}{\partial x_i} \right] N^b \, d\Omega \dot{\phi}^b}_{M^{ab} \dot{\phi}^b} \\ &+ \underbrace{\int_\Omega \left[N^a (1 + \tau |s|) + \tau u_i \frac{\partial N^a}{\partial x_i} \right] \left[u_i \frac{\partial N^b}{\partial x_i} + s N^b \right] + \nu \frac{\partial N^a}{\partial x_i} \frac{\partial N^b}{\partial x_i} \, d\Omega \phi^b}_{C^{ab} \phi^b} \\ &= \underbrace{\int_\Omega \left[N^a + \tau \left(u_i \frac{\partial N^a}{\partial x_i} + N^a |s| \right) \right] f \, d\Omega + \int_{\Gamma_N} \nu N^a g_N \, d\Gamma_N}_{P^a} \end{aligned} \tag{2.8}$$

Finally, using the Bossak method for time integration, equation (2.8) can be further simplified as illustrated in equation (2.9) where $\underline{\mathbf{Q}} = \theta \underline{\mathbf{M}} + \underline{\mathbf{C}}$, with θ being the time step (i.e. Δt) dependent value derived using Bossak method constants.

$$\underline{\mathbf{Q}}\underline{\phi} = \underline{P} \quad where \quad \underline{\mathbf{Q}} = \left[q^{ab} \right], \ \underline{\phi} = \left[\phi^a \right] \ and \, \underline{P} = \left[P^a \right] \tag{2.9}$$

However, SUPG stabilization (refer equation (2.8)) alone does not guarantee positivity of the solution (Codina [33]). Therefore additional stabilization is required. Research done by Kuzmin et al. [78] identifies a positivity preserving solution for a linear system of equations (depicted in equation (2.9)) that can be achieved by making system matrix (i.e. $\underline{\mathbf{Q}}$) an M-matrix (i.e. making $\underline{\mathbf{Q}}$ a non-singular discrete operator such that $q^{ab} \leq 0 \quad \forall a \neq b$ and all the coefficients of its inverse are nonnegative).

The sufficient conditions for positivity preservation are written in inequalities equation (2.10a) and equation (2.10b), given that components of $\underline{\phi}$ and $\underline{P}$ remain positive.

$$\sum_b q^{ab} \geq 0.0 \quad \forall a \tag{2.10a}$$

$$q^{ab} \leq 0.0 \quad \forall b \neq a \tag{2.10b}$$

The discrete upwind operator (i.e. $\underline{\mathbf{D}} = \left[d^{ab}\right]$) depicted in equation (2.11) is applied to satisfy inequality equation (2.10b). But this still does not guarantee a positivity preserving property because the inequality equation (2.10a) is not satisfied.

$$d^{ab} = d^{ba} = -\max\{0.0, q^{ab}, q^{ba}\} \quad \mathit{and} \quad \sum_b d^{ab} = 0.0 \quad \forall a \tag{2.11}$$

Therefore, the matrix depicted in equation (2.12) is also added as part of the stabilization to ensure that inequality equation (2.10a) is also satisfied.

$$\underline{\mathbf{I}}_{pp} = \left[\beta_{pp}\delta^{ab}\right] \quad \mathit{where} \quad \beta_{pp} = \max_{\forall a}\{0.0, -\sum_b q^{ab}\} \tag{2.12}$$

However, adding matrices from equation (2.11) and equation (2.12) to the system matrix will lead to an inconsistent solution with the original problem defined in equation (2.1). Therefore, a non-dimensional scaling factor (i.e. β_{exact}) as depicted in equation (2.13) is introduced as a scaling factor for $\underline{\mathbf{D}}$ and $\underline{\mathbf{I}}_{pp}$. This β_{exact} is proportional to absolute value of strong form residual (refer to equation (2.6)), which tends to zero when spatial and temporal resolutions are high enough, thus making the β_{exact} reaching to zero. This reduces and eventually removes any effect arising from the addition of $\underline{\mathbf{D}}$ or $\underline{\mathbf{I}}_{pp}$ to equation (2.8) when discretization is refined, eventually solving the original problem stated in equation (2.1) which makes the novel RFC stabilization method consistent with the original problem. The absolute residual (i.e. $|R|$) in β_{exact} is scaled by $|\phi|$ to compute relative residual and made non-dimensional by having τ (refer to equation (2.7)).

$$\beta_{exact} = \begin{cases} \frac{|R|\tau}{|\phi|} & \mathit{if} \qquad |\phi| > 0 \\ 0 & \mathit{otherwise} \end{cases} \tag{2.13}$$

Finally, our modified stabilized positivity preserving elemental equation system is illustrated in equation (2.14).

$$\left[\underline{\mathbf{Q}} + \beta_{exact}\left(\alpha_1 \underline{\mathbf{D}} + \alpha_2 \underline{\mathbf{I}}_{pp}\right)\right]\underline{\phi} = \underline{P} \tag{2.14}$$

Where α_1 and α_2 are stabilization control parameters defined by the user. These two parameters are there to guide the solution field to be positivity preserving and to make the condition number of the system matrix better. As will be discussed in section 2.2.4, a wide range of values can be used for these parameters.

2.2 Steady State Scalar Numerical Experiments

All of the stabilization methods introduced in section 2.1 add artificial diffusion to the solution so that they can prevent numerical instabilities and preserve positivity. Their approach is to add the least amount of artificial diffusion to avert shortcomings of the standard Galerkin formulation while minimizing error. Therefore, there will be differences in solutions obtained from the novel RFC stabilization and existing AFC and CWD stabilization methods. In order to have a proper understanding about the significance of the added artificial diffusion from the novel method, it is evaluated against three different problems on which the analytical answer is available (refer section 2.2.1 and section 2.2.2).

Error between the solution from stabilization method and the exact solution is calculated using equation (2.15), where N is the number of points in mesh, m_i is the nodal weight obtained from diagonal component of the lumped mass matrix as in work done by Kuzmin et al. [77], ϕ_i being the i^{th} point value, and ϕ_i^{exact} being its exact solution in the mesh.

$$Error_{L2} = \sqrt{\sum_{n=1}^{N} m_i \left(\phi_i - \phi_i^{exact}\right)^2} \tag{2.15}$$

2.2.1 Circular convection

Problem definition

This experiment is taken from work of Hubbard [62]. The experiment consists of solving a hyperbolic Paritial Differential Equation (PDE) as depicted in equation (2.16).

$$\nabla \cdot \left(\underline{u} \phi \right) = 0 \quad in \quad \Omega = (-1, 1) \times (0, 1) \tag{2.16}$$

Steady circular convective velocity (i.e. $\underline{u}$) is defined in equation (2.17). $\underline{u}$ is divergence free. Therefore equation (2.16) can be re-organized such that the same equation (2.1) can be applied to solve it by making all terms zero except the convective term (Hubbard [62]).

$$\underline{u}\left(x, y\right) = \left[y, -x\right] \tag{2.17}$$

Exact solution for this numerical experiment is given by equation (2.18). The boundary values are also computed from this exact solution.

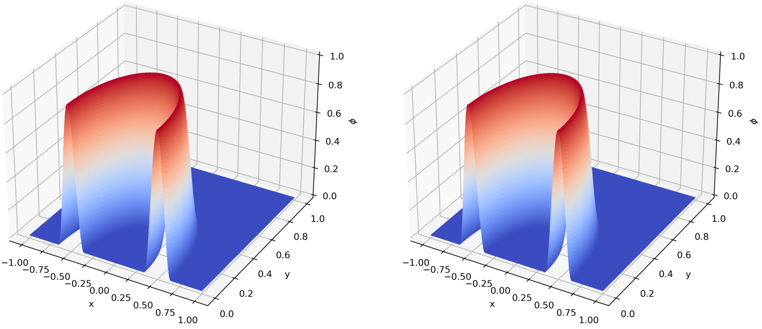
$$\phi\left(x, y\right) = \begin{cases} G\left(r\right) & if \qquad 0.35 \leq r = \sqrt{x^2 + y^2} \leq 0.65 \\ 0 & otherwise \end{cases} \tag{2.18}$$

We have chosen two functions for $G\left(r\right)$ representing a smooth solution (i.e. G_1) and a discontinuous solution (i.e. G_2) as depicted in equation (2.19). The smooth solution is chosen to determine the novel RFC stabilization method's ability to maintain a smooth peak, and the discontinuous solution is chosen to determine the RFC method's ability to preserve positivity of the solution.

$$G_1\left(r\right) = \cos^2\left(5\pi \frac{2r - 1}{3}\right), \qquad G_2\left(r\right) = 1 \tag{2.19}$$

Results

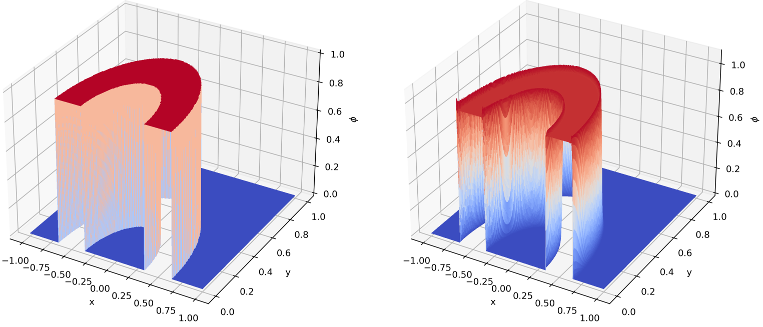
Figure 2.1 illustrates the exact solution and the computed solution using RFC stabilization with $h = 1/256$ mesh for the circular convection problem with the smooth functional. As depicted in table 2.1, the error is small, this suggests that the computed solution is adding diffusion only to stabilize the solution and not to destroy smoothness and peak properties of the solution of a convection dominated problem while preserving non-oscillatory properties. It does not exhibit any under shoots or over shoots near the curvatures of the exact solution for different mesh sizes as well as different RFC stabilization parameters (refer section 2.2.4).



(a) Exact solution

(b) RFC stabilized solution with $h = 1/256$

Figure 2.1: Circular convection with the smooth functional solution comparison.

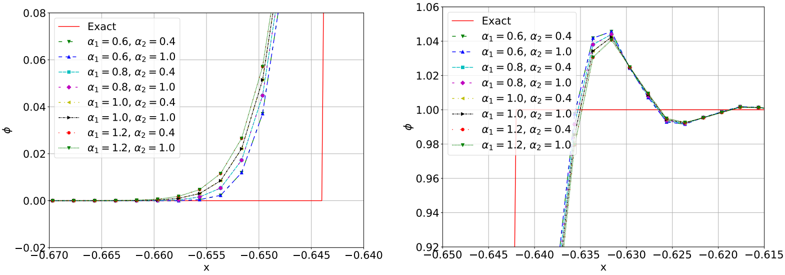


(a) Exact solution

(b) RFC stabilized solution with $h = 1/256$

Figure 2.2: Circular convection with the discontinuous functional solution comparison.

Result of a circular convection experiment with discontinuous functional is illustrated in figure 2.2. This compares the exact solution and the solution obtained from the RFC stabilization method with $h = 1/256$. The RFC stabilized solution shows overshoots near the top of the discontinuity, hence it shows higher $Error_{L2}$ value than its smooth counterpart.



(a) Lower discontinuity

(b) Upper discontinuity

Figure 2.3: Circular convection with discontinuous functional solution comparison for different stabilization parameters, with $h = 1/256$, $x \in [-1.0, -0.5]$.

To illustrate this better, the linearly interpolated ϕ distribution on line $y = 0.1$ is plotted in figure 2.3. Figure 2.3(a) illustrates lower discontinuity and figure 2.3(b) illustrates upper discontinuity where $x \in [-1, -0.5]$ for different α_1 and α_2 values. Both figures do not exhibit significant differences in solution when α_2 is changed, thus indicating α_2 has lower influence on the solution from the added diffusion by RFC stabilization method. We can identify that the solution differs by varying α_1 from 0.6 to 1.2. From figure 2.3(a), at the lower discontinuity, it shows relatively less deviation from exact solution for lower α_1 values indicating better approximation of solution is done when α_1 is reduced. However, no undershoots are shown. This demonstrates that the novel RFC stabilization method is capable of preserving the positivity near discontinuities as well. Figure 2.3(b) illustrates oscillations near the top discontinuity. For lower α_1 values, it shows higher overshoots, and for higher α_1 values, it shows lower overshoots. This is because, α_1 also controls the contributions from discrete upwind operator. Higher the diffusion, smoother the solution, and diffusive the peaks. It is evident that to reduce overshoots, α_1 needs to be reduced which contradicts with the observations on the solution accuracy at the lower discontinuity. This is further analyzed in section 2.2.4.

2.2.2 Body force driven bump

Problem definition

An experiment with scalar steady CDR equation where an exact solution is available also investigated as explained in Barrenechea et al. [12]. This experiment is conducted to solve equation (2.1) in $\Omega = (0,1)^2$ with $\nu = 1 \times 10^{-8}$, $\underline{u} = [3,2]$, and $s = 1$. f is chosen such that equation (2.20) can be obtained as the exact solution. Values at the boundaries are also evaluated using the exact solution.

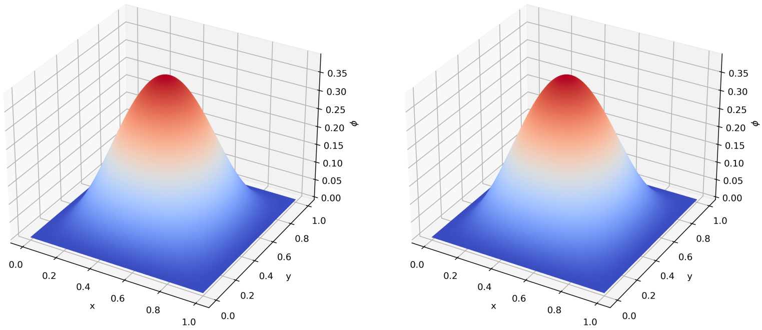
$$\phi\left(x, y\right) = 100x^2(1-x)^2\, y^2\left(1-y\right)^2 \tag{2.20}$$

Results

This experiment focuses on understanding how the novel RFC stabilization method performs if a scalar CDR is presented where convective and reactive terms are dominated over diffusive terms. Figure 2.4 illustrates the exact solution (figure 2.4(a)) and the computed solution via the RFC stabilization method (figure 2.4(b)). If the given exact solution is smooth, consequently no oscillations are visible in this experiment as well. It is also evident from table 2.1, this experiment shows lower error. Consequently it suggests that the proposed novel RFC stabilization method is capable of approximating solution near peaks without losing its characteristics.

2.2.3 Effect of added diffusion in RFC stabilization method

It is important to understand how added diffusion in the novel RFC stabilization method affect the solution. We use the exact solutions from experiments explained in section 2.2.1 and section 2.2.2 to investigate it. Mesh convergence studies have been carried out on these two experiments following the same methodology explained in Kuzmin et al. [77] and respective $Error_{L2}$ values are depicted in table 2.1. All the experiments in this section are carried out with $\alpha_1 = 1.0$ and $\alpha_2 = 1.0$ unless otherwise stated explicitly.



(a) Exact solution

(b) RFC stabilized solution with $h = 1/256$

Figure 2.4: Body force driven bump solution comparison.

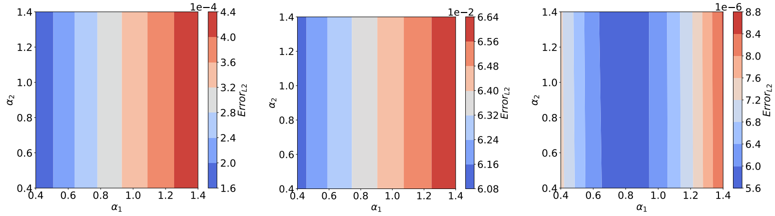
Table 2.1: $Error_{L2}$ values of mesh convergence study for scalar PDE numerical experiments (ref. Kuzmin et al. [77]).

h	Circular convection				Body force driven bump
	Smooth (i.e. G_1)		Discontinuous (i.e. G_2)		
	RFC	Ref.	RFC	Ref.	
1/32	1.70×10^{-2}	5.51×10^{-2}	1.75×10^{-1}	1.52×10^{-1}	4.78×10^{-4}
1/64	4.68×10^{-3}	2.04×10^{-2}	1.14×10^{-1}	1.08×10^{-1}	1.10×10^{-4}
1/128	1.27×10^{-3}	5.95×10^{-3}	8.35×10^{-2}	8.60×10^{-2}	2.69×10^{-5}
1/256	3.38×10^{-4}	1.60×10^{-3}	6.45×10^{-2}	6.01×10^{-2}	6.12×10^{-6}

All of the experiment results shown in table 2.1 are obtained by iteratively solving the governing PDEs until their residuals are converged below the tolerance of 1×10^{-8} using linear triangle element meshes. From table 2.1, we can observe that the smooth functional exhibits less error than the reference, indicating that the proposed novel RFC stabilization method can perform better if a smooth solution is expected. In the case of the discontinuous functional, RFC stabilization method also shows same order of error as the reference indicating, it also suffers in estimating the solution near discontinuities. To clarify these findings, a detailed discussion on the experiment results from section 2.2.1 and section 2.2.2 is carried out henceforth.

2.2.4 RFC stabilization method parameter study

As explained in section 2.1.2, the novel RFC stabilization method introduces two new parameters: α_1 which controls the diffusion added via discrete upwind operator, and α_2, which controls the added mass matrix to satisfy positivity preserving quality (refer to equation (2.14)). Therefore, it is crucial to understand how flow solutions and convergences are affected by these parameters. We have chosen the scalar CDR PDE circular convection and body force driven bump (refer section 2.2.1 and section 2.2.2) experiments to investigate the effects of the novel RFC stabilization method parameters on the solution field.



(a) Circular convection - smooth (b) Circular convection - discontinuous (c) Body force driven bump

Figure 2.5: $Error_{L2}$ for different RFC stabilization parameters.

Figure 2.5 illustrates the $Error_{L2}$ distributions for the RFC stabilization parameters α_1 and α_2 varied within the range $[0.4, 1.4]$. Figure 2.5(a) depicts error distribution of the circular convection experiment with the smooth functional and figure 2.5(b) represents the same experiment with the discontinuous functional. It can be observed from both figures that the error between the solution from the RFC stabilization and the exact solution increases with the increase in α_1, and no significant changes with the increase in α_2. When α_1 is increased, it increases the added diffusion by increasing the contributions from discrete upwind operator, consequently smoothing out the solution. This is clearly evident in the case of discontinuous functional as illustrated in figure 2.3(a) and figure 2.3(b) as well. So in order to achieve non-oscillatory solution at discontinuities, one has to increase α_1 at the cost of the accuracy at the lower discontinuity.

Figure 2.5(c) illustrates the $Error_{L2}$ distribution for the body force driven bump experiment. It also depicts higher error when α_1 is increased as in the case with circular convection experiment. It also depicts an increase in error when α_2 is decreased below 0.6 as well indicating, for a problem governed by CDR equation requires a discrete upwind operator to have non-oscillatory positivity preserving solution. The effect of changes in α_2 is still not having a significant impact on the solution field.

2.3 Applicability to Reynolds Averaged Navier-Stokes Problems

2.3.1 Turbulence models

In this section, turbulence models are introduced as applications of the novel RFC stabilization method and evaluates the method's capabilities and limitations. As explained in section 2.1.2, two equation turbulent eddy-viscosity based RANS models are investigated with different stabilization methods in this study. These two equation models comprise $k{-}\epsilon$, $k{-}\omega$ and $k-\omega-sst$. And for these turbulence models, the implemented solutions of the novel RFC stabilization method are compared against those of the AFC and CWD methods in order to assess shortcomings and advantages.

In the case of turbulence modeling using RANS, it is important to understand how wall treatment is performed. In this study, wall boundaries are assumed to be in logarithmic region, therefore wall modeling with logarithmic wall law is used to approximate near wall effects. Logarithmic wall law is based on wall friction velocity (i.e. u_τ). Since y^+ of our first node in domain falls in the logarithmic region, u_τ can be calculated using two alternative methods: turbulent kinetic energy based (i.e. $u_\tau = f(k)$ based) given in equation (2.21a) (Grotjans et al. [54]) or velocity-based (i.e. $u_\tau = f(u)$ based) given in equation (2.21b) (Von Kármán [135]). κ is the Von Kármán constant based on logarithmic wall law (Högström [59]), and C_μ is a turbulence model constant based on (Versteeg et al. [134]). β is a roughness coefficient used in logarithmic wall law. k at wall boundaries is applied with zero gradient Neumann boundary condition since it is determined by equation (2.21a).

$$u_\tau = C_\mu^{0.25}\sqrt{k} \tag{2.21a}$$

$$u_\tau = \frac{|\underline{u}|}{\frac{1}{\kappa}\ln\left(y^+\right)+\beta} \tag{2.21b}$$

The $k-\epsilon$ turbulence model is implemented based on work done by Kuzmin et al. [79] and Chien [32]. Equation (2.22) is used as wall normal flux for Neumann boundary condition at walls, where ν is fluid kinematic viscosity.

$$g_N = \frac{\partial \epsilon}{\partial x_i}\eta_i = \frac{u_\tau^5}{\kappa\left(y^+\nu\right)^2} \tag{2.22}$$

The $k-\omega$ turbulence model is implemented based on studies conducted by Wilcox [143, 144]. ω wall law is modeled by applying Neumann boundary condition having wall normal ω flux, as illustrated in equation (2.23).

$$g_N = \frac{\partial \omega}{\partial x_i}\eta_i = \frac{u_\tau^3}{\kappa C_\mu^{0.5}\left(y^+\nu\right)^2} \tag{2.23}$$

The $k-\omega-sst$ model is implemented based on research done by Menter et al. Menter [94] and Wilcox [142]. The same wall function for ω is applied in $k-\omega-sst$ model as in $k-\omega$ turbulence model.

All three turbulence model's Neumann wall condition integral contribution for the system of equations are computed using the algorithm shown in algorithm 1. First, y^+ computed from the momentum Neumann condition is taken as an input to compute u_τ based on equation (2.21b) or equation (2.21a). Then, based on the turbulence model, either equation (2.22) or equation (2.23) is used to compute integral contributions to system of equations.

As far as flow solution is concerned (i.e. $\underline{u}$ and p), two methods, Monolithic (ML) and Fractional Step (FS), are investigated in this study. The Monolihtic method is based on study done by Cotela Dalmau et al. [35] where both "transient" and "steady-state" formulations are available. This method is first used with the novel RFC stabilization method, since it allows easy implementation and understanding of convergence behavior

Algorithm 1 Calculate CDR transport equation Neumann wall condition integral.

```
$y^+ \leftarrow y^+$ from momentum Neumann condition (refer to alg. 2)
Compute $u_\tau$ from either equation (2.21b) or equation (2.21a)
Compute Neumann condition integral based on equation (2.22) or
  equation (2.23)
```

Algorithm 2 Calculate momentum Neumann wall condition integral.

```
Solve equation (2.25) for $y^+_{limit}$
Solve equation (2.26) for $u_\tau$
$y^+ \leftarrow \max\{\frac{u_\tau y}{\nu}, y^+_{limit}\}$
Compute new blended $u_\tau$ from equation (2.24a) for condition integral
Compute Neumann condition integral using equation (2.24b) and new
  blended $u_\tau$ from equation (2.24a)
Store $y^+$ on condition
```

for relatively small cases. Afterwards, the FS method based on Codina [34] is also investigated in this study due to the fact that ML method suffers from convergence issues when applied to problems involving large 3D complex domains. This is because the resulting linear system from ML becomes highly ill-conditioned, thus making it challenging to use iterative linear solvers. The FS method is not implemented to obtain a ''steady-state'' solution. It has commonly been assumed that, having large time steps will guide the flow field solution towards steady state. However, this will violate Courant–Friedrichs–Lewy (CFL) number requirements for convergence. Applying smaller time steps will result in oscillations, thus non-convergence for steady solution in FS. Therefore, the pressure gradient modification suggested by Firoozjaee et al. [45] is applied to achieve ''steady-state'' solutions in FS method.

As discussed earlier, walls are applied with wall functions and slip boundary conditions for $\underline{u}$. Nodes falling on boundaries are assumed to be in the logarithmic region, where wall friction velocity (i.e. u_τ) is calculated using equation (2.24a). This hybrid formulation prevents momentum flux from going to zero at separation/stagnation points. Equation (2.24b) shows Neumann wall condition applied on the momentum equations, where t_i

Algorithm 3 Solving strategy.

```
$t \leftarrow 0.0$
for $t \le T_{end}$ do
    while $i \le m_{upt}$ or $\mathbf{u}^i, p^i, \nu^i_{effective}$ not converged do
        while $j \le m_{up}$ do
            Compute $y^+$ on conditions using algorithm 2
            Solve for $\mathbf{u}^j$ and $p^j$
        while $j \le m_{k\epsilon}$ or $\nu_t^j$ is not converged do
            while $c \le m_k$ or $k^c$ is not converged do
                Compute Neumann integral contributions from alg. 1
                Solve for $k^c$
                $c \leftarrow c+1$
            while $c \le m_\epsilon$ or $\epsilon^c$ is not converged do
                Compute Neumann integral contributions from alg. 1
                Solve for $\epsilon^c$
                $c \leftarrow c+1$
            Compute turbulent eddy viscosity $\nu_t^j$
            $j \leftarrow j+1$
        $\nu^i_{effective} \leftarrow \nu + \nu_t$
        $i \leftarrow i+1$
    $t \leftarrow t+\Delta t$
```

is the traction on the Neumann boundary Γ_N.

$$u_\tau = \max\left\{ C_\mu^{0.25}\sqrt{k}, \frac{|\underline{u}|}{\frac{1}{\kappa}\ln\left(y^+\right)+\beta} \right\} \tag{2.24a}$$

$$\int_{\Gamma_N} N^a t_i d\Gamma_N = -\int_{\Gamma_N} N^a \rho u_\tau^2 \frac{u_i}{|\underline{u}|} d\Gamma_N \tag{2.24b}$$

In order to compute the Neumann wall condition integral shown in equation (2.24b), first y^+_{limit} is calculated from equation (2.25) using the Newton Raphson method.

$$y^+_{limit} = \frac{1}{\kappa}\ln\left(y^+_{limit}\right)+\beta \tag{2.25}$$

Then u_τ is computed solving equation (2.26) using the Newton-Raphson method, where $y^+ = \frac{u_\tau y}{\nu}$ and $u^+ = \frac{|u|}{u_\tau}$. y is wall distance at nodes in wall boundary. Afterwards, y^+ is computed from equation (2.27), which is used in computing new blended u_τ from equation (2.24a).

$$u^+ = \begin{cases} y^+ & \textit{for} \quad y^+ \leq y^+_{limit} \\ \frac{1}{\kappa}\ln\left(y^+\right) + \beta & \textit{for} \quad y^+ > y^+_{limit} \end{cases} \tag{2.26}$$

In this, y at wall nodes is calculated as the wall normal distance to its tetrahedral (3D) or triangle (2D) cell center from the wall boundary. Algorithm 2 illustrates steps used in calculating y^+ and momentum Neumann wall condition integral contributions for momentum system of equations.

$$y^+_{wall} = \max\left(y^+, y^+_{limit}\right) \tag{2.27}$$

Finally, algorithm 3 depicts the overall steps involved in solving for momentum equation, k, ϵ using $k-\epsilon$ turbulence model as an example. The outermost for loop iterates over time steps. In case of a dynamic simulation, this loop iterates over all the time steps. Otherwise it will only run once when solving a steady-state problem. The m_{upt} loop is the flow solver and turbulence model coupling loop, where convergence is computed by checking convergence between two adjacent i iteration values of $\underline{u}, p, \nu$, where ν is the total kinematic viscosity. Then m_{up} loop is the momentum equation solving loop which solves for $\underline{u}, p$ and computes appropriate y^+ values on Neumann wall conditions and stores them for later use. The loop with $m_{k\epsilon}$ is solved afterwards, which couples two CDR transport equations: k and ϵ CDR transport equation. Once they are solved, the turbulent eddy viscosity (i.e. ν_t) is updated. The two CDR transport equation coupling iterations will continue until max iterations (i.e. $m_{k\epsilon}$) are reached or ν_t is converged between iterations. Once the two equation turbulence model is converged enough, then fluid effective kinematic viscosity (i.e. $\nu_{effective}$) is updated. Then the same process is repeated again until the flow solution converges. Once the flow solution is converged for the time step, time is incremented to solve for next time step.

We assume logarithmic law for the wall boundary for the examples presented in section 2.3.2 and section 2.3.6, thus forcing nodes at the wall to have at least $y^+ = y^+_{limit}$ (refer to equation (2.27)). Therefore, results are only plotted in range $y^+ \in \left[y^+_{limit}, \infty\right)$. k^+ and τ^+ non-dimensional values are computed using equation (2.28), where ν_t is turbulent eddy viscosity, ν is kinematic viscosity of fluid, $\frac{du}{dy}$ is wall tangential velocity derivative with respect to wall distance (y), and u_τ is the wall friction velocity (obtained from respective benchmark case data).

$$k^+ = \frac{k}{u_\tau^2} \quad \text{and} \quad \tau^+ = \frac{-1}{u_\tau^2}\left(\nu_t + \nu\right)\frac{du}{dy} \tag{2.28}$$

2.3.2 Benchmark Problem Definitions - 2D

Channel flow

Moreover, a 2D channel flow for different friction Reynolds numbers (i.e. Re_τ) is investigated; the numbers are 590, 950, and 2000. DNS flow solution data for $Re_\tau = 590$ case is obtained from studies conducted by Moser et al. [99]. For cases of $Re_\tau = 950$ and $Re_\tau = 2000$, DNS flow solution data is obtained from studies conducted by Hoyas et al. [60, 61].

Figure 2.6 presents the domain and boundaries in this problem. A source term (mimicking pressure gradient $\frac{dp}{dx}$) derived from the respective Re_τ is used to drive the fluid flow. Walls are assumed to be smooth (i.e. $\beta = 5.2$).

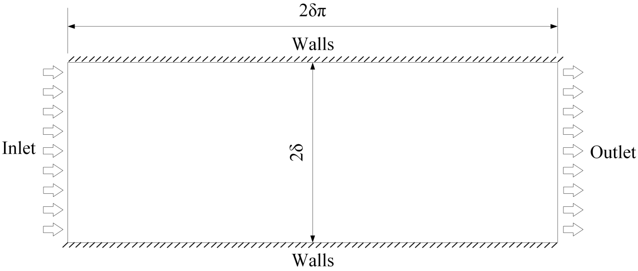


Figure 2.6: Channel flow 2d domain and boundary conditions.

Table 2.2 shows the boundary conditions applied.

Figure 2.7 shows the mesh chosen after a convergence study which is used for spatial discretization. An unstructured triangular mesh is used for spatial discretization.

Table 2.2: Channel and pipe flow boundary conditions.

Variable	Boundary Name		
	Walls	Inlet	Outlet
p	Free	$0\,Pa$	$0\,Pa$
$\underline{u}$	equation (2.24b) u_τ from equation (2.24a)	$Periodic$	$Periodic$
k	Zero gradient		
ϵ	equation (2.22) u_τ from equation (2.21)		
ω	equation (2.23) u_τ from equation (2.21)		

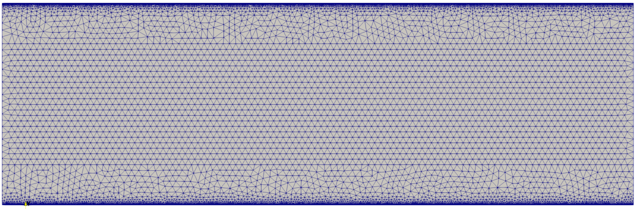


Figure 2.7: Channel flow 2d spatial discretization.

Fluid flow computation is computed with half channel length $\delta = 1\,m$. Therefore, the channel height is 2δ, and channel length is $2\delta\pi$. A boundary layer mesh is applied near wall regions, and a coarser mesh is maintained at the middle of the channel. Minimum edge length of mesh is $1 \times 10^{-3}\,m$, and maximum edge length of mesh is $0.125\,m$.

Backward facing step

The backward facing step is also studied for different turbulence models with stabilization methods introduced in section 2.1. Results are compared against measurement data obtained from research conducted by Jovic et al. [74].

Figure 2.8 illustrates the domain geometry and dimensions as well as boundary conditions used in this problem.

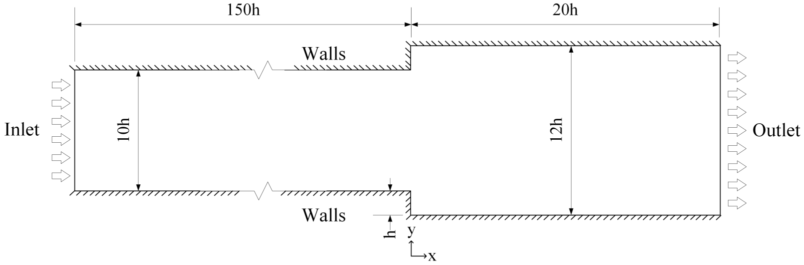


Figure 2.8: Backward facing step 2d boundary conditions.

The backward facing step size is $h = 9.8\,mm$. Entry length for flow development is $150h$, and trailing length after step is $20h$. Channel flow development region height is $10h$, and trailing length channel height is $12h$. x is flow direction and y is wall normal direction. The step is located at $x = 0.0\,m$. Table 2.3 illustrates the boundary conditions applied for this numerical setup, where I_k is the turbulent intensity, and L_T is the turbulent mixing length.

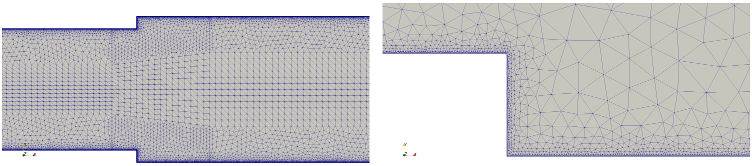
Figure 2.9 illustrates the 2D unstructured triangular mesh used for spatial discretization chosen following the mesh convergence study. Figure 2.9(a) depicts the overall domain, and figure 2.9(b) shows how boundary layer mesh is applied to near wall regions.

2.3.3 Stabilization method comparison

Until now we have discussed effects of added diffusion by novel RFC stabilization method for simple steady scalar PDEs. This section focuses on comparing CWD, AFC, and the novel RFC stabilization methods under one turbulence model, namely $k-\epsilon$ to have a better understanding of how these methods perform with highly non-linear CDR PDEs. Channel flow problems described in section 2.3.2 and backward facing step de-

Table 2.3: Backward facing step 2d boundary conditions.

Variable	Boundary Name		
	Walls	Inlet	Outlet
p	Free	Free	$0\,Pa$
$\underline{u}$	equation (2.24b) u_τ equation (2.24a)	Constant value of $7.72\,m s^{-1}$	Zero gradient
k	Zero gradient	Constant value with $I_k = 6.1 \times 10^{-4}$	
ϵ	equation (2.22) u_τ from equation (2.21)	Constant value with $L_T = 0.0588\,m$	
ω	equation (2.23) u_τ from equation (2.21)		



(a) Overall domain

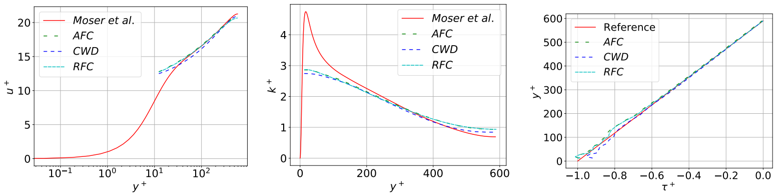
(b) Step close-up

Figure 2.9: Backward facing step 2D spatial discretization.

scribed in section 2.3.2 are used to illustrate the differences in the different stabilization methods.

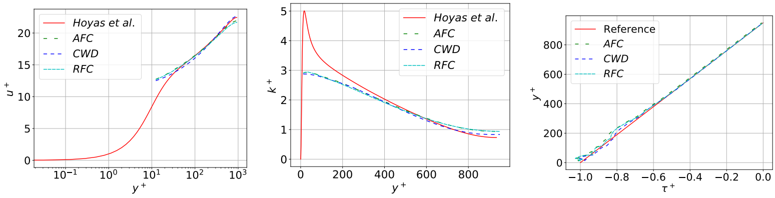
Channel flow results

Figure 2.10, figure 2.11, and figure 2.12 illustrate results obtained from 2D channel flow solutions with $k-\epsilon$ turbulence model for different stabilization methods and different Re_τ (respectively 590, 950, and 2000) using the ML method. u_τ for wall nodes are calculated based on k as given in equation (2.21a) for ϵ.



(a) u^+ vs y^+ (b) k^+ vs y^+ (c) y^+ vs τ^+

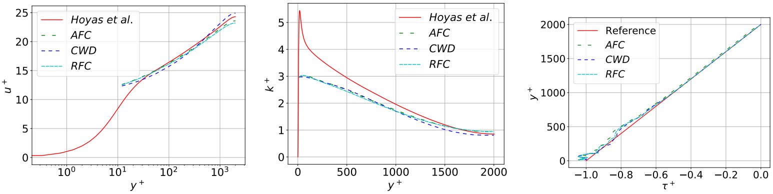
Figure 2.10: Channel flow 2D stabilization method comparison for $Re_\tau = 590$.



(a) u^+ vs y^+ (b) k^+ vs y^+ (c) y^+ vs τ^+

Figure 2.11: Channel flow 2D stabilization method comparison for $Re_\tau = 950$.

A likely explanation for inherent error shown in all the turbulence models includes errors in the turbulence model assumptions used, errors due to assumptions made in $k-\epsilon$ model not being valid near walls, and errors due



(a) u^+ vs y^+ (b) k^+ vs y^+ (c) y^+ vs τ^+

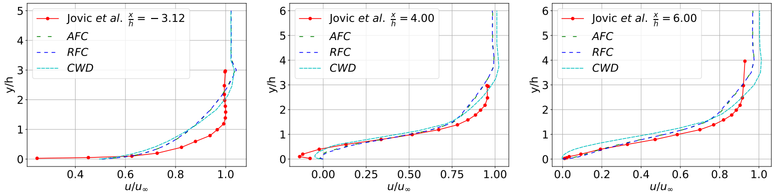
Figure 2.12: Channel flow 2D stabilization method comparison for $Re_\tau = 2000$.

to added artificial diffusion. Specifically, in figure 2.10(b), figure 2.11(b), and figure 2.12(b), we can observe that k^+ near the wall cannot achieve peak as in the DNS results. The most likely reason for that is the wall law assumptions are not valid at wall for k computations. In figure 2.10(c), figure 2.11(c), and figure 2.12(c) exhibit oscillatory behavior near the walls. This is because, τ^+ is calculated with wall tangential velocity gradient which is relatively large near the walls. Gradients computed on linearly interpolated elements result in staggered gradients. This causes higher tangential velocity gradients to have oscillatory behavior thus having oscillations in τ^+ computation.

Backward facing step results

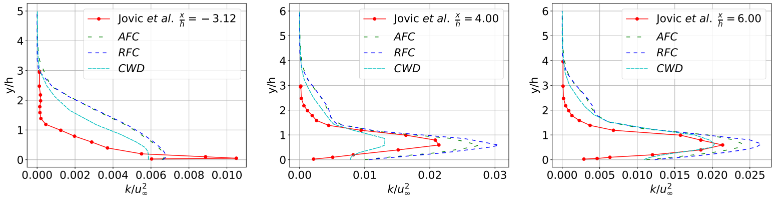
Figure 2.13 and figure 2.14 illustrate results obtained for 2D backward facing step using $k{-}\epsilon$ turbulence model for different stabilization methods. Here for comparison as well, wall laws for ϵ use k-based u_τ calculation (refer to equation (2.21a)). Inlet velocity is taken as $u_\infty = 7.72\,ms^{-1}$.

It is apparent from figure 2.13(b) that CWD method performs better in the region of recirculation. All three methods perform in a relatively indifferent manner in other regions. It can be seen from figure 2.14(c) that the CWD stabilization method performs better than others. The CWD method is based on adding diffusion in stream line direction and crosswind direction in addition to SUPG stabilization. Despite its better performance, its resulting global system matrix has large additions to off-diagonal terms, thus increasing the condition number. Therefore, in order to obtain a solution for the CWD method, direct solvers are used, because iterative solvers have



(a) $x = -3.12h$ (b) $x = 4h$ (c) $x = 6h$

Figure 2.13: Backward facing step 2D stabilization method velocity comparison.



(a) $x = -3.12h$ (b) $x = 4h$ (c) $x = 6h$

Figure 2.14: Backward facing step 2D stabilization method turbulent kinetic energy comparison.

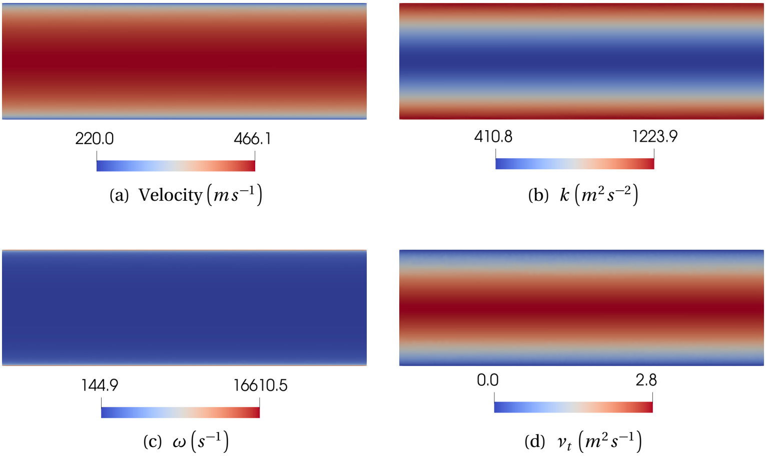
challenges in converging with high matrix condition numbers. But in the case of AFC and RFC, the condition numbers of their system matrices are comparatively better due to the M-Matrix property as explained for AFC in work done by Jha et al. [70], and therefore we are able to use iterative solvers to obtain the solution. This is important since our study aims at obtaining solution fields for 3D complex problems where direct solvers are infeasible. Therefore, the rest of the discussion is mainly focused on AFC and RFC methods.

2.3.4 Applicability of stabilization methods to different turbulence models

This section provides a comparison of the $k-\epsilon$, $k-\omega$, and $k-\omega-sst$ turbulence models' solutions for channel flow and backward facing step implemented with AFC and RFC stabilization methods. It focuses on identifying applicability of AFC and the novel RFC stabilization methods for different CDR transport equations.

Channel flow results

Figure 2.15 illustrates velocity, k, ω, and ν_t solution distributions for channel flow 2D problem with $Re_\tau = 2000$. This illustrated representative solution is obtained using the $k-\omega-sst$ turbulence model with the novel RFC stabilization method, where CDR transport equation Neumann wall conditions are based on equation (2.21b).



(a) Velocity $\left(ms^{-1}\right)$

(b) $k\left(m^2 s^{-2}\right)$

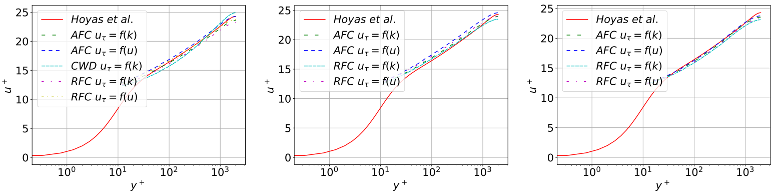
(c) $\omega\left(s^{-1}\right)$

(d) $\nu_t\left(m^2 s^{-1}\right)$

Figure 2.15: Channel flow 2D solution distributions.

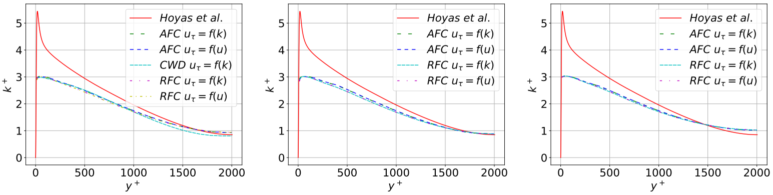
Figure 2.16, figure 2.17, and figure 2.18 provide u^+, k^+ and τ^+ variations with respect to y^+ for channel flow 2D problem with $Re_\tau = 2000$. From these figures, we can observe that $k-\omega-sst$ model has better agreement with DNS results, followed by $k-\epsilon$ and then $k-\omega$ turbulence models.

A likely explanation is that the presented case of $Re_\tau = 2000$ falls under high Re flows while $k-\omega-sst$ turbulence model is well known for its ability to model high Re flows. This is followed by the $k-\epsilon$ turbulence model.



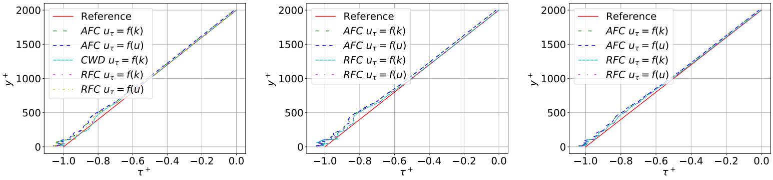
(a) $k-\epsilon$ (b) $k-\omega$ (c) $k-\omega-sst$

Figure 2.16: Channel flow 2D u^+ vs y^+ comparison for different turbulence models.



(a) $k-\epsilon$ (b) $k-\omega$ (c) $k-\omega-sst$

Figure 2.17: Channel flow 2D k^+ vs y^+ comparison for different turbulence models.



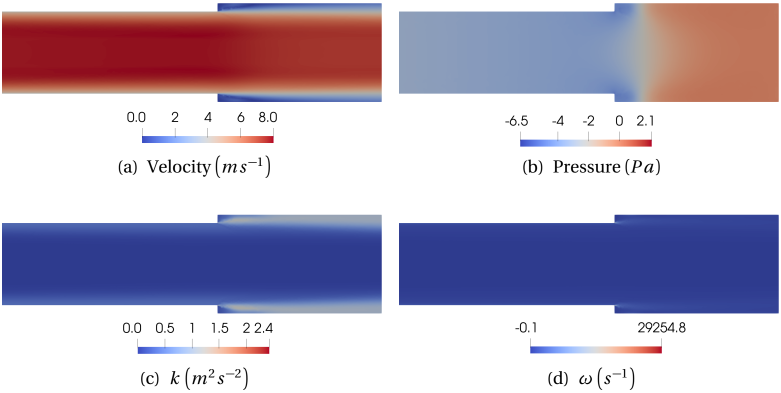
(a) $k-\epsilon$ (b) $k-\omega$ (c) $k-\omega-sst$

Figure 2.18: Channel flow 2D y^+ vs τ^+ comparison for different turbulence models.

As shown in figure 2.16, figure 2.17, and figure 2.18, there is no significant difference in solutions between AFC and RFC stabilization methods for each turbulence model. This indicates that the novel RFC stabilization method performs similarly to the AFC method for different CDR transport equations, which reflects its robustness in achieving physically meaningful solutions. The comparison of solutions for different u_τ calculations can also be done with same figures. They also show no significant difference between them. In addition, we have observed that 2D channel flow benchmark cases with Re_τ of 590, and 950 also depict similar behavior.

Backward facing step results

Figure 2.19 illustrates velocity, pressure, k, and ω solution distributions for the 2D backward facing step problem. This illustrated representative solution is obtained using the $k-\omega-sst$ turbulence model with the novel RFC stabilization method, where CDR transport equation Neumann wall conditions are based on equation (2.21b).



(a) Velocity $\left(ms^{-1}\right)$

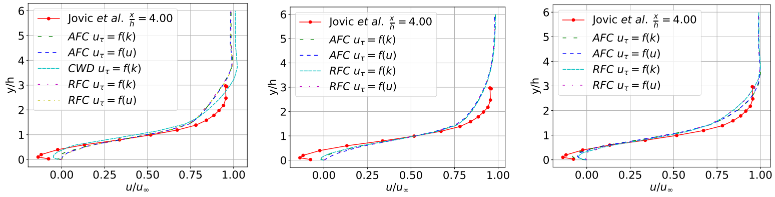
(b) Pressure (Pa)

(c) $k\left(m^2s^{-2}\right)$

(d) $\omega\left(s^{-1}\right)$

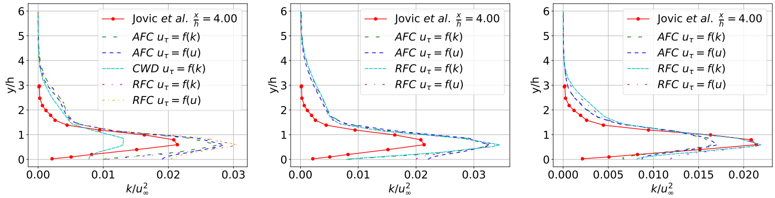
Figure 2.19: Backward facing step 2D solution distributions.

Figure 2.20 and figure 2.21 illustrate non-dimensionalized $\mathbf{u}$ and k fluctuations along wall normal direction at $x = 4h$ distance for different turbulence models, stabilization methods, and u_τ calculation methods used in 2D backward facing step benchmark.



(a) $k-\epsilon$ (b) $k-\omega$ (c) $k-\omega-sst$

Figure 2.20: Backward facing 2D step velocity solution comparison with different turbulence models.



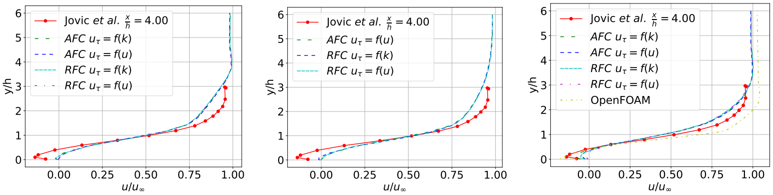
(a) $k-\epsilon$ (b) $k-\omega$ (c) $k-\omega-sst$

Figure 2.21: Backward facing 2D step turbulent kinetic energy solution comparison with different turbulence models.

We can observe from the results that for $k-\omega$ and $k-\omega-sst$, turbulence models do not show significant difference in the solution with respect to change in the u_τ calculation method. But in the case of $k-\epsilon$, we can see slight deviations in solutions depending on the u_τ calculation method, closest to DNS results are presented with the k-based u_τ calculation method. It is also evident that the novel RFC stabilization method is showing superior solution quality than the AFC method in the $k-\omega-sst$ turbulence model. This indicates that the novel RFC method is capable of adding less artificial diffusion than the AFC method to achieve a positivity preserving solution.

2.3.5 Coupling with different flow solving methods

So far, we have presented results obtained using monolithic method to solve $\underline{u}$ and p. As discussed in section 2.1, the monolithic method produces a system matrix which is large, sparse, and highly ill-conditioned when used to solve large problems (specifically 3D complex problems) where the number of degrees of freedom is large. Direct solvers are infeasible for large sparse systems, therefore iterative solvers need to be used. However, it is challenging to use iterative solvers to solve a highly ill-conditioned sparse system of equations without having additional support from pre-conditioners. Therefore we considered investigating the fractional step method as $\underline{u}$ and p solver in RANS for some of the benchmarks described in section 2.3.2. We also take the opportunity to compare results of turbulence models with different flow solver implementation found in OpenFOAM (Weller et al. [141]) to understand performance of the novel RFC stabilization method. FS approach is used here to compare results with OpenFOAM because, it also uses segregated manner to solve for variables $\underline{u}$ and p.

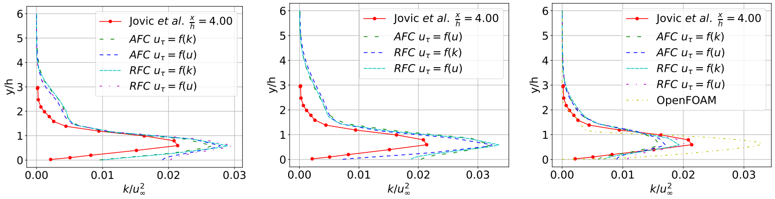


(a) $k-\epsilon$ (b) $k-\omega$ (c) $k-\omega-sst$

Figure 2.22: Backward facing step 2D fractional step **u** variation turbulence model comparison.

In the case of fractional step, $\underline{u}$ and p is solved in a segregated manner, and thus it does not suffer from the difficulties arising from the monolithic method. We have chosen the benchmark explained in section 2.3.2 to illustrate differences and similarities of solutions obtained using the fractional step method with the AFC and RFC stabilization methods.

Figure 2.22 and figure 2.23 show non-dimensionalized $\underline{u}$ and k variation in backward facing step using the fractional step method. We observe



(a) $k-\epsilon$ (b) $k-\omega$ (c) $k-\omega-sst$

Figure 2.23: Backward facing step 2D fractional step k variation turbulence model comparison.

no significant differences in velocity distribution in different turbulence models, stabilization, and u_τ calculation methods. This FS approach introduces coupling between $\underline{u}$ and p. We have observed that the AFC method with u_τ based on k failed to converge with the FS method. But, the novel RFC method was able to converge to an acceptable solution even with the FS method. This indicates that the novel RFC stabilization method is capable of coupling with different flow solution methods, thereby further highlighting its robustness. We can observe from figure 2.23 that there are differences in the solution of k depending on stabilization, turbulence model and u_τ calculation method. $k-\epsilon$ and $k-\omega$ show indifferent results for k. But in $k-\omega-sst$, the novel RFC stabilization method shows better agreement with experimental data from the benchmark case.

Figure 2.22(c) and figure 2.23(c) compares solutions achieved for $k-\omega-sst$ turbulence model through different stabilization methods implemented in "Kratos Multiphysics" (Dadvand et al. [37]Dadvand et al. [36]) and OpenFOAM (Weller et al. [141]). We can observe that, velocity distribution depicted in figure 2.22(c) has better agreement near the walls in the solution obtained from OpenFOAM, whereas solution from "Kratos Multiphysics" deviates slightly. "Kratos Multiphysics" has the peak near the wall under estimated due to the added diffusion. As depicted in figure 2.23(c), OpenFOAM over estimates the near wall peak in k distribution, whereas "Kratos Multiphysics" stabilization methods are able to closely estimate it. Even though, a similar mesh is used in both "Kratos Multiphysics" and OpenFOAM numerical experiments. They differ in used element technologies (i.e. "Kratos Multiphysics" is FEM based, OpenFOAM is FVM based). Due to this difference, there exists differences in how wall functions are applied, and how turbulent quantities are being modelled which have contributed towards differences in solutions obtained.

2.3.6 3D Pipe flow

A 3D experiment is also evaluated under different stabilization methods to understand the robustness of the novel stabilization method. We have chosen a turbulent pipe flow experiment with $Re = 100000$ case from the study done by Perry et al. [108] to compare results of the different methods.

Problem definition

The numerical domain and boundaries are illustrated in figure 2.24.

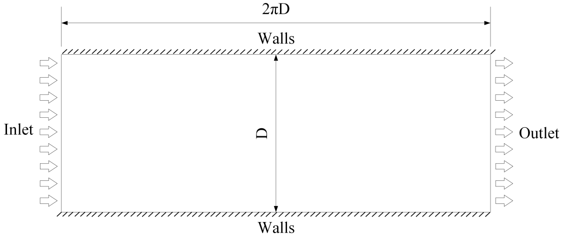


Figure 2.24: Pipe flow boundary conditions.

The diameter of this pipe is $D = 0.099\, m$ and its length is $2\pi D$. The boundary conditions are illustrated in table 2.2. Walls are assumed to be smooth (i.e. $\beta = 5.2$). A source term (mimicking pressure gradient $\frac{dp}{dx}$) derived from respective $Re_\tau = 2080$ is used to drive fluid flow within the pipe. Figure 2.25 presents unstructured tetrahedral spatial discretization (approx. 4 million elements) used for this investigation, which is again chosen following a mesh convergence study.

Results

Until now, we have discussed results from stabilization methods used to solve 2D benchmark cases. Now let's turn to a 3D benchmark case whose

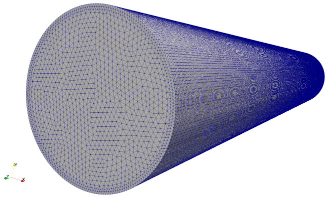
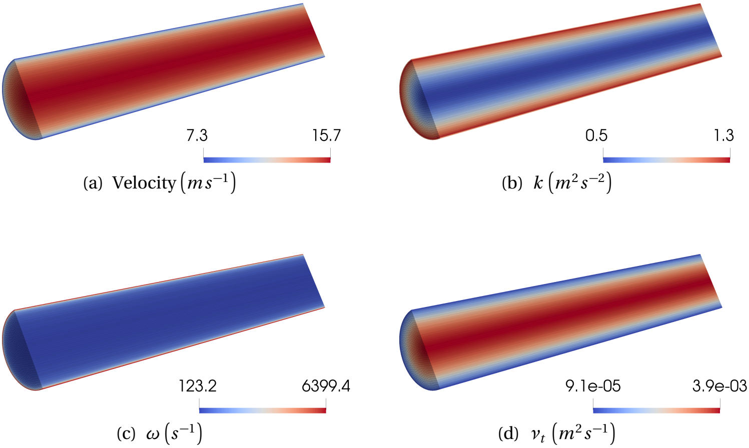


Figure 2.25: Pipe flow spatial discretization.

numerical setup is explained in section 2.3.6. The numerical investigation on this is carried out across all three different turbulence models under two different stabilization methods; namely AFC and RFC as discussed in previous sections. Figure 2.26 illustrates velocity as well as, k, ω and ν_t distributions in the 3D pipe domain, which is solved with the novel RFC method using $u_\tau = f(u)$ for the $k-\omega-sst$ turbulence model as a representative illustration.



(a) Velocity $\left(m s^{-1}\right)$

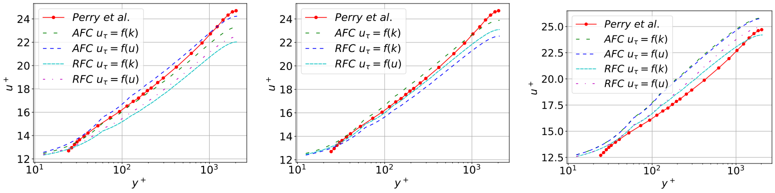
(b) $k\left(m^2 s^{-2}\right)$

(c) $\omega\left(s^{-1}\right)$

(d) $\nu_t\left(m^2 s^{-1}\right)$

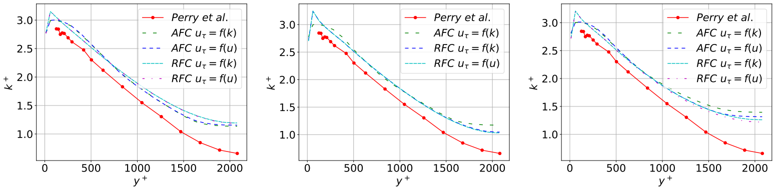
Figure 2.26: Pipe flow 3D solution distributions.

The non-dimensional velocity distributions for different turbulence models are presented in figure 2.27; non-dimensional turbulent kinetic energy distribution is illustrated in figure 2.28. It is apparent from the velocity distribution that all three turbulence models show only slight deviations from the experimental results. Figure 2.28 presents the k^+ variation, which shows similar results for RFC stabilization method compared to the AFC method. The $k-\omega$ model with AFC stabilization using wall functions based on $\underline{u}$ (i.e. $u_\tau = f(u)$) failed to converge. The RFC counterparts for the same turbulence model were successful in converging to acceptable solutions, indicating its capability for solving different CDR equations used in FEM. All of these results indicate that, the novel RFC stabilization method produces acceptable results for even 3D large domains.



(a) $k-\epsilon$ (b) $k-\omega$ (c) $k-\omega-sst$

Figure 2.27: Pipe flow 3D u^+ vs y^+ comparison for different turbulence models.



(a) $k-\epsilon$ (b) $k-\omega$ (c) $k-\omega-sst$

Figure 2.28: Pipe flow 3D k^+ vs y^+ comparison for different turbulence models.

2.4 URANS in Fluid-Structure Interaction Mok Benchmark Problem

Until now we have discussed applicability of the novel RFC stabilization method to different flow solver strategies and different turbulence models in the context of RANS. This section focuses on investigating solution behavior in more complex FSI problems in the context of URANS, thus emphasizing the novel RFC method's capability in obtaining acceptable solutions in Arbitrary Lagrangian-Eulerian (ALE) framework.

2.4.1 Mok benchmark problem definition

Figure 2.29 illustrates configuration used in solving the problem following the work of Mok [97].

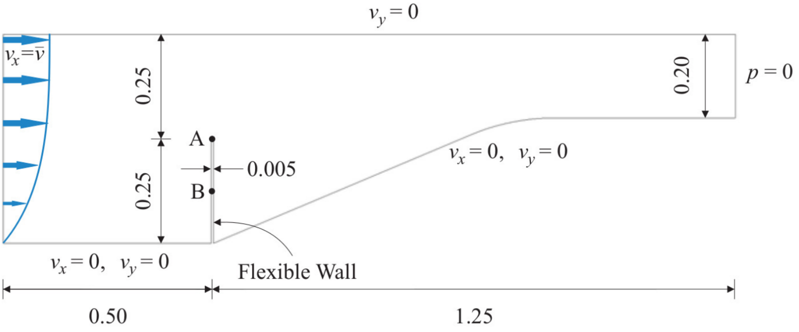


Figure 2.29: Mok benchmark case configuration (Zorrilla [151]).

FSI mok problem is considered to be a low Re problem. Therefore, the solutions from implemented $k-\omega$ and $k-\omega-sst$ turbulence models are investigated because these turbulence models are known to present good results in the cases of low Re problems. On the other hand, the implemented variation of $k-\epsilon$ turbulence model is not suitable for low Re problems, therefore results from it are omitted. Spatial discretization used in this problem is illustrated in figure 2.30 which is chosen after a mesh convergence study. Fluid and structural domain properties are illustrated in table 2.4.

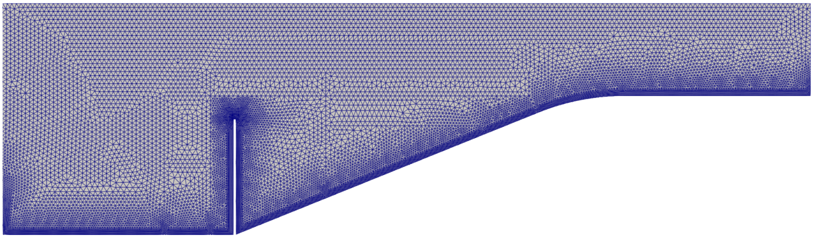


Figure 2.30: Mok benchmark mesh.

Table 2.4: Fluid and structural domain properties used in FSI Mok problem.

Domain	Property	Value
Fluid	Density	$956.0\, kg\, m^{-3}$
	Kinematic viscosity	$1.5 \times 10^{-4}\, m^2 s^{-1}$
Solid	Density	$1500.0\, kg\, m^{-3}$
	Young's modulus	$2.3\, MPa$
	Poisson ratio	0.45

Inlet is specified with a fixed parabolic inlet velocity with a ramp up of $t \in [0\, s, 10\, s]$ as depicted in equation (2.29). The walls are applied with no slip boundary condition for $u = 0\, m s^{-1}$, $k = 1 \times 10^{-12}$. The top boundary is applied with $u_y = 0$ making it a slip boundary condition. Pressure is applied with $P = 0$ at the outlet.

$$u = \begin{cases} 0.1214\left[1 - \cos\left(\frac{\pi t}{10}\right)\right] y\left(1 - y\right) & if \qquad t < 10.0 \\ 0.2428 y\left(1 - y\right) & else \end{cases} \tag{2.29}$$

ω in $k - \omega$ and $k - \omega - sst$ turbulence models tends to infinity at walls. Therefore, special care has to be taken to correctly identify feasible ω values for wall if no slip is used for u without any wall modelling as in (ANSYS [6]). One other concern in this problem is, there is a common spatial location (lower left corner) which belongs to inlet and wall. The classical approach (refer equation (2.30b)) of calculating ω creates singularities near that corner because the adjacent wall spatial point will have higher ω due to its nature. In order to avoid that, a special blended ω formulation is used as depicted in equation (2.30c). In there, H represents height of the inlet, L_{turb} is the turbulent mixing length, C_μ is a constant, y is the wall distance, and ω_{wall} is the nodes' respective wall ω computed via wall ω estimation method as described in ANSYS [6].

$$f = e^{-\frac{20y}{H}} \tag{2.30a}$$

$$\omega_{classical} = \frac{\sqrt{k}}{C_\mu^{\frac{1}{4}} L_{turb}} \tag{2.30b}$$

$$\omega_{blended} = f\,\omega_{wall} + \left(1 - f\right)\omega_{classical} \tag{2.30c}$$

Strong coupling is used between FSI coupling iterations as depicted in figure 2.31. First flow variables u and p are solved with an initial ν_t until their governing equations' residual convergence. Then, k and ϵ are solved until their governing equations' residual converges. Afterwards ν_t is updated. The process of solving k and ϵ is repeated until their residual convergence and ν_t convergence up to the specified tolerance. Then, again the governing equations' residuals of u and p are checked using the updated ν_t. If the residuals' convergence criteria are satisfied, then the forces on the structural surface are calculated. This is followed by solving for displacements using the structural solvers with the calculated forces acting on the structural surfaces. Then the mesh is updated using Laplacian mesh motion solvers for new displacements. This is continued until displacements are converged as explained in figure 2.31. Once the displacement field is converged, then time is incremented by Δt to obtain the solution for the next time step.

2.4.2 Results

Tip displacement (at point ''A'' in figure 2.29) variation with time for different turbulence models is illustrated in figure 2.32. Reference values are taken from work done by Mok [97] and Vázquez [131] which are shown in the same figure.

As seen in figure 2.32, $k-\omega$ and $k-\omega-sst$ turbulence models show only a slight deviation in the displacement than the reference values (i.e. reference from Valdes) at the time instance where maximum displacement

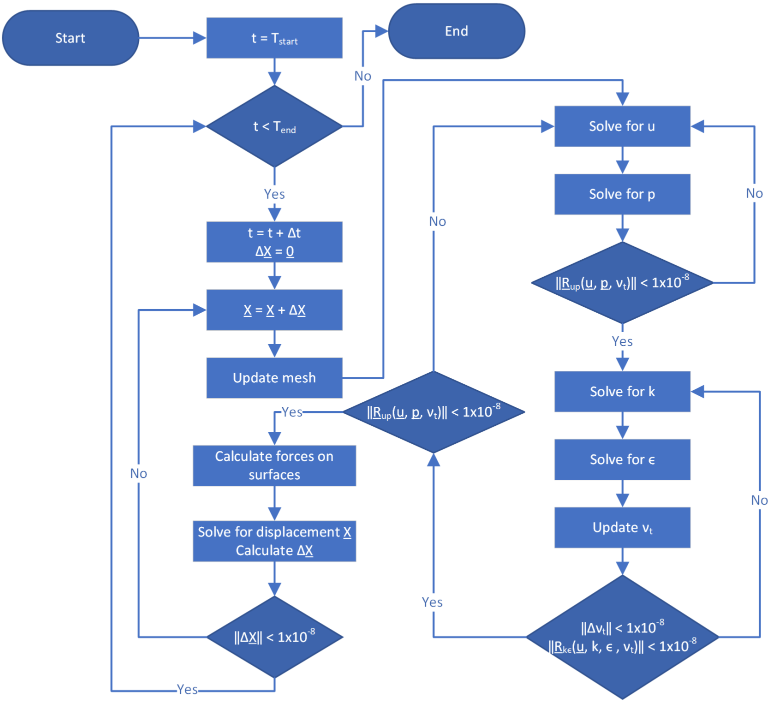


Figure 2.31: FSI flow solver implementation with URANS for $k-\epsilon$ turbulence model.

is observed (i.e. $t = 9.0\ s$). Even though there is a slight deviation in solution at maximum displacement point, solutions from both turbulence models illustrate almost the same displacement.

It is observed that $y^+ \leq 1$ near the flexible wall, causing k to be relatively low (refer figure 2.33(d)), and ω being relatively high (refer figure 2.33(e)). Consequently, ν_t distribution is also relatively low near the flexible wall (refer figure 2.33(f)). However, near the upper left corner of the inlet shows relatively higher ν_t distribution. This is because, the highest k is found at the top most location in the inlet boundary, and it is the furthest point from wall making ω distribution relatively small consequently making the ν_t relatively large.

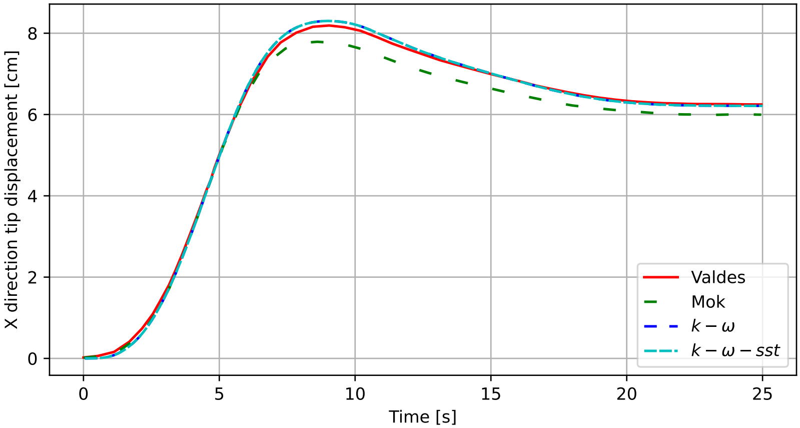
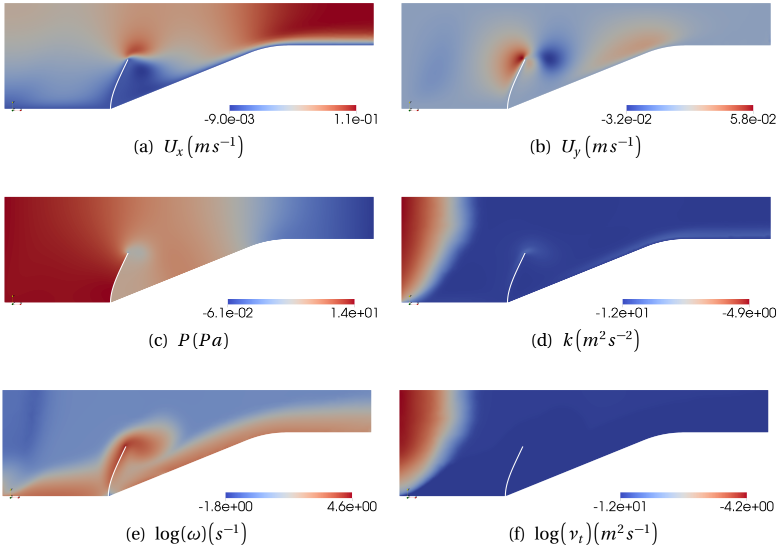


Figure 2.32: Tip displacement comparison for different turbulence models.



(a) $U_x\left(ms^{-1}\right)$

(b) $U_y\left(ms^{-1}\right)$

(c) $P\,(Pa)$

(d) $k\left(m^2s^{-2}\right)$

(e) $\log(\omega)\left(s^{-1}\right)$

(f) $\log\left(\nu_t\right)\left(m^2s^{-1}\right)$

Figure 2.33: FSI mok solution fields at $t = 9.0\,s$.

2.5 RANS/URANS in Wind Engineering Problems

As pointed out in chapter 1, RANS and URANS can be used to reduce the time consumption in overall numerical wind tunnel tests. Therefore, this section investigates the applicability of RANS and URANS using the novel RFC stabilization method to solve Benchmark on the Aerodynamics of a Rectangular 5:1 Cylinder (BARC) problem. Despite the simple geometry in BARC, it is well suited for this study because it provides insights to turbulence characteristics, flow separation and bluff body aerodynamics which are present in more complex and practical problems in CWE (Bruno et al. [22]).

2.5.1 BARC problem definition

Figure 2.34 illustrates size of the fluid domain and the rectangular cylinder geometry with $H = 0.05\,m$ which is used in BARC problem. The inlet is specified with $15.26\,ms^{-1}$. The top and bottom is applied with slip wall ($u_y = 0$). The outlet is applied with $P = 0$. Fluid density (ρ) is $1.2\,kg\,m^{-3}$ and fluid dynamic viscosity ($\mu_{molecular}$) is $1.83 \times 10^{-5}\,Pa\,s$ which corresponds to a flow having $Re = 5 \times 10^4$ around the rectangular cylinder. The walls are applied with automatic wall functions which are able to approximate wall effects using either logarithmic or linear wall laws. ω near the walls are computed with special care since it tends to infinity at walls (refer section 2.4.1). Different rectangular cylinder inclination angles measured in counter clockwise direction (i.e. θ_{IA}) are also considered in this study to understand the flow representations under different Angle of Attack (AoA). The turbulence models presented here are $k - \epsilon$ and $k - \omega - sst$ which are evaluated using a 2D domain. Afterwards their solutions are compared against the reference solution (internal study carried out by Pentek, et al. using LES) to understand limitations and capabilities of RANS and URANS with aforementioned turbulence models.

2.5.2 Flow field distributions

Table 2.5 and table 2.6 illustrate solution flow field distributions for $\theta_{IA} = 0\,deg$ and $\theta_{IA} = 10\,deg$ problems respectively. The columns represent the flow field variables and the rows represent different turbulence models

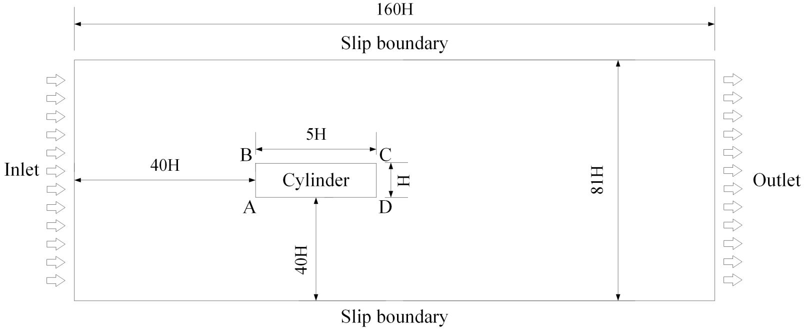


Figure 2.34: BARC configuration.

and the reference solution. All the distributions shown in two tables are time averaged results.

Table 2.5: Time averaged quantities comparison between different turbulence models for $\theta_{IA} = 0\,deg$.

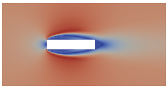
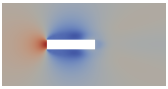
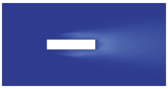
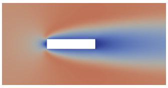
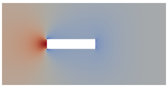
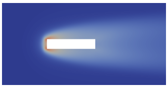
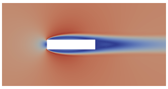
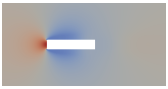
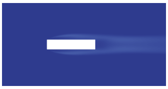
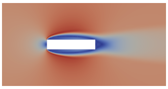
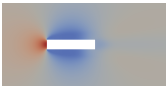
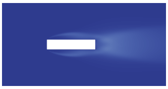
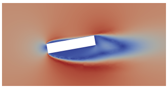
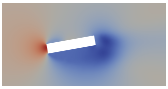
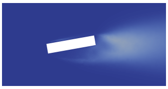
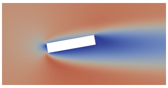
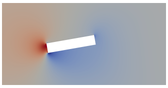
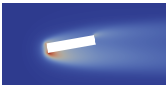
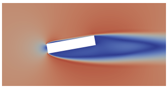
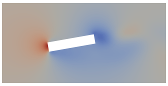
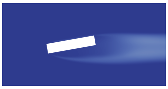
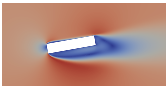
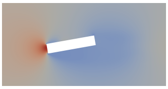
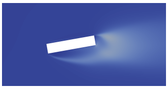
	$\lvert\underline{u}\rvert[m s^{-1}]$ $\lvert\underline{u}\rvert \in [0, 22]$	$p\,[Pa]$ $p \in [-160, 160]$	$k\,[m^2 s^{-2}]]$ $k \in [0, 120]$
LES Ref.			
RANS $k-\epsilon$			
RANS $k-\omega$ sst			
URANS $k-\omega$ sst			
	Min ▬ Max		

Table 2.6: Time averaged quantities comparison between different turbulence models for $\theta_{IA} = 10\,deg$.

	$\lvert\underline{u}\rvert\,[ms^{-1}]$ $\lvert\underline{u}\rvert \in [0, 22]$	$p\,[Pa]$ $p \in [-160, 160]$	$k\,[m^2 s^{-2}]]$ $k \in [0, 120]$
LES Ref.			
RANS $k-\epsilon$			
RANS $k-\omega$ sst			
URANS $k-\omega$ sst			
	Min ▬ Max		

The frontal regions (AB section in figure 2.34) in both θ_{IA} angles depict solutions in agreement with the reference solution. In the case of $\theta_{IA} = 0\,deg$, RANS $k-\epsilon$ turbulence model does not show flow re-attachment, whereas all the other models clearly identify it. Among those, URANS depicts the most accurate flow re-attachment compared to its steady counterpart. The peak velocity depicted by RANS $k-\epsilon$ is lower than other turbulence models. p distribution is also showing a similar variation between RANS $k-\omega-sst$ and URANS $k-\omega-sst$ turbulence models. However, only URANS $k-\omega-sst$ turbulence model illustrates a good agreement with the reference solution whereas all RANS turbulence models show poor agreements. This is mainly because of the RANS turbulence models' inability to accurately predict turbulent mixing of the flow field. In the case of $\theta_{IA} = 10\,deg$, it is also evident that the URANS $k-\omega-sst$ turbulence model shows the best agreement with the reference solution among all the turbulence models being investigated. The RANS $k-\omega-sst$ turbulence model is showing the next best results followed by solution of the RANS $k-\epsilon$.

Even though, the RANS $k-\epsilon$ solution shows poor agreement with the reference solution due to its inability to predict the turbulent mixing of the flow, it is used as the initial guess for the solution fields in the cases of RANS $k-\omega-sst$. This is because, the RANS $k-\omega-sst$ shows poor convergence rates if it is initialized with the same flow field distributions as in the RANS $k-\epsilon$ which is initialized with constants.

2.5.3 Drag and lift coefficient variation

Time averaged drag and lift coefficients are evaluated for different θ_{IA} and different turbulence models in this section. The $k-\omega$ turbulence model produced lift and drag coefficients with significantly higher errors, hence they are omitted from the results. This is likely because, the implemented $k-\omega$ turbulence model does not have the ability to obtain solutions for high Re flow problems. The drag and lift coefficients shown in figure 2.35 depict that the RANS $k-\epsilon$ turbulence model overestimates drag coefficient whereas, the RANS $k-\omega-sst$ turbulence model under estimates it. Error shown by the RANS $k-\epsilon$ turbulence model is relatively higher than the error from $k-\omega-sst$. One likely reason for this is, $k-\omega-sst$ turbulence model is known to perform better with recirculation, flow separation as opposed to the RANS $k-\epsilon$. Among all the turbulence models, URANS $k-\omega-sst$ turbulence model shows the results which has the best agreement with the reference solution.

2.5.4 Pressure variation

In order to better understand the C_D and C_L distributions obtained by different turbulence models, this section compares their pressure distribution along the rectangular cylinder surface. Figure 2.36 illustrates the time averaged pressure variation along the center line of the rectangular cylinder moving from points in the order "A", "B", "C", "D", "A" (refer figure 2.34). All of the turbulence models show higher accuracy in predicting time averaged pressure field in the frontal area (i.e. AB face) which is also made evident from the flow field variations depicted in table 2.5 and table 2.6. However, prediction of p in the RANS $k-\epsilon$ in all faces shows weak agreement with the reference solution than the predictions from other turbulence models. The RANS $k-\omega-sst$ and URANS $k-\omega-sst$

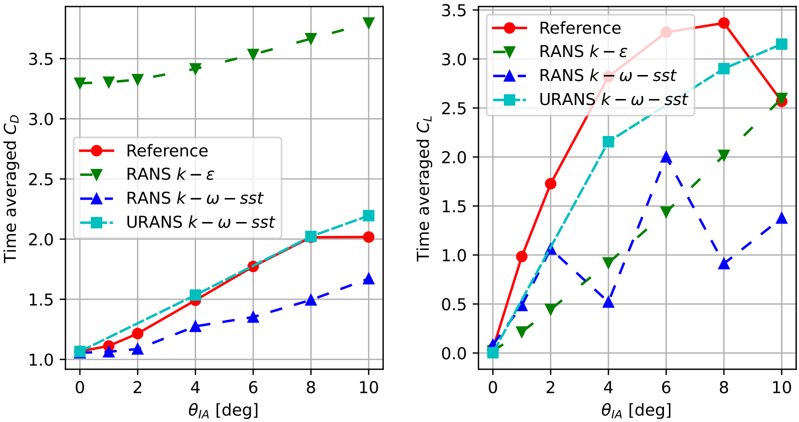


Figure 2.35: Time averaged drag (left) and lift (right) coefficient variation in BARC for different inclination angles.

turbulence models show good approximation of the pressure along the line on BC face as well, whereas RANS $k-\epsilon$ overshoots.

In the case of $\theta_{IA} = 0\,deg$, RANS $k-\omega-sst$ and URANS $k-\omega-sst$ turbulence models show a good agreement with the reference solution in faces CD and DA as well. When θ_{IA} is increased, vortex mixing can be observed in the flow. This result in a complex flow field, for which RANS $k-\omega-sst$ turbulence model shows poor p approximation on the faces CD and DA. This indicates that RANS $k-\omega-sst$ is not suitable for predicting flow fields where there are significant vortex mixing. Whilst RANS turbulence models depict poor predictive capabilities in the CD and DA surfaces, URANS $k-\omega-sst$ turbulence model shows a good agreement with the reference solution.

2.5.5 Velocity variation

Understanding velocity (i.e. $\underline{u}$) is also crucial to understand how well these turbulence models behave in complex CWE problems. Therefore velocity distributions along lines perpendicular to the rectangular cylinder are investigated. Figure 2.37 shows velocity distributions along line L1, L2, L3

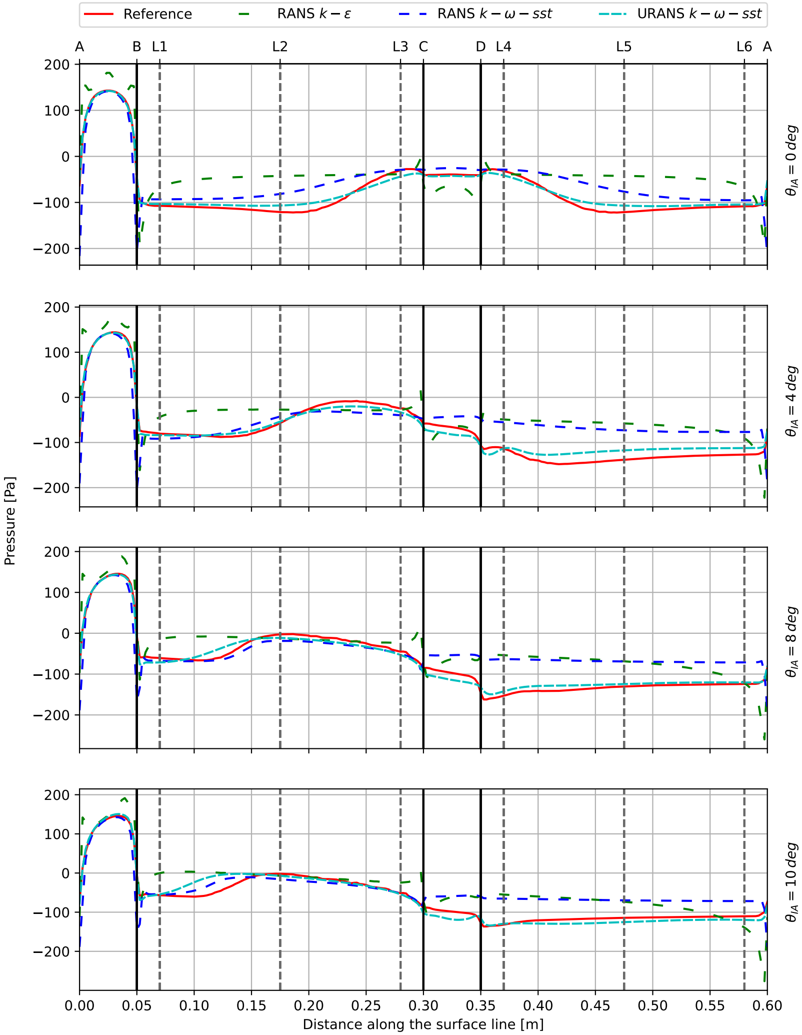


Figure 2.36: Pressure variation along the surface of BARC for different inclination angles.

(at 0.07 m, 0.175 m, and 0.28 m distances from point "A" along the center line of surface). In here also, RANS $k-\epsilon$ shows poor agreement in velocity distribution near walls which may be one of the reasons why it does not show good approximation of p and hence the C_D and C_L variations. RANS $k-\omega-sst$ and URANS $k-\omega-sst$ turbulence models show better near wall effect prediction, among them URANS $k-\omega-sst$ being the best.

Figure 2.38 shows velocity distributions along surface perpendicular lines positioned at L4, L5, L6 (at 0.37 m, 0.475 m, and 0.58 m distances from point "A" along the center line of surface). This represents velocity distributions near the lower surface of the rectangular cylinder. As explained in the section 2.5.4, vortex mixing near the lower surface makes it difficult to predict velocity field accurately using RANS models. Even URANS $k-\omega-sst$ turbulence model shows slight deviations from the reference solution when $\theta_{IA} \geq 6\,deg$, which is one of the main reasons for discrepancies shown in figure 2.35. However, overall most accurate velocity field distribution is predicted by the URANS $k-\omega-sst$ turbulence model.

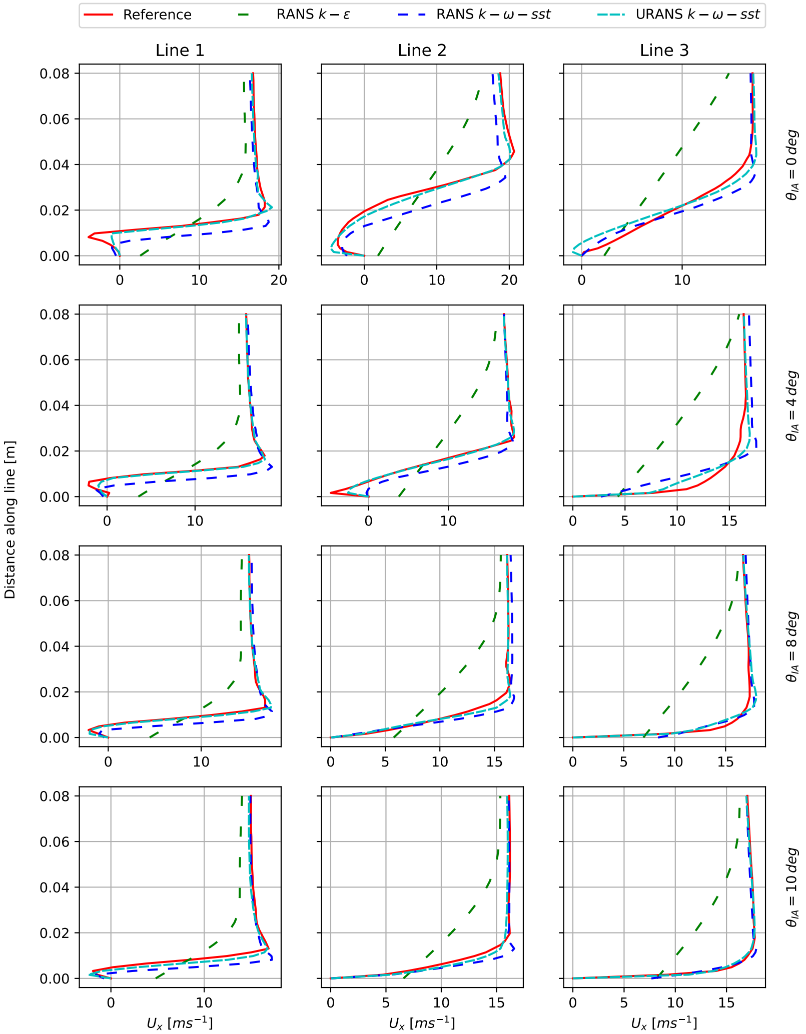


Figure 2.37: Time averaged streamwise velocity variation along lines in BARC deck top surface for different inclination angles.

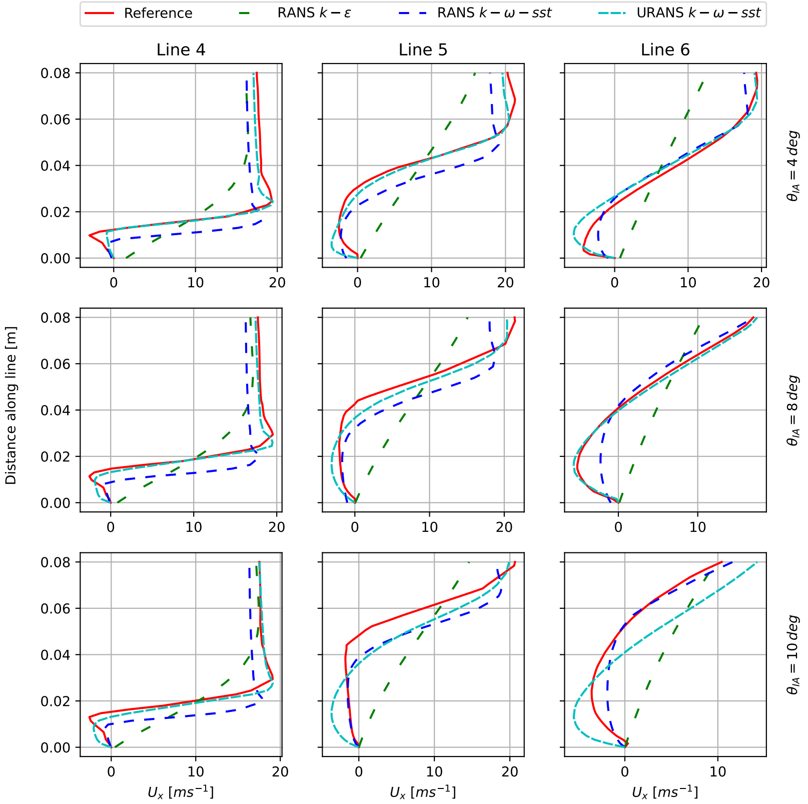


Figure 2.38: Time averaged streamwise velocity variation along lines in BARC deck bottom surface for different inclination angles.

2.5.6 Frequency variation

Drag and lift forces' frequency distribution for different θ_{IA} angles are illustrated in figure 2.39 for URANS $k-\omega-sst$ turbulence model. Drag force frequency shows small variations in the force amplitudes compared to that of lift force variations. This is because, vortex shedding occurs in lift direction in the BARC problem. This makes lift forces to have more prominent oscillations than in the drag force direction. Hence this discussion focuses mainly on lift force frequency distribution here onwards. There is only one prominent lift force frequency shown for problems which have $\theta_{IA} \leq 4$. The lift force frequency predicted by URANS $k-\omega-sst$ is lower than the reference solution, reducing the error as θ_{IA} increases. There are two major frequencies present in the problems with $\theta_{IA} > 4$ as depicted in figure 2.39. One reason for this is, when θ_{IA} is increased, it produces vortex mixing which becomes more significant for lift force predictions. URANS $k-\omega-sst$ turbulence model is capable of predicting two major lift force frequencies, where highest is lower than the reference, and lowest is higher than the reference. This makes URANS $k-\omega-sst$ turbulence model also less accurate on predicting frequency distribution even though it is good at predicting time averaged p, $\underline{u}$, C_D, and C_L variations for different θ_{IA} values.

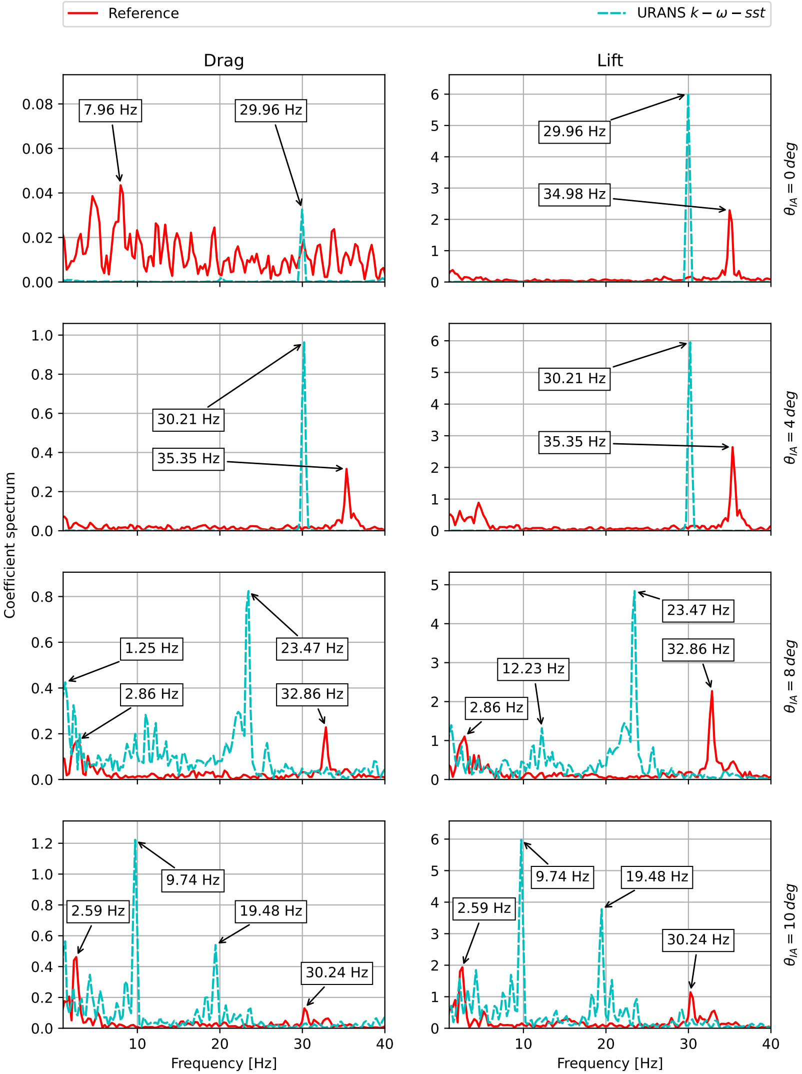


Figure 2.39: Drag and lift coefficient frequency distribution in BARC deck for different inclination angles.

2.6 Conclusions and Outlook

A novel positivity preserving residual-based flux-corrected (RFC) stabilization method adding artificial diffusion for solving convection-diffusion-reaction (CDR) equation is presented for unstructured standard Galerkin linear finite element formulation. This novel RFC stabilization method is developed paying attention to its ability to obtain adjoint formulations for adjoint sensitivity analysis. Then, it is tested against 2D scalar transport problems having analytical answers. There, the novel RFC stabilization method produced low errors for continuous solutions and relatively higher errors for discontinuous solutions while preserving the positivity in both solution types. Afterwards, the novel RFC stabilization method is applied for different turbulent eddy-viscosity based turbulence modeling CDR transport equations to identify its robustness in solving RANS equations. $k-\epsilon$, $k-\omega$, and $k-\omega-sst$ are chosen as the respective turbulence models for investigation. The solutions from all stabilization methods (including RFC, Algebraic Flux Corrected (AFC), and Crosswind Diffusion (CWD) methods) and turbulence models are compared against their respective DNS or experiment results to evaluate physical meaningfulness of solutions and to identify the effect of added artificial diffusion on solution via stabilization. This study showed that the novel RFC stabilization method can perform better or at least the same as the existing AFC method when used in turbulence modeling for different flow fields. The CWD method outperformed the novel RFC stabilization method, but the CWD method proved infeasible to be used in large complex domains, where the novel RFC stabilization and AFC methods are able to obtain satisfactory solutions. Furthermore, the novel RFC stabilization and existing AFC methods are evaluated against two different velocity and pressure flow variable solvers: monolithic and fractional step. It was found that the novel RFC stabilization method performs similarly to the AFC method in all turbulence models based on the monolithic solver for flow variables. This method is also able to obtain solutions similar to AFC for all turbulence models in the fractional step method, indicating its flexibility in coupling with different flow solvers. The fractional step study is carried out because monolithic solvers become infeasible for problems with large complex domains. The novel RFC stabilization method introduced two different controlling parameters which were not found to have significant influence over the final converged solution for a given range of values, but they have influence over the rate of convergence.

Furthermore, the novel RFC stabilization method is used to solve a FSI problem with turbulence models URANS $k-\omega$ and $k-\omega-sst$. Both turbulence models produced almost similar solution fields. They predicted the peak displacements slightly higher than the reference solution. However, both turbulence models were able to predict displacements within an agreeable range except for the peak. Thereafter, BARC problem solutions obtained from RANS $k-\epsilon$, RANS $k-\omega-sst$, and URANS $k-\omega-sst$ using the novel RFC stabilization method are compared against a reference solution obtained from LES study. This included investigations of flow fields for different inclination angles of the rectangular cylinder. Afterwards, drag coefficient, lift coefficient, pressure along the surface, velocity near walls, and frequency distribution from RANS and URANS solutions are compared against the reference solution. URANS $k-\omega-sst$ produced best agreeable solution in drag coefficient, lift coefficient, pressure along the surface, velocity near walls whereas others depicted shortcomings and RANS $k-\epsilon$ showing the least agreeable solution. However, URANS $k-\omega-sst$ turbulence model also depicted shortcomings in predicting frequency distributions for different inclination angles.

CHAPTER 3

STEADY SENSITIVITY ANALYSIS

Sensitivity analysis is the study of how a response (i.e. goal) of a physical system behaves when it is subjected to infinitesimal perturbations in the input parameters (i.e. response function gradient w.r.t. input parameters). This can also be recognized as the uncertainty of a response in a physical system with respect to its input parameters. Sensitivity analysis is used in many fields of research such as optimization, uncertainty quantification, robustness estimation (Arriola et al. [8] and Gunzburger [55]). Therefore, this chapter focuses on developing extensible, modular, and efficient steady state sensitivity analysis for novel stabilization method introduced in chapter 2. Developed steady state sensitivity analysis methodology acts as a solid basis for further extensions required for development of transient sensitivity analysis and goal oriented AMR as explained in chapter 1.

Sensitivities for a given response function can be determined mainly using two methods namely "direct method" and "adjoint method". In the direct method, the sensitivity of given number of response functions (i.e. M) are computed by first computing a reference solution (i.e. primal solution) for a given input parameters which is followed by another solution computation with a perturbed input parameter as elaborated in Dickinson et al. [39]. This method requires evaluating primal solutions for each input parameter perturbation, which has a $O(N)$ cost complexity (where N being the number of input parameters). In CFD problems, it is of interest to compute sensitivities with respect to domain discretization coordinates (i.e. mesh nodal coordinates). Hence the number of input parameters for sensitivity analysis will be prohibitively large to use in "direct approach" since this requires $N+1$ primal solution evaluations.

The next method is the "adjoint approach". This method derives a Lagrange function from primal method's governing equations' residuals (i.e. $\underline{R}$ a function of mesh nodal coordinates $\underline{x}$ and state variables $\underline{w}$) as equality constraints to a mathematical optimization problem for a given response function (i.e. J) as depicted in equation (3.1). Thus, the Lagrange multipliers (i.e. $\underline{\tilde{w}}$) are computed by solving for extremum points of the Lagrange function (i.e. adjoint problem) followed by sensitivity computation.

$$L = J\left(\underline{x}, \underline{w}\right) + \underline{\tilde{w}}^T \underline{R}\left(\underline{x}, \underline{w}\right) \tag{3.1}$$

The sensitivities with respect to nodal coordinates can be computed using equation (3.2) where $\underline{\tilde{R}}$ is the residual of the adjoint problem.

$$\frac{dL}{d\underline{x}}^T = \frac{\partial J}{\partial \underline{x}}^T + \underline{\tilde{w}}^T \frac{\partial \underline{R}}{\partial \underline{x}} + \underbrace{\left[\frac{\partial J}{\partial \underline{w}}^T + \underline{\tilde{w}}^T \frac{\partial \underline{R}}{\partial \underline{w}}\right]}_{\underline{\tilde{R}}^T} \frac{d\underline{w}}{d\underline{x}} \tag{3.2}$$

The generalized formulations of Lagrange function (equation (3.1)) and its sensitivity derivatives (equation (3.2)) require the adjoint problem to be solved (i.e. $\underline{\tilde{R}} = \underline{0}$) as depicted in equation (3.3).

$$\underline{\tilde{w}}^T \frac{\partial \underline{R}}{\partial \underline{w}} = -\frac{\partial J}{\partial \underline{w}}^T \Longrightarrow \frac{dJ}{d\underline{x}}^T = \frac{dL}{d\underline{x}}^T = \frac{\partial J}{\partial \underline{x}}^T + \underline{\tilde{w}}^T \frac{\partial \underline{R}}{\partial \underline{x}} \tag{3.3}$$

In the case of two equation turbulence model RANS $k-\epsilon$ primal problems, the $\underline{R}$ has the momentum equation residuals (i.e. $\underline{R}_u, \underline{R}_v, \underline{R}_w$), continuity equation (i.e. $\underline{R}_p$), turbulent kinetic energy equation (i.e. $\underline{R}_k$), and turbulent dissipation rate equation (i.e. $\underline{R}_\epsilon$). The $\underline{w}$ represents $\left[u, v, w, p, k, \epsilon\right]^T$, and $\underline{\mathbf{R}}_{\phi,\omega} = \frac{\partial \underline{R}_\phi}{\partial \underline{\omega}}$.

$$\frac{\partial \underline{R}}{\partial \underline{w}} = \begin{bmatrix} \underline{\mathbf{R}}_{u,u} & \underline{\mathbf{R}}_{u,v} & \underline{\mathbf{R}}_{u,w} & \underline{\mathbf{R}}_{u,p} & \underline{\mathbf{R}}_{u,k} & \underline{\mathbf{R}}_{u,\epsilon} \\ \underline{\mathbf{R}}_{v,u} & \underline{\mathbf{R}}_{v,v} & \underline{\mathbf{R}}_{v,w} & \underline{\mathbf{R}}_{v,p} & \underline{\mathbf{R}}_{v,k} & \underline{\mathbf{R}}_{v,\epsilon} \\ \underline{\mathbf{R}}_{w,u} & \underline{\mathbf{R}}_{w,v} & \underline{\mathbf{R}}_{w,w} & \underline{\mathbf{R}}_{w,p} & \underline{\mathbf{R}}_{w,k} & \underline{\mathbf{R}}_{w,\epsilon} \\ \underline{\mathbf{R}}_{p,u} & \underline{\mathbf{R}}_{p,v} & \underline{\mathbf{R}}_{p,w} & \underline{\mathbf{R}}_{p,p} & \underline{\mathbf{R}}_{p,k} & \underline{\mathbf{R}}_{p,\epsilon} \\ \underline{\mathbf{R}}_{k,u} & \underline{\mathbf{R}}_{k,v} & \underline{\mathbf{R}}_{k,w} & \underline{\mathbf{R}}_{k,p} & \underline{\mathbf{R}}_{k,k} & \underline{\mathbf{R}}_{k,\epsilon} \\ \underline{\mathbf{R}}_{\epsilon,u} & \underline{\mathbf{R}}_{\epsilon,v} & \underline{\mathbf{R}}_{\epsilon,w} & \underline{\mathbf{R}}_{\epsilon,p} & \underline{\mathbf{R}}_{\epsilon,k} & \underline{\mathbf{R}}_{\epsilon,\epsilon} \end{bmatrix} \tag{3.4}$$

3.1 Adjoint Element Formulation Implementation

The adjoint local element matrix depicted in equation (3.4) has derivatives for each governing equation residual with respect to all solvable variables. Often times, it can be found from literature that derivatives of turbulence viscosity is neglected (i.e. "frozen turbulence": $\frac{\partial \nu_t}{\partial \phi} = 0$) (Anderson et al. [4], Jameson et al. [68], Othmer [105], and Papadimitriou et al. [106]). It makes computation of $\frac{\partial \underline{R}}{\partial \underline{w}}$ simplified because terms $\underline{\mathbf{R}}_{u,k}$, $\underline{\mathbf{R}}_{u,\epsilon}$, $\underline{\mathbf{R}}_{v,k}$, $\underline{\mathbf{R}}_{v,\epsilon}$, $\underline{\mathbf{R}}_{w,k}$, $\underline{\mathbf{R}}_{w,\epsilon}$, $\underline{\mathbf{R}}_{p,k}$, $\underline{\mathbf{R}}_{p,\epsilon}$ becomes zero, thereby not anymore requiring to calculate any of the derivatives of turbulence transport residuals (i.e. $\underline{R}_k, \underline{R}_\epsilon$). However, this simplification may cause deviations in calculated sensitivities (Zymaris et al. [152]). Therefore, in this study a methodology is developed to calculate total derivatives effectively and efficiently, while making it extensible for any turbulence model of interest. This is important because, if primal is solved with a different turbulence model (i.e. $k-\omega, k-\omega-sst$, or extending to support energy equation adjoints) and extensibility is not considered, then it may require implementing the local element matrix without the ability to reuse already implemented residual derivatives such as $\underline{\mathbf{R}}_{u,u}$ which does not depend on the chosen turbulence model. This may create opportunities for redundant code, inefficiencies, and errors. All of the local element matrices are computed at element level for each gauss point in adjoint solution, hence these calculations need to be efficient as possible. All of these points out that there is a need for a adjoint formulation implementation which is flexible/extensible and also efficient while keeping the redundancies to minimum for better maintainability. In the context of efficiency, there are few places to look at such as where value storage should take place when computing derivatives. Since "Kratos Multiphysics" is having a C++ backend, it allows us to use either "stack memory" or "heap memory". The stack memory is faster than heap memory because stack allocation is based on last-in-first-out manner (Ferres [43]). Therefore, where ever possible, the variables are stored in stack memory than heap memory in the adjoint formulation implementations.

3.1.1 Primal problem

In order to understand what is required for the adjoint formulation, first it is important to look at the primal problem. Two equation turbulence models in RANS consist of momentum equations as depicted in equation (3.5) where $\underline{u} = [u, v, w]$, continuity equation as depicted in equation (3.6), and two scalar transport equations as shown in equation (3.7) where ϕ can be either k, ϵ, or ω in respective $k-\epsilon$, $k-\omega$, or $k-\omega-sst$ turbulence models.

$$r_{u_i} = u_j \frac{\partial u_i}{\partial u_j} - \frac{\partial}{\partial x_j}\left(\left[\nu + \nu_t\right]\frac{\partial u_i}{\partial x_j}\right) + \frac{1}{\rho}\frac{\partial P}{\partial x_i} - g_i = 0 \tag{3.5}$$

$$r_p = \frac{\partial u_i}{\partial x_i} = 0 \tag{3.6}$$

r_u, r_v, r_w has ν_t term which depends on the turbulence model being used. This is computed via "Constitutive Laws" present in "Kratos Multiphysics" framework. For each turbulence model, different constitutive laws are implemented. The constitutive laws are made with an interface to compute derivatives of ν_t. Hence, the same constitutive law is used in adjoints to get derivatives such as $r_{u,k}, r_{v,k}, r_{w,k}$ which only have contributions from derivatives of ν_t.

$$r_\phi = u_{\phi,i} \frac{\partial \phi}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\nu_\phi \frac{\partial \phi}{\partial x_i}\right) + s_\phi \phi - f_\phi = 0 \tag{3.7}$$

Scalar transport equation depicted in equation (3.7) is used to compute scalar turbulence quantities used in RANS turbulence models. It consist of effective velocity (i.e. $u_{\phi,i}$), effective kinematic viscosity (i.e. ν_{ϕ}), reaction term (i.e. s_{ϕ}), and source term (i.e. f_{ϕ}). "ConvectionDiffusionReactionElementData" interface is developed with member variables "mEffectiveVelocity" to store $u_{\phi,i}$, "mEffectiveKinematicViscosity" to store ν_{ϕ}, "mReactionTerm" to store s_{ϕ}, and "mSourceTerm" to store f_{ϕ}. "CalculateGaussPointData" method calculates all of these variables for each gauss point once and stores in the derived class primal data container (such as "KElementData"). These variables are stored in the stack memory of the common primal data container which is then used to obtain different derivatives multiple times, and computing these coefficients on the fly would be redundant and time consuming.

3.1.2 Adjoint problem

Adjoint problem as depicted in equation (3.4) requires to compute derivatives of each residual w.r.t. each solving variable. Momentum equation residual w.r.t. u, v, w, p does not change with turbulence models except for the contributions from ν_t derivatives w.r.t. all the solving variables. Therefore ν_t derivatives are implemented in the "Constitutive Laws" which already has an interface to compute and provide derivatives.

On the other hand, scalar transport equation has only four coefficients which need to be given in the primal problem, consequently only four derivatives of coefficients required in the adjoint problem. Therefore an interface with "CalculateEffectiveVelocityDerivative", "CalculateEffectiveKinematicViscosityDerivative", "CalculateReactionTermDerivative", and "CalculateSourceTermDerivative" is implemented to compute each coefficients' derivatives respectively. These methods are made to compute derivatives on the fly because they are always only called once per gauss point so there is no point in storing them.

3.1.3 Stabilized residual derivatives

Equation (3.5), equation (3.6), and equation (3.7) illustrate strong form residuals of the governing equations. These strong form residuals are transformed to weak formulations in the context of FEM. The bare weak formulations of these strong form residuals are present with numerical instabilities (refer chapter 2), therefore stabilization terms are introduced in the primal problem. Hence it is required to compute derivatives of the weak formulation (i.e. derivatives of $R_u, R_v, R_w, R_k, R_\epsilon, R_\omega$) which includes derivatives of coefficients as well as stabilization terms. "RFCVariableDerivatives" class is designed to compute these derivatives. It takes one template argument "T" which provides derivatives of the coefficients used in the scalar transport equations. As an example, in figure 3.1, "RFC_CDR_U_Derivative" represents $\underline{\mathbf{R}}_{k,u}$ derivative computation class, "RFC_CDR_P_Derivative" represents $\underline{\mathbf{R}}_{k,p}$ derivative computation class.

3.1.4 Adjoint element formulation

The most naive way of computing adjoint local element matrix would be to compute each residual contribution matrix separately and add the final discrete matrices at the end. This may require computing same gauss point quantities repeatedly which will have a major impact on the computational efficiency.

Algorithm 4 illustrates basic flow in building the adjoint element matrix. $d\phi$ corresponds to data holders for each equation $\forall \phi \in \{u, v, w, p, k, \epsilon\}$. These equation data holders are implemented such that stack memory allocation is maximized and heap memory allocations are minimized. $d\phi w$ holds respective residual equation's (i.e. ϕ) derivatives with respect to each variable (i.e. w) used in the residual equation. It has one constructor argument which passes its residual equation data holder (i.e. $d\phi$). So these equation derivatives are designed not as data holders but as execution methods using data from equation data holders. This is because there needs to be a different object for each equation and its derivatives; and each derivative with respect to design variable may not hold common data. All of these objects are created in the stack memory space to enhance the efficiency of the overall computation. "CalcGaussPointData" method involves calculating gauss point data of the primal and common data for the adjoints as well. This is done once and given to all the equation derivative objects (i.e. $d\phi w$). "CalcGaussPointResidualsDerivativeContributions" method takes in output vector (i.e. $r\phi w$), nodal index (i.e. c), and derivatives of shape function data.

Figure 3.1 illustrates a representative class diagram of the adjoint element implementation. "ConvectionDiffusionReactionElement" is the primal el-

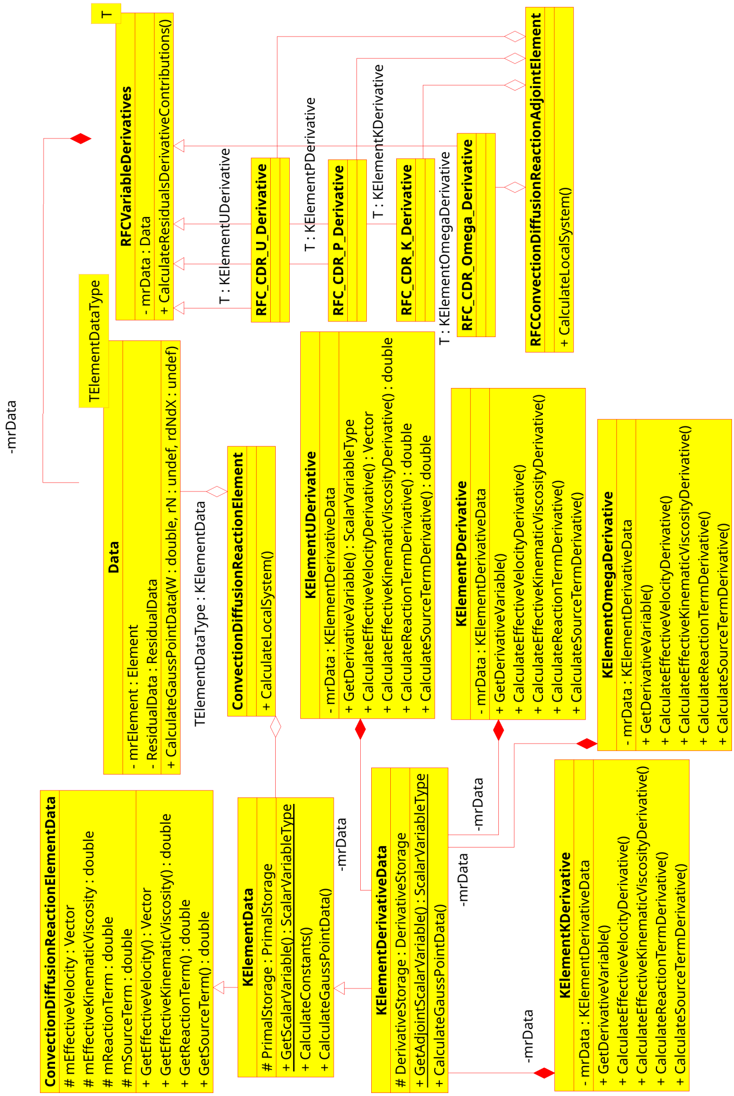

T
RFCVariableDerivatives
- mrData : Data
+ CalculateResidualsDerivativeContributions()
TElementDataType
Data
- mrElement : Element
- ResidualData : ResidualData
+ CalculateGaussPointData(W : double, rN : undef, rdNdX : undef)
-mrData
TElementDataType : KElementData
ConvectionDiffusionReactionElement
+ CalculateLocalSystem()
T : KElementUDerivative
RFC_CDR_U_Derivative
T : KElementPDerivative
RFC_CDR_P_Derivative
T : KElementKDerivative
RFC_CDR_K_Derivative
T : KElementOmegaDerivative
RFC_CDR_Omega_Derivative
RFCConvectionDiffusionReactionAdjointElement
+ CalculateLocalSystem()
KElementUDerivative
- mrData : KElementDerivativeData
+ GetDerivativeVariable() : ScalarVariableType
+ CalculateEffectiveVelocityDerivative() : Vector
+ CalculateEffectiveKinematicViscosityDerivative() : double
+ CalculateReactionTermDerivative() : double
+ CalculateSourceTermDerivative() : double
KElementPDerivative
- mrData : KElementDerivativeData
+ GetDerivativeVariable()
+ CalculateEffectiveVelocityDerivative()
+ CalculateEffectiveKinematicViscosityDerivative()
+ CalculateReactionTermDerivative()
+ CalculateSourceTermDerivative()
KElementOmegaDerivative
- mrData : KElementDerivativeData
+ GetDerivativeVariable()
+ CalculateEffectiveVelocityDerivative()
+ CalculateEffectiveKinematicViscosityDerivative()
+ CalculateReactionTermDerivative()
+ CalculateSourceTermDerivative()
ConvectionDiffusionReactionElementData
# mEffectiveVelocity : Vector
# mEffectiveKinematicViscosity : double
# mReactionTerm : double
# mSourceTerm : double
+ GetEffectiveVelocity() : Vector
+ GetEffectiveKinematicViscosity() : double
+ GetReactionTerm() : double
+ GetSourceTerm() : double
KElementData
# PrimalStorage : PrimalStorage
+ GetScalarVariable() : ScalarVariableType
+ CalculateConstants()
+ CalculateGaussPointData()
-mrData
KElementDerivativeData
# DerivativeStorage : DerivativeStorage
+ GetAdjointScalarVariable() : ScalarVariableType
+ CalculateGaussPointData()
-mrData
-mrData
-mrData
KElementKDerivative
- mrData : KElementDerivativeData
+ GetDerivativeVariable()
+ CalculateEffectiveVelocityDerivative()
+ CalculateEffectiveKinematicViscosityDerivative()
+ CalculateReactionTermDerivative()
+ CalculateSourceTermDerivative()

Figure 3.1: Primal and adjoint element class diagram.

Algorithm 4 Local system matrix building procedure for $k-\epsilon$ turbulence model.

```
procedure CalculateLocalSystem(Element E, Matrix Output)
    Ws ← Gauss point weights
    Ns ← Shape function values at gauss points
    dNdXs ← Shape function derivative values at gauss points
    $d\phi \leftarrow \underline{R}_\phi(E)$                                  ▷ $\forall\phi \in \{u,v,w,p,k,\epsilon\}$
    $d\phi w \leftarrow \underline{R}_{\phi,w}(d\phi)$                         ▷ $\forall\phi \times w \in \{u,v,w,p,k,\epsilon\}^2$
    $r\phi w \leftarrow Vector$
    for $g$ < Number of gauss points do
        $W \leftarrow Ws[g]$
        $N \leftarrow Ns[g]$
        $dNdX \leftarrow dNdXs[g]$
        $d\phi$.CalcGaussPointData(
                $W, N, dNdX$)                                  ▷ $\forall\phi \in \{u,v,w,p,k,\epsilon\}$
        for $c$ < Number of points do
            following two operations are $\forall\phi \times w \in \{u,v,w,p,k,\epsilon\}^2$
            $d\phi w$.CalcGaussPointResidualsDerivativeContributions(
                    $r\phi w, c, W, N, dNdX$)
            AssembleVectorToMatrix(Output, $r\phi w$)
    $d\phi$.CalcDataAfterGaussPointPointLoop()    ▷ $\forall\phi \in \{u,v,w,p,k,\epsilon\}$
    for $c$ < Number of points do
        following two operations are $\forall\phi \times w \in \{u,v,w,p,k,\epsilon\}^2$
        $d\phi w$.CalcAfterGaussPointPointLoop($r\phi w, c$)
        AssembleVectorToMatrix(Output, $r\phi w$)
```

ement which is used in solving the primal problem. This element includes the general weak formulation (refer equation (2.14)) with stabilization. The coefficients of generalized CDR equation (i.e. $\underline{u}_\phi$, ν_ϕ, s_ϕ, f_ϕ) is defined via respective equation's element data (example "KElementData") which is derived from "ConvectionDiffusionReactionElementData". The class "Data" is the data holder for stabilized general CDR weak formulation. "KElementDerivativeData" is the data holder for adjoint element's k equation which is derived from "KElementData" data to have maximum reuse of the existing classes. "KElementUDerivative", "KElementPDerivative", "KElementKDerivative", "KElementOmegaDerivative" are the derivatives of k equation w.r.t. u, p, k, ω variables. They are designed not to hold any data except their common data storage object reference (i.e. "KElementDerivativeData"). "RFCVariableDerivatives" class is used to compute derivatives of the stabilized CDR equation where coefficient derivatives are given by the template parameter "T" (refer "RFC_CDR_U_Derivative", "RFC_CDR_P_Derivative", ...). Finally the class "RFCConvectionDiffusionReactionAdjointElement" computes the adjoint element matrix using each equations' derivatives. This structure is made while avoiding virtual table calls, and all the derivatives are defined at compiler time using template arguments increasing the run-time efficiency. The other advantage of this structure is, it is easily extensible to any stabilized CDR formulation because one only has to implement its storage container and the coefficient derivatives of the CDR transport equation.

3.1.5 Adjoint slip condition formulation implementation

Slip condition is one of the useful boundary conditions in fluid dynamics. When slip condition is applied, it makes the velocity along the boundary to be always in the tangential plane making the normal direction component to be zero. This is heavily used when simulating far field boundaries which are supposed to have zero gradient velocity in the wall normal direction near the boundaries due to zero traction along the far field surface. This can be achieved in different ways, one is applying constraints on the degrees of freedom which are perpendicular to the normal and penalizing them, secondly, solving the solution in a rotated coordinate system which is aligned with the normal. In the latter case, this can be achieved by simply using a rotation matrix as depicted in equation (3.8) where $\underline{\hat{R}}$ is the rotated residual and $\underline{\mathbb{C}}$ is the node based rotation matrix.

$$\underline{\hat{R}} = \underline{\mathbb{C}}\,\underline{R} \tag{3.8}$$

The adjoint problem formulation is depicted in equation (3.9).

$$\underline{\tilde{w}}^T \frac{\partial \underline{\hat{R}}}{\partial \underline{w}} = \underline{\tilde{w}}^T \underline{\mathbb{C}} \frac{\partial \underline{R}}{\partial \underline{w}} = -\frac{\partial J}{\partial \underline{w}}^T \tag{3.9}$$

Afterwards, shape sensitivity can be computed using equation (3.10).

$$\begin{aligned}\frac{dL}{d\underline{x}} &= \frac{\partial J}{\partial \underline{x}}^T + \underline{\tilde{w}}^T \frac{\partial \underline{\hat{R}}}{\partial \underline{x}} \\ &= \frac{\partial J}{\partial \underline{x}}^T + \underline{\tilde{w}}^T \left[\frac{\partial \mathbb{C}}{\partial \underline{x}} \underline{R} + \mathbb{C} \frac{\partial \underline{R}}{\partial \underline{x}}\right]\end{aligned} \tag{3.10}$$

The node based rotation matrix (i.e. $\underline{\mathbb{C}}$) is computed based on nodal normals. Nodal normals are computed by first computing boundary surface normals and then distributing boundary surface normals to their corresponding nodes. Therefore, the term $\frac{\partial \mathbb{C}}{\partial \underline{x}}$ will have contributions w.r.t. nodal position of itself and its neighbouring nodal positions as well. As we mentioned in previous chapters (refer chapter 1 and 2), it is important to have distributed memory supported adjoint problem solving capabilities considering the fact that CFD problems have a large number of degrees of freedom hence the adjoint problem will also have a large number of degrees of freedom. In the case of nodal normal computation in distributed memory architecture, it can be achieved via assemble methods available in "Kratos Multiphysics". But for adjoints, it is required to have node based rotation matrices derivatives w.r.t. node itself and its neighbouring nodes. These neighbouring nodes may not fall in the same distributed memory partition in a distributed memory problem solve. Therefore it is required to carefully collect all the derivatives from all partitions which are required and assemble the final matrix.

This is achieved through a concept called "GlobalPointers". GlobalPointers holds memory locations of each node in each owning distributed memory partition. If one node's neighbours are in a remote partition, then this node will have all the GlobalPointers for neighbour nodes. These GlobalPointers are used to identify and assemble the node based rotation matrix's derivatives.

3.2 Testing and Verification

The next part of this chapter is dedicated to one of the most important aspects of software implementations which is testing and verification. It is important to design unit tests which can test small parts of the more complicated adjoint solvers as well as to validate sensitivities obtained by them.

3.2.1 Unit testing

Unit tests are developed for each of the primal residual data holders (i.e. $d\phi$ in algorithm 4) with random value initialization for mesh nodes. Furthermore, unit tests for primal elements are also developed in the same manner by randomly initialized nodal quantities to ensure expected matrices are built when constructing local system matrices.

3.2.2 Verification

Verification is carried out using finite difference method at two different levels. First level is at the derivative data holder level (i.e. $d\phi w$ in algorithm 4, or "RFC_CDR_U_Derivative"... in figure 3.1). All the derivative methods in the interface are verified against finite difference sensitivities calculated from nodal quantity perturbations.

$$\frac{\partial q}{\partial p} \approx \frac{q(p+\Delta p)-q(p)}{\Delta p} \tag{3.11}$$

Equation (3.11) is used to calculate finite difference approximation to analytical sensitivities where q is the function of interest, p being the differentiation variable. The perturbations(i.e. Δp) are chosen carefully after a perturbation study to avoid having issues related to noise due to smaller perturbations, and to avoid having issues related to non-linearity due to larger perturbations. Secondly, verification based on finite difference sensitivities is carried out on the method "CalculateLocalSystem" for each turbulence model, each residual equation, and each dependent variable for comprehensive testing of the adjoint local element matrix.

3.3 Numerical Experiment

This numerical investigation is designed to investigate the influence on different two equation RANS turbulence models on drag optimization. The experiment is carried out with $k-\epsilon$, $k-\omega$, and $k-\omega-sst$ turbulence models.

3.3.1 Experimental setup

The fluid domain $\Omega = (-22.5D, 22.5D) \times (-16D, 16D) \subset \mathbb{R}^2$ chosen after a domain size study is illustrated in figure 3.2 where $D = 0.1\,m$. Inlet (i.e. Γ_{inlet} is applied with a constant velocity (i.e. u_{inlet}), and turbulence quantities determined using turbulence intensity of 5% and turbulent mixing length of $45D$. Outlet (i.e. Γ_{outlet}) is applied with $0\,Pa$ Dirichlet boundary condition for P, zero gradient boundary conditions for u, k, ϵ, ω variables. Slip condition (i.e. Γ_{far}) is applied on top and bottom slip boundary for u variable, and all other variables are applied with zero gradient boundary condition. Linear-log law wall functions developed by Launder et al. [80] are used on aerofoil boundary (i.e. Γ_s) to accommodate wide range of meshes with $y^+ \in (0, 300)$ in the first element near the wall boundary.

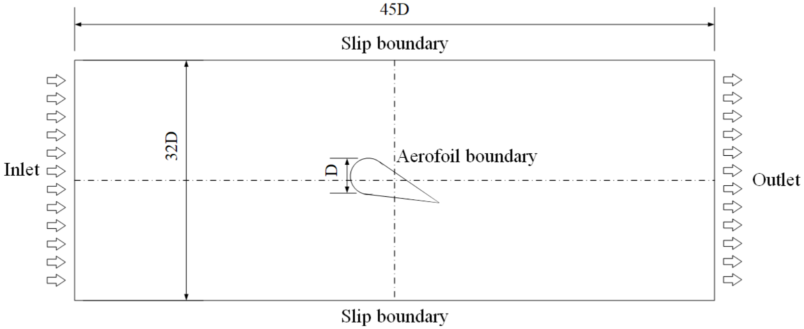
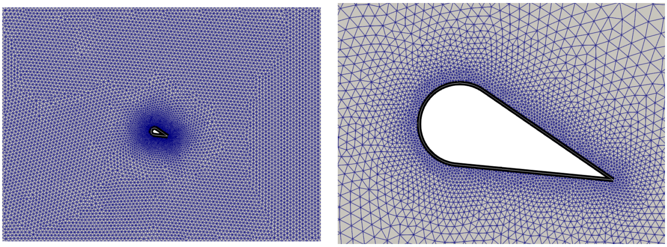


Figure 3.2: Aerofoil problem configuration used in RANS in 2D.

Figure 3.3 illustrates the overall mesh (refer figure 3.3(a)) and enlarged view of the same mesh near the initial aerofoil geometry (refer figure 3.3(b)) consisting of 20183 triangle elements.

3.3.2 Optimization procedure

Drag and lift forces are the interested scalar QOIs in this numerical experiment because, the aerofoil's usefulness depends on having maximum lift with minimum drag force. Equation (3.12) describes optimization problem of interest.



(a) Initial overall mesh

(b) Initial enlarged mesh

Figure 3.3: Initial mesh for 2D aerofoil optimization problem.

$$\min_{\underline{s}} \quad J_{drag}\left(\underline{w}(\underline{s}), \underline{s}\right)$$

subjected to

$$\begin{aligned} &\underline{R}_{\phi} = \underline{0} \quad \forall \phi \in \left\{u, v, w, p, k, (\epsilon \text{ or } \omega)\right\} \\ &G_{centroid}\left(\underline{s}\right) = 0 \\ &J_{lift}\left(\underline{w}(\underline{s}), \underline{s}\right) = J_{lift}\left(\underline{w}(\underline{s}_{initial}), \underline{s}_{initial}\right) \end{aligned} \tag{3.12}$$

$G_{centroid}$ is computed by averaging all nodal coordinate deviations along aerofoil boundary as illustrated in equation (3.13) where N represents number of nodes in Γ_s. It is applied to constrain aerofoil geometry to be present at the center of Ω for all the design iterations.

$$G_{centroid} = \left\| \frac{1}{N} \sum_{i=1}^{N} \underline{x}_i - \underline{x}_i^{initial} \right\|_2^2 \tag{3.13}$$

Initially this study was carried out with $\frac{J_{lift}}{J_{drag}}$ as the sole objective for minimization problem. This can give rise to numerical instabilities and significant oscillating response in the optimization process. Therefore, J_{drag} is used as the minimization problem while J_{lift} is used as a constraint keeping the initial lift force through out the optimization process to achieve a similar effect as maximizing $\frac{J_{lift}}{J_{drag}}$.

First of all, the aforementioned minimization problem is solved using a flow problem where $u_{inlet} = 1.0\,ms^{-1}$ with $\nu = 1 \times 10^{-1}\ m^2 s^{-1}$ representing a flow of $Re = 1$. Algorithm 5 is used as the optimization procedure for this problem. RANS turbulence models introduced and implemented in section 2.3.1 are used to solve these primal problems. Afterwards, algorithm 6 explains the procedure in detail which is used to solve aforementioned minimization problem for a chaotic problem where $\nu = 1 \times 10^{-5}\,m^2 s^{-1}$. The chaotic minimization problem is solved with and without frozen turbulence assumption for $k - \epsilon$ and $k - \omega$ turbulence models. Each of them consists of four different phases where each phase is performed with 100 design iterations. First two phases are considered as ramping up phases where inlet is applied with constant velocity of $u_{inlet} = 1.0\,ms^{-1}$ which corresponds to $Re = 1 \times 10^4$. This is followed by a second two phases, which apply constant inlet velocity of $u_{inlet} = 10.0\,ms^{-1}$ which corresponds to $Re = 1 \times 10^5$. After each phase, an Ω from all the design iterations of that phase is selected corresponding to best lift to drag ratio. Then this selected Ω is refined using AMR based on Hessian approach (Mataix Ferrándiz [90]). Afterwards, refined mesh is used as the initial mesh for the next phase.

Algorithm 5 2D aerofoil lift to drag ratio optimization procedure for non-chaotic flow.

```
$\Omega^j \leftarrow$ initial design with discretized fluid domain
$j \leftarrow 1$
for $j \leq 400$ do
    Solve primal problem with $\Omega_j$
    Solve $J_{lift}$ adjoint problem with $\Omega_j$ and calculate $\frac{\partial}{\partial \underline{s}}\left(J_{lift}\right)$
    Solve $J_{drag}$ adjoint problem with $\Omega_j$ and calculate $\frac{\partial}{\partial \underline{s}}\left(J_{drag}\right)$
    Compute $\frac{\partial}{\partial \underline{s}}\left(G_{centroid}\right)$ in $\Omega_j$
    Compute final shape update using gradient projection method
    Compute vertex morphed shape update
    Change $\Gamma_s$ according to shape update
    $j \leftarrow j + 1$
    $\Omega_j \leftarrow$ solved mesh for updated shape
```

Both $\frac{\partial}{\partial \underline{x}}\left(J_{lift}\right)$ and $\frac{\partial}{\partial \underline{x}}\left(J_{drag}\right)$ require the same primal solution for each design iteration. Therefore, these two response functions are developed

Algorithm 6 2D aerofoil lift to drag ratio optimization procedure for chaotic flow.

```
$\Omega_0^{best}$ ← initial design with discretized fluid domain
$i = 1$
for $i \leq 4$ do
    if $i \leq 2$ then
        $u_{inlet} = 1.0\, m s^{-1}$                ▷ Corresponds to $Re = 1 \times 10^4$
    else
        $u_{inlet} = 10.0\, m s^{-1}$               ▷ Corresponds to $Re = 1 \times 10^5$
    $j = 1$
    $\Omega_j \leftarrow \Omega_{i-1}^{best}$
    for $j \leq 100$ do
        Solve primal problem with $\Omega_j$
        Solve $J_{lift}$ adjoint problem with $\Omega_j$ and calculate $\frac{\partial}{\partial \underline{s}}\left(J_{lift}\right)$
        Solve $J_{drag}$ adjoint problem with $\Omega_j$ and calculate $\frac{\partial}{\partial \underline{s}}\left(J_{drag}\right)$
        Compute $\frac{\partial}{\partial \underline{s}}\left(G_{centroid}\right)$ in $\Omega_j$
        Compute final shape update using gradient projection method
        Compute vertex morphed shape update
        Change $\Gamma_s$ according to shape update
        $j \leftarrow j + 1$
        $\Omega_j$ ← Optimized mesh using Mmg mesh optimization
    $\Omega_i^{best}$ ← Refined $\Omega_k \left\{ k \in [1, 100] : \left(\frac{J_{lift}}{J_{drag}}\right)_m \leq \left(\frac{J_{lift}}{J_{drag}}\right)_k \forall m \in [1, 100] \right\}$
    $i \leftarrow i + 1$
```

such that whichever is computed first will solve the primal problem and store solution fields using check pointing method developed using HDF5 format (The HDF Group [127]). Then the second response function will use the same solution data from the primal solution which is already solved. Afterwards geometric constraint's shape derivatives are computed (i.e. $\frac{\partial}{\partial \underline{x}}\left(G_{centroid}\right)$). Gradient projection method by Geiser et al. [51] is used to combine constraints and response function shape sensitivities to obtain final shape update on Γ_s. Computed shape updates consist of numerical noise. Therefore vertex morphing method by Baumgärtner et al. [13] and Bletzinger [16] is used to smoothen the computed shape update. Steepest descend algorithm is used to solve minimization problem with a constant step size within a phase. This step size is reduced between each phase by 50% to allow further exploration of the design space to optimize the obtained design in the previous phase.

3.4 Results

Until now we have discussed implementation of steady state adjoints for sensitivity analysis and introduced numerical experiments. This section focusses on systematic analysis and discussion of results from the numerical experiments. First, results from the optimization process with $Re = 1$ non-chaotic flow is discussed, then followed by the discussion of results from the optimization process in the chaotic flow.

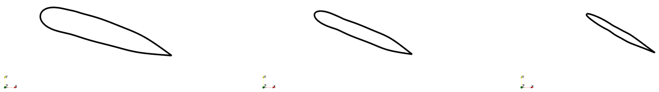
3.4.1 Non-chaotic flow optimization process

First Γ_s surfaces obtained via optimization process using RANS turbulence models are compared. Then it is followed by the lift to drag ratio variation comparison for the same turbulence models. Afterwards, lift forces' and drag forces' variations are compared between the RANS turbulence models. The implemented variation of RANS $k-\epsilon$ turbulence model has deficiencies in modelling low Re flow problems, therfore this non-chaotic flow problem is solved using only RANS $k-\omega$ and $k-\omega-sst$ turbulence models.

Figure 3.4 and figure 3.5 illustrate Γ_s surfaces in each design iteration obtained by the optimization process using the RANS $k-\omega$ and $k-\omega-sst$ turbulence models respsectively. Design surfaces for iterations 50, 100 and 400 are depicted in figure 3.4(a), figure 3.4(b) and figure 3.4(c) for $k-\omega$ turbulence model and figure 3.5(a), figure 3.5(b) and figure 3.5(c) for $k-\omega-sst$ turbulence model. They illustrate that, for the flow problem with $Re = 1$ non-chaotic case, the designs obtained from the optimization process using different RANS turbulence models do not exhibit significant differences in each design iteration.

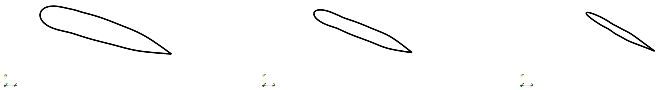
Lift to drag ratio comparison

Figure 3.6 comapres lift to drag ratio variations in design iterations for RANS $k-\omega$ and $k-\omega-sst$ turbulence model. This depicts that corresponding lift to drag ratios do not show differences for different turbulence



(a) Design iteration 50 (b) Design iteration 100 (c) Design iteration 400

Figure 3.4: Design iterations' Γ_s surfaces for optimization process with $k-\omega$ turbulence model used in $Re=1$ flow problem.



(a) Design iteration 50 (b) Design iteration 100 (c) Design iteration 400

Figure 3.5: Design iterations' Γ_s surfaces for optimization process with $k-\omega-sst$ turbulence model used in $Re=1$ flow problem.

models because obtained Γ_s surfaces are similar in each design iteration. This indicates that the use of different RANS turbulence models can obtain similar results using the optimization procedure for $Re=1$ case.

Lift and drag force comparison

In order to further analyse lift to drag ratios (refer figure 3.6), the lift force variations (refer figure 3.7) and drag force variations (refer figure 3.8) in the optimization process for $Re=1$ flow problem solved using RANS $k-\omega$ and $k-\omega-sst$ turbulence models are presented here.

Lift forces computed using each turbulence model depict values having same order of magnitude with small differences. They also illustrate rapid fluctuations between iterations after design iteration 75 eventually causing lift to drag ratios to exhibit the same rapid fluctuations in design iterations for both turbulence models. Drag force fluctuations depicted in figure 3.8

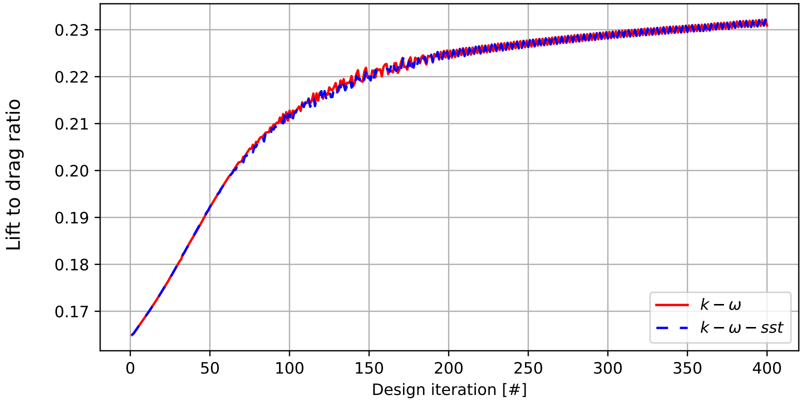


Figure 3.6: Lift to drag ratio variations in design iterations for different turbulence models in $Re = 1$ flow optimization.

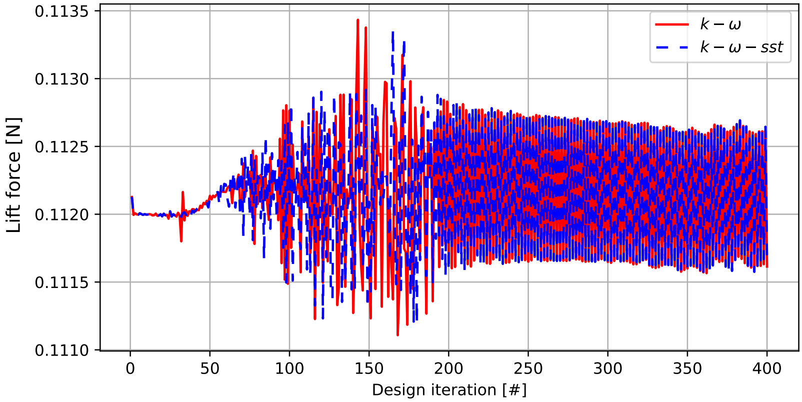


Figure 3.7: Lift force variations in design iterations for different turbulence models used in flow problem with $Re = 1$.

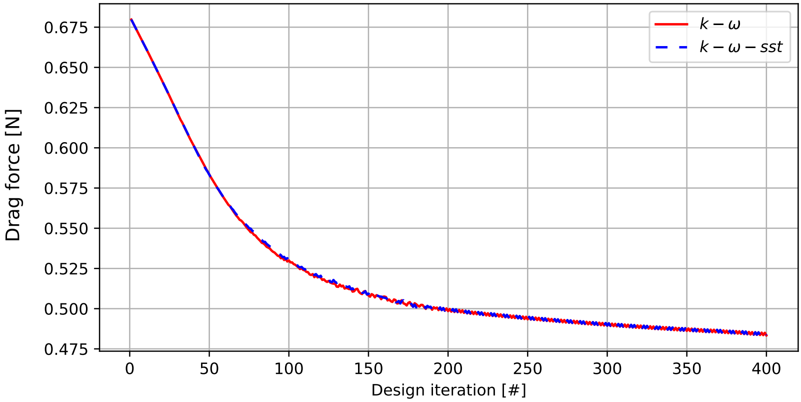


Figure 3.8: Drag force variations in design iterations for different turbulence models used in flow problem with $Re = 1$.

show decreasing trend for optimization procedures with each turbulence model. Both RANS $k-\omega$ and $k-\omega-sst$ turbulence models show similar reductions in drag force indicating both optimization cases with different RANS turbulence models present similar results.

3.4.2 Chaotic flow optimization process

Turning now to the chaotic flow results, first lift to drag force ratio variations are compared between each turbulence model obtained for chaotic flow problem using the optimization procedure where sensitivities are computed with and without the frozen turbulence assumption. This is followed by comparison of drag force minimization in the same optimization process. Afterwards, a discussion about constraints are also carried out.

Lift to drag ratio comparison

Figure 3.9 illustrates lift to drag force ratio variations in each design iteration for each turbulence model used in the optimization procedure. Best design iterations' design surfaces (i.e. Γ_s) for each turbulence model and each phase are illustrated in table 3.1 where frozen turbulence is assumed in sensitivity calculation and in table 3.2 where sensitivities are computed without the frozen turbulence assumption.

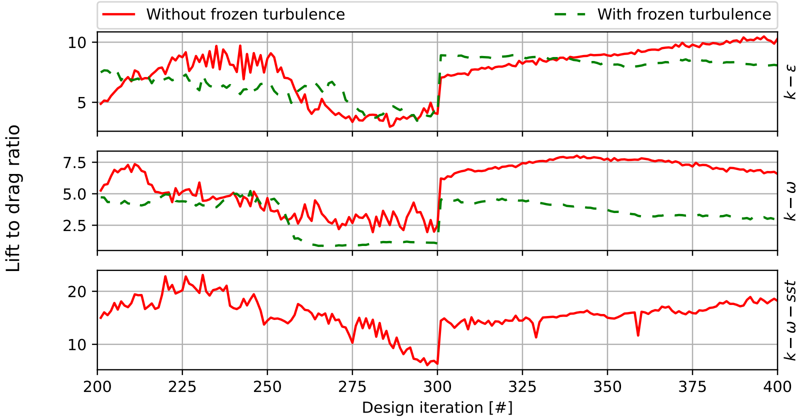
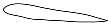
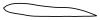
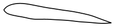
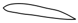
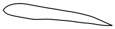
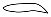
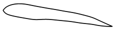
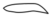


Figure 3.9: Lift to drag ratio variations in design iterations for different turbulence models for chaotic flow problem [Top: $k-\epsilon$, middle: $k-\omega$, bottom: $k-\omega-sst$].

Phase 1 (i.e. design iterations between 0-100), and phase 2 is conducted with $u_{inlet} = 1.0\, ms^{-1}$ which corresponds to $Re = 1 \times 10^4$, whereas phase 3 and phase 4 is conducted with $u_{inlet} = 10.0\, ms^{-1}$ which corresponds to $Re = 1 \times 10^5$. It can be seen from figure 3.9 that the lift to drag values vary within a small interval for phase 1 and 2 than the interval of phase 3 and 4. This is due to increased Re in phase 3 and 4 which causes notable differences in lift to drag ratio. There are significant jumps in lift to drag ratio present in the optimization process with all three turbulence models at design iterations 200 and 300. Main reason for this jump at 200^{th} iteration is the increase in Re. Notable jump at 300^{th} iterations is further analyzed in the following section 3.4.2.

When comparing the lift to drag ratios in a given tubulence models with and without the frozen turbulence assumption, it can be seen from figure 3.9 that $k-\omega$ turbulence model shows lower lift to drag ratios when frozen turbulence assumption is used. However, $k-\epsilon$ turbulence model shows almost similar lift to drag ratios with and without the frozen turbulence

Table 3.1: Best design iteration Γ_s surface for each turbulence model in each phase of optimization process in chaotic flow problem with frozen turbulence assumption.

Phase	Turbulence model	
	$k-\epsilon$	$k-\omega$
1	 Design itr. 94	 Design itr. 95
2	 Design itr. 106	 Design itr. 147
3	 Design itr. 205	 Design itr. 245
4	 Design itr. 305	 Design itr. 302

assumption. In order to understand this better, this is also further analyzed in the following section 3.4.2.

Lift and drag force comparison

The lift and drag force variations for phase 3 and 4 are illustrated in figure 3.10 and figure 3.11 respectively for optimization processes with all turbulence models.

Optimization process with $k-\epsilon$ with and without the frozen turbulence assumption shows a notable jump in lift force at 300^{th} iteration (refer figure 3.10). However the drag force does not show a significant jump for the same model when sensitivities are computed without the frozen turbulence assumption. In the case of optimization procedures with $k-\omega$ and $k-\omega-sst$ turbulence models exhibit significant jumps in drag force,

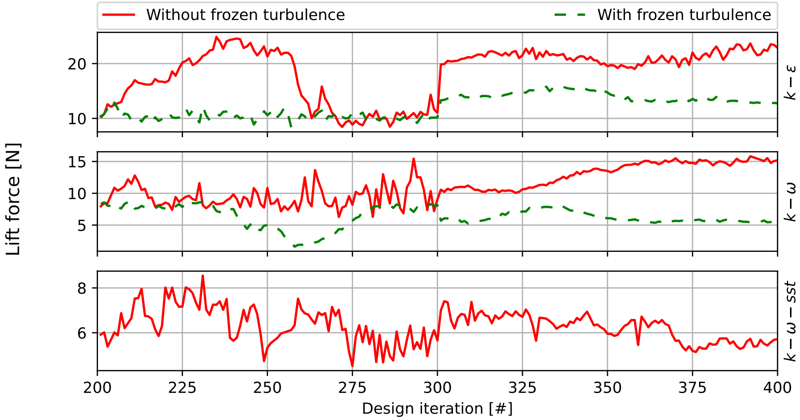


Figure 3.10: Lift force variations in design iterations in phase 3 and 4 for different turbulence models in solving chaotic flow problem [Top: $k-\epsilon$, middle: $k-\omega$, bottom: $k-\omega-sst$].

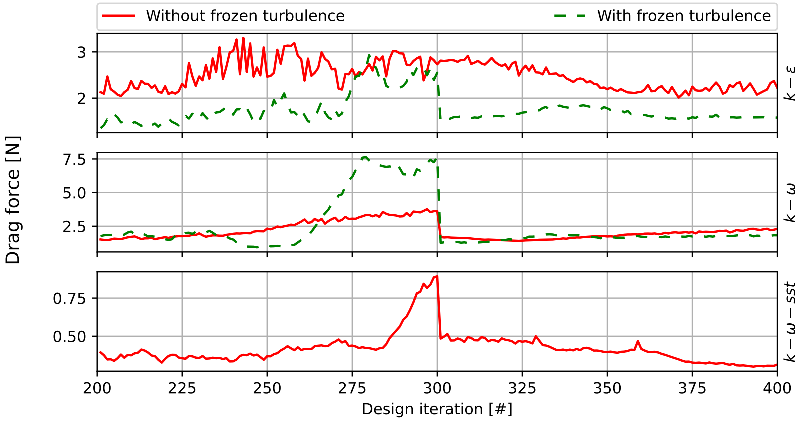
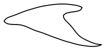
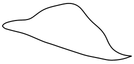
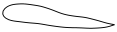
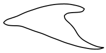
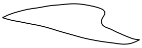
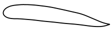
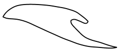
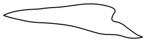
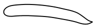
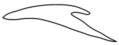
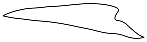
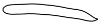


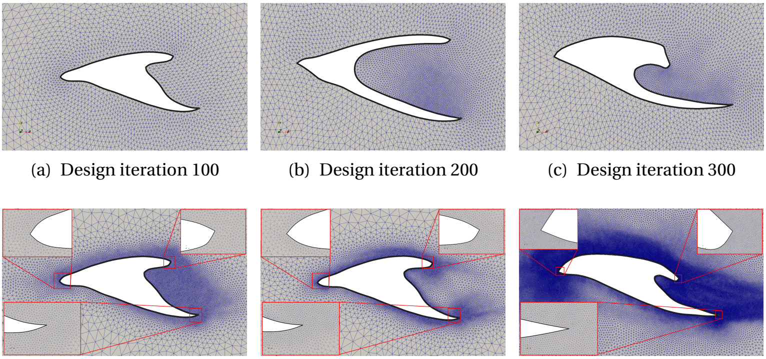
Figure 3.11: Drag force variations in design iterations in phase 3 and 4 for different turbulence models in solving chaotic flow problem [Top: $k-\epsilon$, middle: $k-\omega$, bottom: $k-\omega-sst$].

Table 3.2: Best design iteration Γ_s surface for each turbulence model in each phase of optimization process in chaotic flow problem without frozen turbulence assumption.

Phase	Turbulence model		
	$k-\epsilon$	$k-\omega$	$k-\omega-sst$
1	 Design itr. 97	 Design itr. 44	 Design itr. 46
2	 Design itr. 107	 Design itr. 153	 Design itr. 122
3	 Design itr. 242	 Design itr. 211	 Design itr. 231
4	 Design itr. 396	 Design itr. 341	 Design itr. 392

but not with the lift force. Lift forces calculated on all the design iterations obtained assuming frozen turbulence depict lower values than the same obtained using the respective turbulence model without the frozen turbulence assumption. In the case of drag forces, optimized designs obtained using $k–\epsilon$ turbulence model with the frozen turbulence assumption demonstrate lower values than its counter part which is obtained without using the frozen turbulence assumption. This is the reason why optimized designs obtained using $k-\epsilon$ turbulence model show similar lift to drag ratios despite the fact that frozen turbulence assumption is used or not. However, the optimized designs obtained using $k-\omega$ turbulence model with the frozen turbulence assumption illustrate similar drag forces compared to its counter parts which are obtained without the frozen turbulence assumption. This leads to reduced lift to drag ratios which are depicted in figure 3.9.

Figure 3.12 illustrates discretized fluid domain as well as the design surface on each of the design iterations where they have undergone AMR in the optimization procedure with $k-\epsilon$ turbulence model where sensitivities are computed without the frozen turbulence assumption. Figure 3.12(a) and figure 3.12(d) illustrate similar design surfaces, latter one with a more refined fluid domain discretization. Because of the similarity in design surface, lift to drag ratio does not show any significant jumps at 100^{th} iteration (refer figure 3.9). It suggests that performing AMR procedure is not affecting the final solution. Thereby indicating no mesh dependence is visible in these results for that particular flow Re. On the other hand, design iteration 201 and 301 show major differences in design surface which caused lift forces to have a considerable jump. This is because these iterations use discretized fluid domains obtained after performing AMR on their respective previous phase's best designs (refer table 3.2).


(a) Design iteration 100

(b) Design iteration 200

(c) Design iteration 300

(d) Design iteration 101

(e) Design iteration 201

(f) Design iteration 301

Figure 3.12: Design iteration fluid domain discretizations for optimization process with $k-\epsilon$ turbulence model used in chaotic flow problem solving.

In the same way, the jumps observed in optimization processes with $k-\omega$ and $k-\omega-sst$ turbulence models without the frozen turbulence aasumption can be explained by observing figure 3.13 and figure 3.14 respectively. They depict each fluid domain discretizations for each phases' initial and final design iterations where frozen turbulence is not assumed.

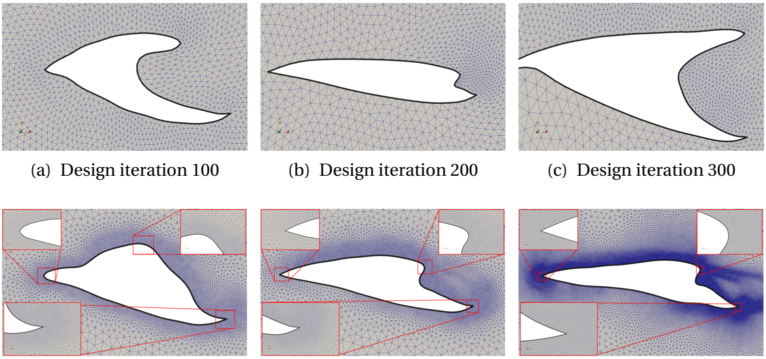

(a) Design iteration 100
(b) Design iteration 200
(c) Design iteration 300
(d) Design iteration 101
(e) Design iteration 201
(f) Design iteration 301

Figure 3.13: Design iteration fluid domain discretizations for optimization process with $k-\omega$ turbulence model used in solving chaotic flow problem.

Figure 3.14 depicts design surfaces obtained with $k-\omega-sst$ turbulence model. These designs show higher lift to drag ratio values in all the phases suggesting $k-\omega-sst$ turbulence model is superior in optimizing lift to drag ratio.

Optimization of these aerofoil surfaces under chaotic flows has been an interesting research field for past few decades (refer work by Ferreira et al. [42], Hicks et al. [58], and Najian Asl [100]). In order to compare the optimized design surfaces from the proposed approach with the literature, NACA0012 at $Re = 1 \times 10^5$ design surface with an AoA which result in having the same lift coefficient as of the best optimized design surface from the proposed optimization procedure is chosen. The best design from the current optimization approach is obtained from the optimization procedure using RANS $k-\omega-sst$ turbulence model. It has a chord length of $0.246\,m$ which corresponds to $C_L = 0.456$ and $C_D = 0.025$ which leads to $\frac{C_L}{C_D} = 18.938$. The corresponding values for NACA0012 are $C_L = 0.454$, $C_D = 0.014$ and $\frac{C_L}{C_D} = 32.429$. This indicates that even though the proposed optimization procedure is capable of optimizing the given initial geomeotry, it is still not reaching for the design with the most optimum lift to drag ratio. One reasoning behind this is, the optimization procedure introduced in this section uses constant step sizes and constant vertex morphing radii which makes the optimizer to adapt designs with coarser step sizes and vertex morphing radii at the obtained optimum design. This results in oscillating behaviour for the observed QOI. Another reason for not obtaining the most optimium design is that, the phases in the optimization procedure is not run until each phases convergence of the QOI.

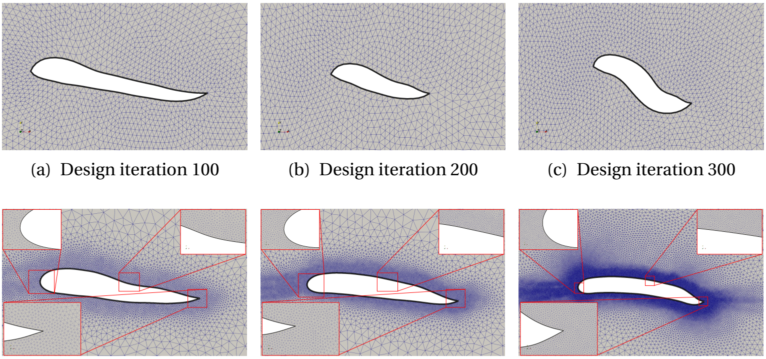
Another siginificant observation is, that the optimized designs for $k-\epsilon$ and $k-\omega$ where sensitivities are computed without the frozen turbulence assumption illustrate novel features compared to most optimum design given by the NACA0012 configuration. These novel protruding features are visible at the back of the aerofoil configuration. However, these protruding features are not visible in the same turbulence models when frozen turbulence assumption is used resulting with almost similar shape as in the designs obtained from $k-\omega-sst$ turbulence model without using the frozen turbulence assumption. This indicates that, not having the frozen turbulence assumption may result with designs having protruding features after flow seperation point for $k-\epsilon$ and $k-\omega$ turbulence models. On the other hand, $k-\omega-sst$ turbulence model without the frozen turbulence assumption is capable of computing better optimized designs than any other turbulence model with or without the frozen turbulence assumption indicating its better applicability in an optimization problem.

Constraints

In addition to discussion about lift and drag, it is also important to analyze behavior of constraints throughout the optimization procedure. Two external equality constraints are used in this numerical experiment.

Figure 3.15 illustrates $G_{centroid}$ constraint value variation with design iterations for each of the turbulence models used. This constraint does not exhibit any non-linear behavior, thereby it does not show any significant constraint violations. This is also evident in the discretized fluid domains where the optimization procedure is successful in placing the design surface closer to the initial position.

Although $G_{centroid}$ has not shown significant violations, J_{lift} constraint shows notable violations. This is because J_{lift} being a highly non-linear constraint (refer figure 3.16). Despite the significant constraint violations, the significance of constraint violations are reduced more in phase 4 when



(a) Design iteration 100

(b) Design iteration 200

(c) Design iteration 300

(d) Design iteration 101

(e) Design iteration 201

(f) Design iteration 301

Figure 3.14: Design iteration fluid domain discretizations for optimization process with $k-\omega-sst$ turbulence model used in solving chaotic flow problem.

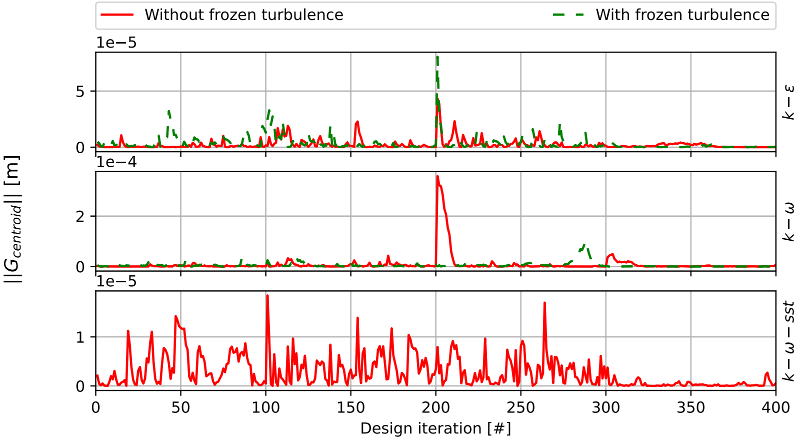


Figure 3.15: $\left\|G_{centroid}\right\|$ constraint variations in design iterations for different turbulence models [Top: $k-\epsilon$, middle: $k-\omega$, bottom: $k-\omega-sst$].

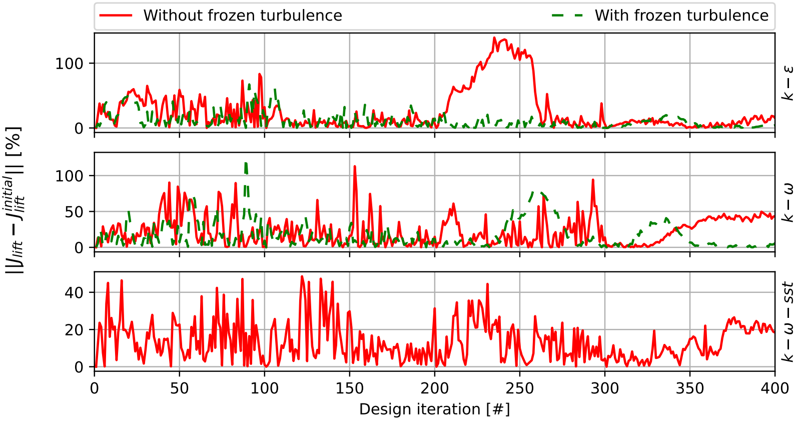


Figure 3.16: $\left\|J_{lift} - J_{lift}^{initial}\right\|$ constraint variations in design iterations for different turbulence models [Top: $k-\epsilon$, middle: $k-\omega$, bottom: $k-\omega-sst$].

compared to other phases of optimization procedure. This suggest that having a refined step size in gradient descend optimization algorithm helps in finding optimized shapes having relatively reduced constraint violations.

3.4.3 Comparison of non-chaotic and chaotic flow optimization processes

By looking at the results from section 3.4.1 and section 3.4.2, it is evident that optimization cases with chaotic flow where no frozen turbulence is assumed resulted in different Γ_s surfaces for different RANS turbulence models whereas non-chaotic flow resulted in similar Γ_s surfaces. This section focuses on investigating further to identify main cause for this behaviour in different turbulence models when utilized in chaotic flow optimization problems. Therefore, first the description on how drag and lift forces are computed is discussed, then followed by a discussion on the comparison of the primal and the adjoint solution fields of the first design iteration for the both non-chaotic and chaotic optimization cases for different RANS turbulence models.

Drag and lift force computation

In order to compute drag and lift forces, first reactions on the $\mathit{\Gamma}_s$ surface's mesh coordinates are computed by using the weak formulation of continuity equations (i.e. $\underline{R}_u$, $\underline{R}_v$ and $\underline{R}_w$) derrived from the strong form formulations depicted in equation (3.5) (Reader is referred to Cotela Dalmau et al. [35] for exact implementation details of the mentioned residuals). The residuals $\underline{R}_u$, $\underline{R}_v$ and $\underline{R}_w$ are computed using only the element contributions in the case where wall functions are used. Reaction force computation is illustrated in equation (3.14) where $\underline{D}$ represents final reaction force acting on the $\mathit{\Gamma}_s$ surface, R^a_ϕ being the weak form residual of a^{th} mesh coordinate in the $\mathit{\Gamma}_s$ surface for all $\phi \in [u, v, w]$.

$$\underline{D} = \sum_{\forall a \in \left\{\text{mesh node indices of } \mathit{\Gamma}_s\right\}} \begin{bmatrix} R^a_u \\ R^a_v \\ R^a_w \end{bmatrix} \tag{3.14}$$

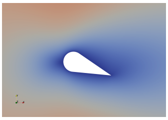
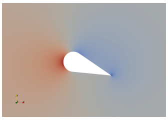
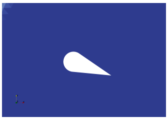
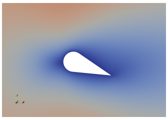
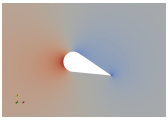
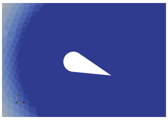
Once the $\underline{D}$ is computed then equation (3.15) is used to compute drag (i.e. J_{drag}) with drag directional vector (i.e. $\underline{\eta}_{drag} = [1, 0, 0]$) and lift (i.e. J_{lift}) with lift directional vector (i.e. $\underline{\eta}_{lift} = [0, 1, 0]$)

$$J_\phi = \underline{D}^T \underline{\eta}_\phi \tag{3.15}$$

Non-chaotic flow optimization first design iteration results

Table 3.3 illustrates primal solution fields of the first design iteration for optimization cases with RANS $k-\omega$ and $k-\omega-sst$ turbulence models. It depicts velocity magnitude distribution (i.e. $|\underline{u}|$), pressure distribution (i.e. p) and turbulent viscosity distribution (i.e. ν_t). ν_t is calculated element wise therefore discontinuities between elements are visible and they are illustrated in the log scale to highlight the differences. It can be seen that, both $|\underline{u}|$ and p do not show any notable differences in considered RANS turbulence models. The differences in elemental ν_t are negligible when compared to kinematic viscosity of $1 \times 10^{-1}\ m^2 s^{-1}$ of the fluid.

Table 3.3: Comparison of non-chaotic flow first design iteration's primal solution fields

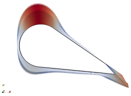
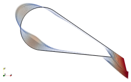
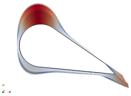
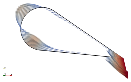
	$\lvert\underline{u}\rvert\,[m s^{-1}]$ $\lvert\underline{u}\rvert \in [0,1.5]$	$p\,[Pa]$ $p \in [-2.5,2.5]$	$\nu_t\,[m^2 s^{-1}]$ $\log(\nu_t) \in [-12,-8]$
$k-\omega$			
$k-\omega-sst$			
	Min ▬ Max		

Since the primal solution fields illustrate negligible differences, consequently the adjoint fields also depict negligible differences for all the turbulence models considered in the optimization case with non-chaotic flow problem. This is evident in table 3.4, where it illustrates directions and magnitudes of the drag sensitivities (i.e. $\left(\frac{dJ}{d\underline{s}}\right)_{drag}$) and the lift sensitivities (i.e. $\left(\frac{dJ}{d\underline{s}}\right)_{lift}$) after vertex morphing and rescaling their magnitudes to range $[0,1]$. Therefore, no significant differences in the Γ_s design surfaces are observed during the optimization procedure.

Chaotic flow optimization first design iteration results

As pointed out in the section 3.4.2, the optimization procedures with different RANS turbulence models where no frozen turbulence are assumed resulted in different Γ_s design surfaces. Therefore this section discusses results of the primal and the adjoint solution fields of the first design iteration of phase 1 in the optimization case with chaotic flow problem. Table 3.5 illustrates velocity magnitude (i.e. $\lvert\underline{u}\rvert$), pressure (i.e. p) and elemental turbulent viscosity (i.e. ν_t) distributions near the Γ_s design surface. Both $\lvert\underline{u}\rvert$ and p illustrate similar distributions and magnitudes for all three RANS turbulence models. However, the ν_t distributions are depicting large deviations. These deviations of the ν_t distributions are observed

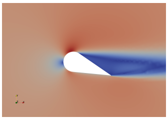
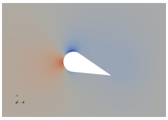
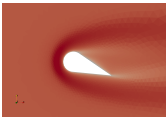
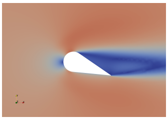
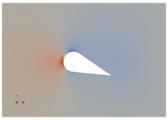
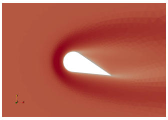
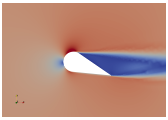
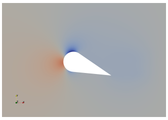
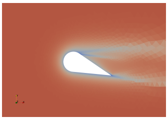
Table 3.4: Comparison of non-chaotic flow first design iteration's vertex morphed sensitivitiy fields

	$\left(\frac{dJ}{d\underline{s}}\right)_{drag}$ $[Nm^{-1}]$ $\left\lvert\left(\frac{dJ}{d\underline{s}}\right)_{drag}\right\rvert \in [0,1.0]$	$\left(\frac{dJ}{d\underline{s}}\right)_{lift}$ $[Nm^{-1}]$ $\left\lvert\left(\frac{dJ}{d\underline{s}}\right)_{lift}\right\rvert \in [0,1.0]$
$k-\omega$		
$k-\omega-$ sst		
	Min ▬ Max	

to be in between $\left[1\times10^{-6}\,m^2s^{-1}, 1\times10^{-2}\,m^2s^{-1}\right]$ whereas the fluid kinematic viscosity is $1\times10^{-5}\,m^2s^{-1}$. Due to these higher ν_t distributions, the reactions computed on Γ_s surface via these RANS turbulence models illustrate different lift and drag forces as depicted in figure 3.10 and figure 3.11 respectively.

Table 3.6 illustrates directions and magnitudes of the drag sensitivities (i.e. $\left(\frac{dJ}{d\underline{s}}\right)_{drag}$) and the lift sensitivities (i.e. $\left(\frac{dJ}{d\underline{s}}\right)_{lift}$) after vertex morphing and rescaling their magnitudes to range $\left[0,1\,Nm^{-1}\right]$ for the optimization case with chaotic flow problem. It is clearly evident from that, the first design iterations demonstrate drag and lift sensitivities with different directions as well as different relative magnitudes in all the turbulence models where sensitivities are computed without the frozen turbulence assumption. However, sensitivities calculated using the frozen turbulence assumption in $k-\epsilon$ and $k-\omega$ turbulence models are similar to sensitivities calculated without the frozen turbulence assumption in $k-\omega-sst$ turbulence model. These similarities in sensitivities lead to similar optimized designs in $k-\epsilon$ and $k-\omega$ turbulence models obtained with the frozen turbulence assumption and optimized designs in $k-\omega-sst$ turbulence model without using the frozen turbulence assumption which are depicted in table 3.1 and table 3.2. It is evident that, even with the frozen

Table 3.5: Comparison of chaotic flow first design iteration's primal solution fields

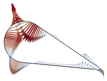
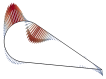
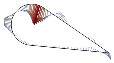
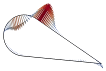
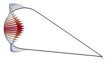
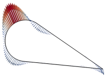
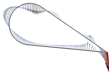
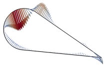
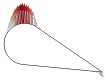
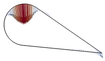
	$\lvert \underline{u} \rvert \, [m s^{-1}]$ $\lvert \underline{u} \rvert \in [0, 1.5]$	$p \, [Pa]$ $p \in [-1.0, 1.0]$	$\nu_t \, [m^2 s^{-1}]$ $\log(\nu_t) \in [-6, -2]$
$k-\epsilon$			
$k-\omega$			
$k-\omega-$ sst			
	Min Max		

turbulence assumption, the obtained sensitivitiy fields and the optimized designs are not the same for all the turbulence models. This is because, all of the turbulence models demonstrate differencs in ν_t distribution (refer table 3.5) which cause differences in the calculated sensitivities. These differences in sensitivities are increased when they are computed without the frozen turbulence assumption because of the differences of ν_t as well as differences in $\frac{\partial \nu_t}{\partial \phi}$.

3.5 Conclusion and Outlook

In this chapter, the aim was to introduce a flexible/extensible adjoint formulation framework developed whilst having the least amount of affect on computational performance and memory consumption. This was achieved by templated element local matrix formulation structure discussed in section 3.1. Comprehensive testing and verification is also carried out on the implemented elements. Finally, a numerical study is introduced to show robustness and flexibility of implemented adjoint formulation by using them to minimize drag whilst lift is kept at a constant using $k-\epsilon$, $k-\omega$,

Table 3.6: Comparison of chaotic flow first design iteration's vertex morphed sensitivitiy fields

	$\left(\frac{dJ}{d\underline{s}}\right)_{drag}$ $[Nm^{-1}]$ $\left\|\left(\frac{dJ}{d\underline{s}}\right)_{drag}\right\| \in [0,1.0]$		$\left(\frac{dJ}{d\underline{s}}\right)_{lift}$ $[Nm^{-1}]$ $\left\|\left(\frac{dJ}{d\underline{s}}\right)_{lift}\right\| \in [0,1.0]$	
	Without frozen turbulence	With frozen turbulence	Without frozen turbulence	With frozen turbulence
$k-\epsilon$				
$k-\omega$				
$k-\omega-$ sst				
	Min ▬ Max			

and $k-\omega-sst$ turbulence models for a chaotic and non-chaotic flow. Results of this minimization problems show even if the initial geometry, boundary conditions are the same, but depending on the used turbulence model, the local minima obtained can be different for chaotic flow problem whereas non-chaotic flow obtains similar local minima. Even though the final obtained local minima is different for each turbulence model, they show significant improvement in lift to drag objective which makes all the turbulence models eligible to be used in maximization of lift to drag ratio in this context. However, only minor differences are notable in the design surfaces obtained in optimization procedures using different turbulence models for non-chaotic flow problem. This indicates, use of turbulence models and their derivatives in high Re flow problems may result in different designs if used in an optimization procedure as explained in this section. The dicussion included results obtained with and without the frozen turbulence assumption for $k-\epsilon$ and $k-\omega$ turbulence models.

They indicate similar designs as in the case with the $k-\omega-sst$ turbulence model where sensitivities are computed without the frozen turbulence assumption. The optimized design surfaces obtained with the frozen turbulence assumption do not illustrate protruding features whereas optimized designs with the same turbulence models where no frozen turbulence are assumed demonstrate protruding features after the flow seperation point. It highlights that, the artificial protrutions presented in $k-\epsilon$ and $k-\omega$ turbulence models without the frozen turbulence assumption can be mitigated by using the frozen turbulence assumption.

The optimization process used in this section solves drag minimization problem with lift and geometric center constraint firstly with non-chaotic flow problem in a single phase and secondly with chaotic flow problem in four phases. In the case of chaotic flow problem optimization, the initial fluid domain for a phase is taken after performing AMR on the best design surface of the previous phase. The current investigation is limited by a constant step size within a phase which is reduced by 50% after each phase and with a constant number of design iterations per phase. Due to constant step size limitation, it is shown that lift constraints show relatively significant violations in first three phases, which is reduced in phase 4 with reduced step size. Therefore, future work needs to be done to establish adaptive step sizing for gradient descend algorithm of the optimization process. Another limitation is performing AMR only at the end of each phase. This limitation is there because of high computational time/memory requirement in the used AMR methodology. Therefore, a further study could assess the effect of performance bottlenecks of used AMR procedure and rectify them to be used in this optimization procedure efficiently.

CHAPTER 4

TRANSIENT SENSITIVITY ANALYSIS

Problems in transient (i.e. time dependent) nature also hold a great importance in engineering sciences. This is because solutions obtained via steady methods may include assumptions/approximations which hinder the final expected outcome, thus making them less accurate or/and even make those solutions to be useless. Therefore, there has been increased interest in solving transient problems in many fields of engineering. For example, aerospace computational engineering (Mak et al. [89], Talnikar et al. [125], and Tucker et al. [128]), multidisciplinary engineering with FSI (Andre et al. [5], Johansen et al. [71], and Korobenko et al. [75]), and multidisciplinary engineering with thermo-structural (Bielecki et al. [15])

Most of the transient problems encountered in CWE are solved with either high fidelity solution methodologies such as DNS, LES or Detached Eddy Simulation (DES) or low fidelity methodologies such as URANS due to their transient nature (refer chapter 1). These high fidelity solution methodologies are becoming increasingly popular due to advancements in computational architecture which can provide greater computational power. However, using these high fidelity methodologies to solve transient problems can lead to solutions with chaotic behavior depending on the problem's flow properties. Chaotic behavior may be explained as deterministic dynamical system exhibiting aperiodic behavior that depends sensitively on the initial conditions (Strogatz [122]). This behavior was also observed by Lorenz [84] in even simpler studies performed to understand behavior of convection rolls in atmosphere. This is named as "Butterfly effect". As a result of the butterfly effect, the time instantaneous values of QOI may become useless in engineering decision making. However, long time averaged QOIs is of interest for engineering disciplines in ergodic systems[1]. Therefore, specially in engineering design, it is of great importance to study sensitivity of long time averaged QOIs with respect to changes in boundary conditions.

This chapter is organized as follows. First the big problem in sensitivity computation of a chaotic flows is presented in section 4.1. This is then followed by the definition of the transient chaotic problem which is considered for evaluation of sensitivities in section 4.2. Afterwards, two novel adjoint stabilization methods are introduced in section 4.3 to tackle the problem defined in section 4.1. The proposed stabilization methods are applied to a chaotic flow example, and verification is carried out in section 4.4. Finally, the proposed stabilization method is used in a transient QOI optimization problem involving a non-chaotic and chaotic flow in section 4.5.

4.1 The big problem

The big problem in calculating sensitivities of long time averaged QOIs in chaotic flow solution is its high sensitivity for initial conditions (i.e. "Butterfly effect") which makes the computed sensitivities useless. This is illustrated in figure 4.1 which represents work done by Warnakulasuriya et al. [140]; computing drag sensitivities using adjoint method for a flow around a cylinder with $Re = 1 \times 10^4$. Even though it seems like converging in the forward time marching problem, in transient adjoint problems, final shape sensitivity is computed by computing adjoint values in reverse time, hence the exponential growth which leads to useless sensitivities. Computing primal problem by forward time marching and adjoint problem

[1] Systems on which long time averaged QOI does not depend on initial conditions (Méla et al. [92])

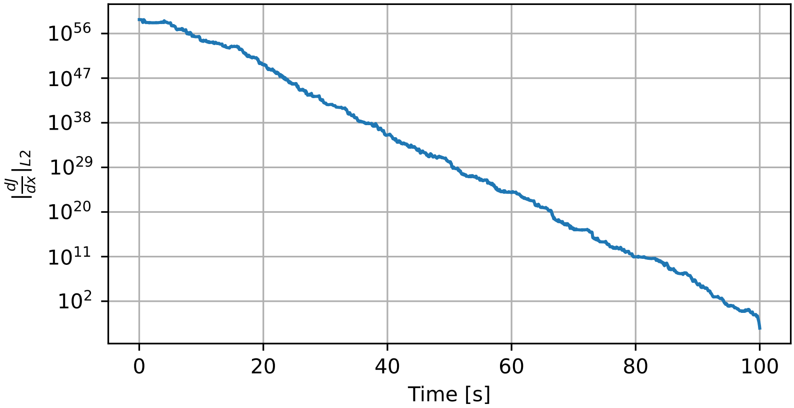


Figure 4.1: Divergence of sensitivities computed with adjoint method Warnakulasuriya et al. [140].

by backward time marching raise challenges in transferring solution from primal problem to adjoint problem, which is also part of the big problem for numerical solver implementations.

Assume the following general strong form of the governing equations depicted in equation (4.1) where $\underline{w}$ represents all the solving variables such as velocity (i.e. u), pressure (i.e. p). s represents a design parameter for which it is of interest to compute sensitivities defined for domain $\Omega \subset \mathbb{R}^d$ where d is the dimensionality of the domain.

$$\frac{\partial \underline{w}}{\partial t} = \underline{f}\left(\underline{w}, s\right) \tag{4.1}$$

As mentioned in the introduction, due to the chaotic behavior, the long time averaged values of a QOI is of interest in engineering decision making. Let's assume QOI as defined in equation (4.2).

$$\bar{J}(s) = \frac{1}{T}\int_0^T J\left[\underline{w}(t,s)\right]dt, \quad \bar{J}^{\infty}(s) = \lim_{T\to\infty} \bar{J}(s) \tag{4.2}$$

Investigations carried out by Blonigan et al. [17] have shown that many non-chaotic systems converge $\bar{J}$ towards $\bar{J}^{\infty}$, but if the system is chaotic, then there exists at least one positive Lyapunov coefficient (Wolf et al. [146]) which makes the adjoint sensitivity linearization depicted in right hand side of equation (4.3) to exhibit exponential growth as the limits of $T \rightarrow \infty$, thus making the limits of T un-commutable.

$$\frac{d\bar{J}^{\infty}}{ds} \neq \lim_{T\rightarrow\infty} \frac{d\bar{J}}{ds} \tag{4.3}$$

4.2 Transient problem definition

This chapter focuses on solving transient problems in the context of CFD. Therefore, governing equations include incompressible Navier-Stokes as the momentum equation depicted in equation (4.4) and continuity equation depicted in equation (4.5) in the residual strong forms.

$$\underline{r}_m = \partial_t \underline{u} + \underline{u} \cdot \nabla \underline{u} - \frac{1}{\rho} \nabla \cdot \underline{\boldsymbol{\sigma}} - \underline{f} = \underline{0} \tag{4.4}$$

$$r_c = \nabla \cdot \underline{u} = 0 \tag{4.5}$$

Where ρ is fluid density, $\underline{u}$ is velocity, $\underline{f}$ represents external forces acting on the fluid domain, and $\underline{\boldsymbol{\sigma}}$ represents stress tensor which is depicted in equation (4.6) with p being pressure, $\underline{\mathbf{I}}$ the identity matrix, μ being molecular dynamic viscosity of fluid.

$$\underline{\boldsymbol{\sigma}} = -p\underline{\mathbf{I}} + 2\mu \left[\nabla^s \underline{u} - \frac{1}{3} \left(\nabla \cdot \underline{u} \right) \underline{\mathbf{I}} \right] \tag{4.6}$$

In the case of two equation URANS turbulence models, momentum and continuity equations are supported by two additional scalar transport equations depicted by equation (4.7) where ϕ is the scalar being solved. ϕ scalar depends on the turbulence model being used. $k - \epsilon$ turbulence model is using ϕ as turbulent kinetic energy (i.e. k) and turbulence energy dissipation rate (i.e. ϵ), $k - \omega$ turbulence model is using ϕ as k and turbulent specific energy dissipation rate (i.e. ω). $k - \omega - sst$ turbulence model is used ϕ as k and ω as well. In the case of turbulence modelling, μ in equation (4.6) is replaced with $\mu = \mu_{molecular} + \mu_{turbulent}$ where $\mu_{turbulent}$ is calculated according to chosen two equation turbulence model.

$$r_\phi = \frac{\partial \phi}{\partial t} + \underline{u}_\phi \cdot \frac{\partial \phi}{\partial \underline{x}} - \frac{\partial}{\partial \underline{x}}\left(\nu_\phi \frac{\partial \phi}{\partial \underline{x}}\right) + s_\phi \phi - f_\phi = 0 \tag{4.7}$$

Time discretization is done using Bossak time integration method introduced by Wood et al. [147] as depicted in equation (4.8) where $\underline{w} = [\underline{u}, p, k, \epsilon\, o\, r\, \omega]$, $\underline{F}^n$ represents the discretized source terms, $\underline{\mathbf{C}}$ represents convective, diffusive and reaction terms in the discretized domain, and $\underline{\mathbf{M}}$ represents the mass matrix.

$$\underline{R}^n = \underline{F}^n - \underline{\mathbf{C}}^n \underline{w}^n - \underline{\mathbf{M}}^n \left[\left(1-\alpha_b\right)\underline{\dot{w}}^n + \alpha_b \underline{\dot{w}}^{n-1}\right] \tag{4.8}$$

Relationship between $\underline{w}$ and $\underline{\dot{w}}$ is depicted in equation (4.9) where γ_n is the newmark constant calculated from $\gamma_n = 0.5 - \alpha_b$, where $\alpha_b = -0.3$ is the Bossak constant.

$$\underline{H}^n = \underline{\dot{w}}^n - \frac{1}{\Delta t \gamma_n}\left(\underline{w}^n - \underline{w}^{n-1}\right) - \frac{\gamma_n - 1}{\gamma_n}\underline{\dot{w}}^{n-1} \tag{4.9}$$

The proposed methodology to solve transient adjoint problem in this chapter is also formulated using discrete adjoint method as in the case with steady state sensitivity analysis (refer chapter 3), thereby allowing to extend steady state methodologies to transient methodologies. Therefore the QOI illustrated in equation (4.2) needs to be discretized in spatial and temporal domains as depicted in equation (4.10)

$$\overline{J} = \frac{1}{N}\sum_{n=1}^{N} J\left(\underline{w}^n, s\right) = \sum_{n=1}^{N} J^n \quad where \quad J^n = \frac{J\left(\underline{w}^n, s\right)}{N} \tag{4.10}$$

Discretized residuals depicted in equation (4.8) and equation (4.9) are enforced as equality constraints over discretized fluid domain $\Omega^H \subset \Omega$. Therefore, a Lagrangian is formulated for transient problem similar to the steady state problem Lagrangian depicted in equation (3.1) followed by transient adjoint problem formulation as depicted in equation (4.11). Here, $\underline{\lambda}_1^n$ and $\underline{\lambda}_2^n$ are discretized time instantaneous Lagrange multipliers for constraint $\underline{R}^n$ and $\underline{H}^n$. Relationship between $\underline{\lambda}_1^n$ and $\underline{\lambda}_2^n$ can be obtained by taking the partial derivatives with respect to $\underline{\dot{w}}^n$ (refer appendix B).

$$\frac{\partial J^n}{\partial \underline{w}^n} + \left(\underline{\lambda}_1^n\right)^T \frac{\partial \underline{R}^n}{\partial \underline{w}^n} + \left(\underline{\lambda}_2^n\right)^T \frac{\partial \underline{H}^n}{\partial \underline{w}^n} = \underline{0} \quad \forall n \in [1, N] \tag{4.11}$$

Afterwards, QOI sensitivities can be computed using equation (4.12).

$$\frac{d\overline{J}}{ds} = \sum_{n=1}^{N} \left[\frac{\partial J^n}{\partial s} + \left(\underline{\lambda}_1^n\right)^T \frac{\partial \underline{R}^n}{\partial s} + \left(\underline{\lambda}_2^n\right)^T \frac{\partial \underline{H}^n}{\partial s} \right] \tag{4.12}$$

4.3 Stabilization of transient adjoint solution

Figure 4.1 depicts that, sensitivities of a chaotic dynamical system shows exponential growth when adjoint approach is used. Therefore, it is necessary to investigate engineering solutions to obtain adequately accurate sensitivity information for dynamical chaotic problems.

One of the most accurate methods so far is the Least-Squares-Shadowing (LSS) by Wang [138] which finds a shadowing trajectory to its original trajectory by perturbing initial conditions, thereby having non-exponential growth in the sensitivities. Investigations done by Blonigan et al. [17, 18] and Chater et al. [29, 30] have used LSS method successfully to obtain sensitivities on chaotic flow problems. A serious weakness in this approach is the immense computational effort required to identify shadowing trajectory which disallows this to be used in practical problems of interest in CWE. Ensemble adjoint method proposed by Chandramoorthy et al. [28], Eyink et al. [41], and Lea et al. [82] calculates sensitivities by sample averaging of adjoint outputs in short trajectories. Even though this circumvent the problem of exponential growth in adjoint variables, but obtained sensitivities are sensitive towards number of sampling trajectories. Furthermore, the convergence of this method also proven to be not practical for large systems (refer Eyink et al. [41]), thereby requiring a novel stabilization method for transient adjoint solution. Therefore, following two stabilization methods are carefully developed such that to reduce overall cost of the sensitivity evaluation while being in the acceptable error range of the sensitivity solution.

4.3.1 Time averaged primal

QOIs used in this study are first linearized in spatial and temporal domains. In such a situation, time instantaneous values of QOI (such as drag and lift forces) will be a linear combination of time instantaneous solutions. Therefore, we can interchange the averaging as depicted in equation (4.13) where $\overline{\underline{w}} = \frac{1}{N}\sum_{n=1}^{N} \underline{w}^n$.

$$\overline{J} = J\left(\overline{\underline{w}}, s\right) \tag{4.13}$$

Consequently, the QOI depends on the long time averaged solution (i.e. $\overline{\underline{w}}$), then the adjoint problem is reformulated using long time averaging strong form of equation (4.4) and equation (4.5) followed by discretizing their weak formulations. Long time averaging of equation (4.4) raises the closure problem because of the non-linearity in the convection term. Therefore, two equation turbulence modelling methodologies developed in chapter 2 are used in this adjoint stabilization method. Time averaged $\overline{\underline{u}}, \overline{p}$ is used as input for two equation turbulence model's scalar transport equations to obtain their respective turbulence quantities. This results in a steady adjoint formulation as depicted in equation (4.14) where $\overline{\underline{\lambda}}$ is the Lagrange multiplier which is solved for the long time averaged and discretized equality constraint of the residual $\overline{\underline{R}}$.

$$\frac{\partial \overline{J}}{\partial \overline{\underline{w}}} + \overline{\underline{\lambda}}^T \frac{\partial \overline{\underline{R}}}{\partial \overline{\underline{w}}} = \underline{0} \tag{4.14}$$

Afterwards, equation (4.15) can be used to compute long time averaged QOI's sensitivities.

$$\frac{d\overline{J}}{ds} = \frac{\partial \overline{J}}{\partial s} + \overline{\underline{\lambda}}^T \frac{\partial \overline{\underline{R}}}{\partial s} \tag{4.15}$$

Algorithm 7 describes followed aforementioned methodology in obtaining QOI sensitivities. Firstly, transient primal problem is solved for $\underline{u}, p$ by marching forward in time. While forward marching, time averaging of the solution field (i.e. $\overline{\underline{w}}$) is also done for solution fields $\underline{u}$ and p. After solving the transient primal problem, then $\overline{\underline{w}}$ is stored in a file using HDF5 format (The HDF Group [127]) for efficiency in reading and writing. Thereafter, these stored $\overline{\underline{w}}$ values are fed in to another steady state problem defined by desired turbulence model where $\underline{u}$ is taken as a input to solve for turbulence transport quantities using long time averaged equation (4.7). All of these solved quantities are saved using HDF5 format. Finally steady RANS adjoint problem is solved using long time averaged $\underline{u}$, p, and turbulence transport quantities, followed by sensitivity calculation using equation (4.15).

Algorithm 7 Time averaged primal method for sensitivity calculation.

```
$\underline{w}^0 \leftarrow \underline{0}$
$\underline{\dot{w}}^0 \leftarrow \underline{0}$
$\overline{\underline{w}} \leftarrow \underline{0}$
$n \leftarrow 1$
for $n \leq N$ do
    Solve for $\underline{w}^n$ using equation (4.8)
    Calculate $\underline{\dot{w}}^n$ using equation (4.9)
    Update time averaged $\overline{\underline{w}}$ (refer appendix A)
Store $\overline{\underline{w}}$ using HDF5 format
Read $\overline{\underline{w}}$ from stored HDF5 files
Solve two equation turbulence model transport equations using steady
form of equation (4.7)
Store turbulence model transport quantities (i.e. $\overline{\underline{\phi}}$) using HDF5 format
Read $\overline{\underline{w}}$ and $\overline{\underline{\phi}}$ from stored HDF5 files
Solve adjoint RANS problem as depicted in equation (4.14)
Calculate QOI sensitivities using equation (4.15)
```

4.3.2 Artificial diffusion

As depicted in figure 4.1, QOI sensitivities calculated via adjoint method exhibit exponential growth. The proposed stabilization methodology in section 4.3.1 lacks the ability to obtain meaningful sensitivities in cases where weighted time average is used in a QOI (e.g. frequency). The exponential growth in figure 4.1 is due to exponential growth of the adjoint solution, thus growth in adjoint energy in L_2 norm. Therefore, this section investigates a methodology to control adjoint energy generation rate by adding controlled artificial diffusion to the adjoint problem, thereby stabilizing adjoint solution and allowing to use it in any QOI.

Adjoint solution energy generation rate

Adjoint energy generation rate is computed using spatially discretized but temporally continuous primal transient problem as depicted in equation (4.16). These residuals are scaled by a constant diagonal $\underline{\mathbf{G}}$ matrix to make residuals to have same dimensions of T^{-1} for all the residuals. Spatially discretized state variables are represented by $\underline{w}$, and $\underline{s}$ represents shape parameters.

$$\begin{aligned}\underline{\tilde{R}} &= \underline{\mathbf{G}}\left[\underline{F}\left(\underline{w},\underline{s}\right)-\underline{\mathbf{C}}\left(\underline{w},\underline{s}\right)\underline{w}-\underline{\mathbf{M}}\left(\underline{w},\underline{s}\right)\underline{\dot{w}}\right]\\ &= \underline{\mathbf{G}}\left[\underline{\tilde{F}}\left(\underline{w},\underline{s}\right)-\underline{\mathbf{M}}\left(\underline{w},\underline{s}\right)\underline{\dot{w}}\right]\end{aligned} \tag{4.16}$$

Likewise, spatially discretized, but temporally continuous considered QOI also used in this as depicted in equation (4.17).

$$\overline{J} = \int_0^T J\left(\underline{w},\underline{s}\right)dt \tag{4.17}$$

Afterwards, Lagrangian is formulated as depicted in equation (4.18)

$$L = \int_0^T \left[J\left(\underline{w},\underline{s}\right)+\underline{\tilde{\lambda}}^T\,\mathbf{G}\,\underline{\tilde{F}}\left(\underline{w},\underline{s}\right)-\underline{\tilde{\lambda}}^T\,\mathbf{G}\,\mathbf{M}\left(\underline{w},\underline{s}\right)\underline{\dot{w}}\right]dt \tag{4.18}$$

Time derivative of $\underline{w}$ is transferred to time derivative of $\underline{\lambda}$ using integration by parts over the temporal integration as shown in equation (4.19). $\underline{\lambda}\big|_{t=T} = 0$ because transient adjoint problem is calculated by reverse time marching, hence last time step Lagrange multiplier of the residuals is considered as zero. The term $\underline{\dot{\mathbf{M}}}$ also becomes zero because there are no time dependence in the considered mass matrix. Also the initial condition $\underline{w}\big|_{t=0}$ is assumed to be zero, making all the boundary terms in integration by parts to vanish. For brevity, the terms are shortened without indicating their input parameters here onwards.

$$L = \int_0^T \left[J+\underline{\tilde{\lambda}}^T\,\mathbf{G}\,\underline{\tilde{F}}+\underline{\dot{\tilde{\lambda}}}^T\,\mathbf{G}\,\mathbf{M}\,\underline{w}\right]dt \tag{4.19}$$

Now partial derivatives with respect to $\underline{w}$ is taken to formulate adjoint problem as depicted in equation (4.20). $\underline{\tilde{\mathbf{D}}}$ is the introduced artificial diffusion matrix and ψ is diffusion controlling parameter with dimensions same as kinematic viscosity. $\underline{\mathbf{G}'}$ constant diagonal matrix is used to make the deriving variable $\underline{w}$ dimensionless.

$$\frac{\partial L}{\partial \underline{w}} = \int_0^T \left[\frac{\partial J}{\partial \underline{w}}\underline{\mathbf{G}'}+\underline{\tilde{\lambda}}^T\,\underline{\mathbf{G}}\frac{\partial \underline{\tilde{F}}}{\partial \underline{w}}\underline{\mathbf{G}'}-\psi\underline{\tilde{\lambda}}^T\,\underline{\tilde{\mathbf{D}}}+\underline{\dot{\tilde{\lambda}}}^T\,\underline{\mathbf{G}}\left(\frac{\partial \underline{\mathbf{M}}}{\partial \underline{w}}\underline{w}+\underline{\mathbf{M}}\right)\underline{\mathbf{G}'}\right]$$

$$\tag{4.20}$$

In order to make $\underline{\tilde{\mathbf{D}}}$ diffusion matrix positive definite, semi-positive definite $\underline{\mathbf{D}}$ matrix is added with an identity matrix multiplied by $\eta > 0$ as depicted in equation (4.21).

$$\underline{\tilde{\mathbf{D}}} = \underline{\mathbf{D}} + \eta \left\| \underline{\mathbf{D}} \right\|_{fro} \underline{\mathbf{I}} \tag{4.21}$$

Finally equation (4.22) is used to calculate adjoint energy generation rate (negative because of the reverse time marching property of transient adjoint problems).

$$\begin{aligned} \frac{dE}{dt} &= -\frac{1}{2}\frac{\partial}{\partial t}\left(\underline{\tilde{\lambda}}^T \underline{\tilde{\lambda}}\right) \\ &= \left[\frac{\partial J}{\partial \underline{w}}\underline{\mathbf{G}}' + \underline{\tilde{\lambda}}^T \underline{\mathbf{G}} \frac{\partial \underline{\tilde{F}}}{\partial \underline{w}} \underline{\mathbf{G}}' - \psi \underline{\tilde{\lambda}}^T \underline{\tilde{\mathbf{D}}}\right]\left[\underline{\mathbf{G}}\left(\frac{\partial \underline{\mathbf{M}}}{\partial \underline{w}} \underline{w} + \underline{\mathbf{M}}\right)\underline{\mathbf{G}}'\right]^{-1} \underline{\tilde{\lambda}} \end{aligned} \tag{4.22}$$

One main limitation of adjoint energy generation rate computed by equation (4.22) is, the $\frac{\partial \underline{\mathbf{M}}}{\partial \underline{w}} \underline{w} + \underline{\mathbf{M}}$ is not invertible. That is because the incompressible Navier-Stokes equation does not have any contribution from continuity equation to mass matrix hence, producing a singular mass matrix and its derivative. Artificial compressibility can be used in this case to avoid singular mass matrix (Merkle [95]). However, this requires inversion of a matrix which may increase computational cost of adjoint evaluation. Therefore, instead of calculating actual adjoint energy generation rate, approximate energy generation rate is calculated as depicted in equation (4.23).

$$\begin{aligned} \frac{dE_m}{dt} &= -\dot{\underline{\tilde{\lambda}}}^T \underline{\mathbf{G}}\left(\frac{\partial \underline{\mathbf{M}}}{\partial \underline{w}} \underline{w} + \underline{\mathbf{M}}\right)\underline{\mathbf{G}}' \underline{\tilde{\lambda}} \\ &= \frac{\partial J}{\partial \underline{w}} \underline{\mathbf{G}}' \underline{\tilde{\lambda}} + \underline{\tilde{\lambda}}^T \underline{\mathbf{G}} \frac{\partial \underline{\tilde{F}}}{\partial \underline{w}} \underline{\mathbf{G}}' \underline{\tilde{\lambda}} - \psi \underline{\tilde{\lambda}}^T \underline{\tilde{\mathbf{D}}} \underline{\tilde{\lambda}} \end{aligned} \tag{4.23}$$

If $\dot{E}_m$ is positive, it indicates energy growth in reverse time marching adjoint problem. Therefore, if inequality shown in equation (4.24) is satisfied in temporal integration domain, then it should stabilize the adjoint solution.

Equation (4.24) disregards term $\frac{\partial J}{\partial \underline{w}}\underline{\tilde{\lambda}}$ because, adjoint solution exhibits exponential growth due to physical phenomena which is governed by $\underline{R}$.

$$\psi \underline{\tilde{\lambda}}^T \underline{\tilde{\mathbf{D}}} \underline{\tilde{\lambda}} \geq \left|\underline{\tilde{\lambda}}\right| \left| \underline{\mathbf{G}} \frac{\partial \underline{\tilde{F}}}{\partial \underline{w}} \underline{\mathbf{G}}' \right| \left|\underline{\tilde{\lambda}}\right| \geq \underline{\tilde{\lambda}}^T \underline{\mathbf{G}} \frac{\partial \underline{\tilde{F}}}{\partial \underline{w}} \underline{\mathbf{G}}' \underline{\tilde{\lambda}} \tag{4.24}$$

If σ_i are the singular values of matrix $\underline{\mathbf{G}}\frac{\partial \underline{\tilde{F}}}{\partial \underline{w}}\underline{\mathbf{G}}'$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N$ and Ψ_i are eigen values of $\underline{\tilde{\mathbf{D}}}$ where $\Psi_1 \geq \Psi_2 \geq \cdots \geq \Psi_N$, then equation (4.24) can be simplified as shown in equation (4.25).

$$\psi \geq \frac{\sigma_1}{\Psi_N} = \frac{\sigma_1}{\eta \left\|\underline{\mathbf{D}}\right\|_{fro}} \tag{4.25}$$

Finally, the stabilized adjoint problem is shown in equation (4.26) where θ_{sc} is a user defined stabilization coefficient used to control added artificial diffusion.

$$\begin{aligned} \frac{\partial J}{\partial \underline{w}} + \underline{\lambda}^T \frac{\partial \underline{\tilde{F}}}{\partial \underline{w}} - \theta_{sc}\sigma_1 \underline{\lambda}^T \underline{\mathbf{G}}^{-1} \left(\frac{1}{\eta} \frac{\underline{\mathbf{D}}}{\left\|\underline{\mathbf{D}}\right\|_{fro}} + \underline{\mathbf{I}} \right) \left(\underline{\mathbf{G}}'\right)^{-1} \\ = -\underline{\dot{\lambda}}^T \left(\frac{\partial \underline{\mathbf{M}}}{\partial \underline{w}} \underline{w} + \underline{\mathbf{M}} \right) \end{aligned} \tag{4.26}$$

One key parameter in this method is θ_{sc} which controls relative amount of added artificial diffusion to the aforementioned adjoint problem, hence it is a controller which affects the accuracy of the adjoint solution. Therefore, in order to determine most suitable θ_{sc}, a modified bisection method is utilized. In this bisection method, first slope (i.e. ξ) is calculated by least squares fitting of adjoint shape sensitivities' L_2 norm as depicted in equation (4.27).

$$\begin{gathered} \xi = \frac{\sum_{n=n_1}^{n_2} \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\sum_{n=n_1}^{n_2} \left(x_i - \bar{x}\right)} \\ \textit{where} \\ x_i = \Delta t\, n_i \quad \textit{and} \quad y_i = \left| \frac{dJ}{d\underline{s}}^{n_i} \right|_{L2} \qquad \forall n_i \in \left[n_1, n_2\right] \end{gathered} \tag{4.27}$$

In here, n_1 and n_2 are user defined step indices where it make sense to compute the slope of the $\left|\frac{dJ}{d\underline{s}}^{n_i}\right|_{L2}$. Once ξ is calculated, it is checked against user defined allowed maximum slope ξ_{max}.

Algorithm 8 Adjoint solution stabilization coefficient calculation.

```
procedure ComputeStabilizationCoefficient($a$, $b$, $n_1$, $n_2$, $\delta$, $M$)
    $i \leftarrow 1$
    $\underline{\theta} \leftarrow [a, \frac{a+b}{2}, b]$                ▷ Assumes $a \le b$
    while $\theta_2 - \theta_0 \ge \delta$ or $i \le M$ do
        $i \leftarrow i + 1$
        for $j \in [0,1,2]$ do
            $\theta_{SC} \leftarrow \theta_j$
            If not solved, then solve adjoint problem (refer equation (4.26))
            Calculate $\xi_j$ using equation (4.27)
        if $|\xi_j| > \xi_{max} \forall \xi_j \in \underline{\xi}$ then
            $\underline{\theta} \leftarrow \left[\theta_2, 1.5\theta_2, 2\theta_2\right]$
        else
            if $|\xi_j| < \xi_{max} \forall \xi_j \in \underline{\xi}$ then
                $\underline{\theta} \leftarrow \left[\frac{\theta_0}{2}, \frac{3\theta_0}{4}, \theta_0\right]$
            else
                $\theta_0 \leftarrow \theta_{k-1} \left\{k \in [1,2] : |\xi_k| < \xi_{max}\right\}$
                $\theta_2 \leftarrow \theta_{k+1} \left\{k \in [1,0] : |\xi_k| > \xi_{max}\right\}$
                $\theta_1 \leftarrow \frac{\theta_0+\theta_2}{2}$
    return $\theta_2$
```

Algorithm 8 depicts the procedure followed in computing optimal θ_{SC}. a and b are initial bounds for θ_{SC} where $a \le b$. QOI sensitivities' L_2 norm is calculated between n_1 and n_2 time step indices. δ is the tolerance for convergence of θ_{SC} and M is the maximum number of iterations used to find optimal θ_{SC}.

Adding artificial diffusion will diffuse the adjoint solution. Therefore having large enough θ_{SC} will make the adjoint solution stable if it is unstable or make the adjoint solution diffused if it is already stable without having any positive Lyapunov exponents. Reducing it will make adjoint solution unstable if there are any positive Lyapunov exponents, or make adjoint solution more consistent with the primal solution, hence making adjoint solution more accurate. Therefore, If initial $\theta_{SC} \in [a, b]$ is not able to stabilize the adjoint solution then, the initial range is expanded to $\theta_{SC} \in [b, 1.5b, 2b]$. On the other hand, if initial range produces stabilized adjoint solution $\forall \theta_{SC} \in [a, b]$, then the range is constrained to $\theta_{SC} \in \left[\frac{a}{2}, \frac{3a}{4}, a\right]$. If a $\theta_{SC} \in [a, b]$ is found with unstable and stable adjoint solutions, then the range is bisected for the next iteration. This process is repeated until the given δ tolerance between $[a, b]$ range is achieved or, maximum number of iterations (i.e. M) is reached.

Even though it seems that algorithm 8 has more inputs to determine one undefined θ_{SC} parameter, a, b, δ, M are input parameters which can be considered as case-independent. a, b only provides an initial boundary for bisection, if it is closer to the optimal θ_{SC} then there will be faster algorithm convergence. δ and M control the accuracy of the algorithm. Hence the only parameters which are case dependent are n_1 and n_2. They determine which time span to be used for slope calculation explained in equation (4.27). Another important fact to note in this algorithm is the complexity of convergence. If an initial guess is provided which produces stable and unstable adjoint solutions at their bounds, then $\log_2\left(\frac{a-b}{\delta}\right)+2$ adjoint solutions are required to compute optimal θ_{SC}.

4.4 Application to chaotic flow problem

Until now we have discussed stabilization for shape sensitivities computed for transient chaotic problems. This section introduces and discusses applicability of each stabilization method and significance of parameters using a 2D example.

4.4.1 Chaotic flow problem definition

This problem is adapted from work done by Schäfer et al. [116]. The original problem is designed to investigate flow behavior around a cylinder with low Re. In this experiment, $Re = 1 \times 10^4$ is chosen by changing the average inlet velocity. Figure 4.2 illustrates basic configuration used in this problem.

This problem is solved using Variational Multi-Scale (VMS) stabilized incompressible NS equations in 2D developed by Cotela Dalmau et al. [35].

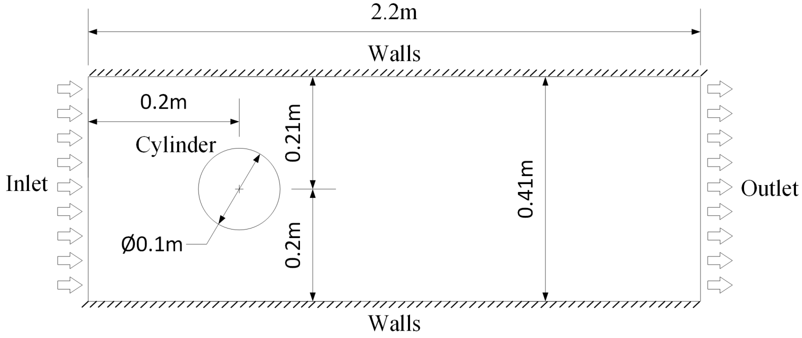


Figure 4.2: 2D channel flow configuration.

$\rho = 1.0\,kg\,m^{-3}$ (i.e. density) and $\nu = 1.0 \times 10^{-5}\,m^2 s^{-1}$ (i.e. kinematic viscosity) are used as flow properties. Outlet is applied with $0.0\,Pa$ Dirichlet boundary condition. Walls and cylinder surfaces are applied with no-slip (i.e. $0.0\,m s^{-1}$) velocity Dirichlet boundary condition. Inlet is applied with a parabolic inlet using equation (4.28) with $u_m = 1.0\,m s^{-1}$ (i.e. mean velocity) which corresponds to $Re = 1 \times 10^4$.

$$u = \frac{4u_m y\left(0.41 - y\right)}{0.41^2} \tag{4.28}$$

Drag over the cylinder surface is chosen as QOI for this problem calculated using equation (4.29). η represents the outward surface normals, $\underline{m}$ represents drag force direction.

$$\overline{J} = \frac{1}{N}\sum_{n=1}^{N}\left[\int_{\Gamma_{cylinder}} \underline{\boldsymbol{\sigma}}\left(\underline{w}^n, \underline{s}\right) \cdot \underline{\eta} \cdot \underline{m}\, d\Gamma\right] \tag{4.29}$$

4.4.2 Verification methodology

Verification is carried out for the proposed stabilized adjoint problem using finite difference method as depicted in equation (3.11) where nodal coordinates are perturbed to obtain finite difference sensitivities. However, spatial discretizations obtained by perturbing nodal coordinates suffer from mesh irregularity and non-smooth shape derivatives (Chenais [31], Le et al. [81], and Mohammadi et al. [96]). Therefore vertex morphing is used to smoothen the sensitivities (Bletzinger [16]).

Vertex morphing method calculates sensitivities with respect to control points (i.e. $\underline{\tilde{s}}$) designed to obtain a smooth surface for shape parameters (i.e. $\underline{s}$). Smoothing is performed by a transformation matrix (i.e. $\underline{\mathbf{A}}_r$) which is dependent on the filter radius of r. Equation (4.30) illustrates the relationship between control points and the shape parameters.

$$\underline{\tilde{s}} = \underline{\mathbf{A}}_r \underline{s} \tag{4.30}$$

Hence, the smoothened shape sensitivities can be computed by using equation (4.31) where N are number of time steps, and $\frac{dJ}{d\underline{s}}^n$ is the n^{th} time step's sensitivity with respect to shape parameters.

$$\frac{d\overline{J}}{d\underline{\tilde{s}}} = \underline{\mathbf{A}}_r^T \sum_{n=1}^{N} \frac{dJ}{d\underline{s}}^n \tag{4.31}$$

In order to obtain analytical finite difference sensitivities, equation (3.11) needs to be modified to obtain sensitivities with respect to control parameters as depicted in equation (4.32).

$$\frac{d\overline{J}}{d\underline{\tilde{s}}} \approx \frac{\sum_{n=1}^{N} \left[J\left(\underline{w}^n + \delta \underline{w}^n, \underline{s} + \underline{\mathbf{A}}_r^{-1} \delta \underline{\tilde{s}}\right) - J\left(\underline{w}^n, \underline{s}\right)\right]}{\delta \underline{\tilde{s}}} \tag{4.32}$$

Where $\delta \underline{w}^n$ is the solution field obtained using the perturbed control points $\delta \underline{\tilde{s}}$ as depicted in equation (4.33).

$$\delta \underline{\tilde{s}} = \left\{ \tilde{s}_i \in \mathbb{R} : \max_{\forall s_i \in \underline{s}} s_i = \delta s \right\} \tag{4.33}$$

4.4.3 Verification results

Verification requires analytical solutions obtained via equation (4.32) as explained in section 4.4.2. Figure 4.3 illustrates considered node ids and their nodal positions used in this verification study. These nodes are carefully chosen after observing the flow patterns in the problem with $Re = 1 \times 10^4$. Node 1 and node 2 are chosen near the separation point, node 3 is chosen

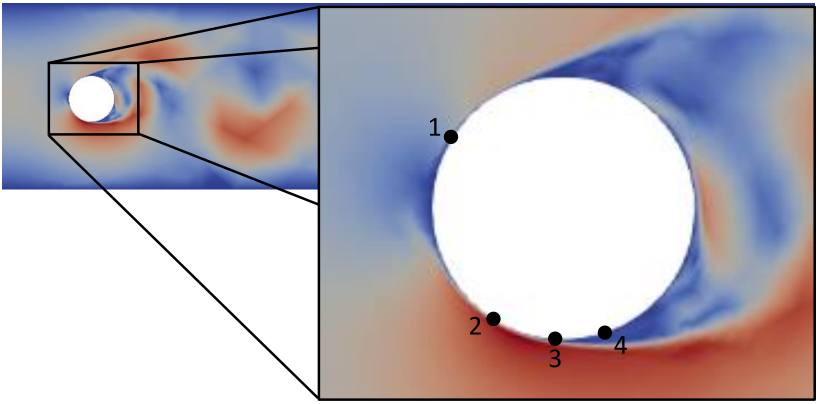


Figure 4.3: Node ids used in verification of sensitivities.

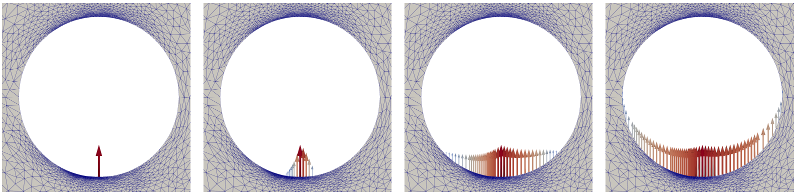
after the separation point at lower most point on the cylinder surface, and node 4 is chosen in the back region of the cylinder.

Nodal perturbations alone make mesh-irregularities and non-smooth shape derivatives as explained in section 4.4.2. The proposed vertex morphing methodology has the filter radius and perturbation size as input parameters. Therefore, firstly, a filter radius study is done and then followed by perturbation size to identify the effect of them towards analytical finite difference sensitivities.

Filter radius study

Filter radius study is carried out with the same problem but with lower $Re = 1\times10^2$. Lower Re is used to fool proof the methodology used to obtain the analytical finite difference sensitivities without having exponentially growing adjoint solutions. Figure 4.4 illustrates nodal perturbations at node 3 obtained with $\delta s = 1 \times 10^{-8}\ m$ in the radial direction for different filter radii. It can be observed at node 3, $r = 1 \times 10^{-4}\ m$ represents perturbations without vertex morphing, whereas all other radii illustrate vertex morphed perturbations.

Figure 4.5 illustrates Finite Difference (FD) sensitivities and Adjoint (AD) sensitivities computed for different filter radii at node with id 4. It can be



(a) $r = 1 \times 10^{-4}\,m$ (b) $r = 1 \times 10^{-2}\,m$ (c) $r = 4 \times 10^{-2}\,m$ (d) $r = 8 \times 10^{-2}\,m$

Figure 4.4: Radial vertex morphed perturbations for different r at node 2.

seen that, both FD and AD are agreeing with each other for all r values. Therefore, hereafter $r = 8 \times 10^{-2}$ is used as the filter radii.

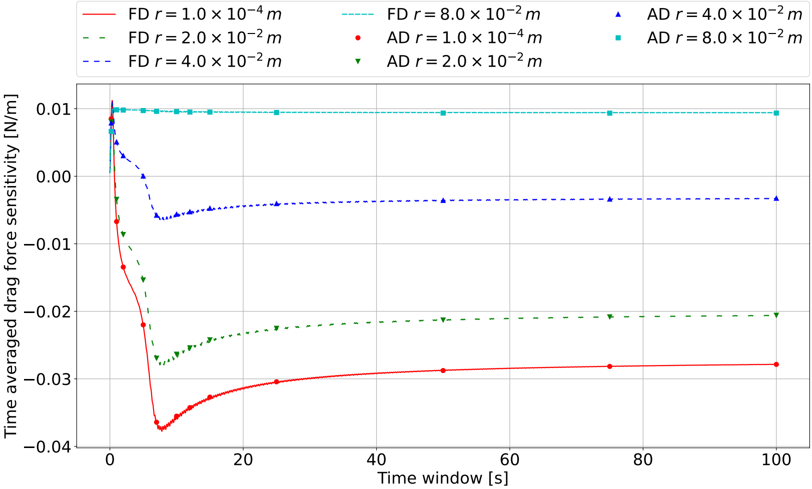


Figure 4.5: Finite difference and adjoint sensitivities for different r with $Re = 1 \times 10^2$ at node 4.

Perturbation size study

Having defined the filter radii, it is also important to identify correct perturbation size to be used to compute FD sensitivities. Figure 4.6 illustrates time averaged drag forces obtained for different radii perturbations sizes. It can be shown that, time averaged drag values show noise for smaller perturbations sizes as expected. Larger perturbations sizes may go beyond the linearization region for the FD sensitivity approximation. Therefore, $\delta s \in \left[2.5 \times 10^{-4}, 4.5 \times 10^{-3}\right]$ is selected as the linear region in calculating polynomial fit to obtain FD sensitivities for the node 1.

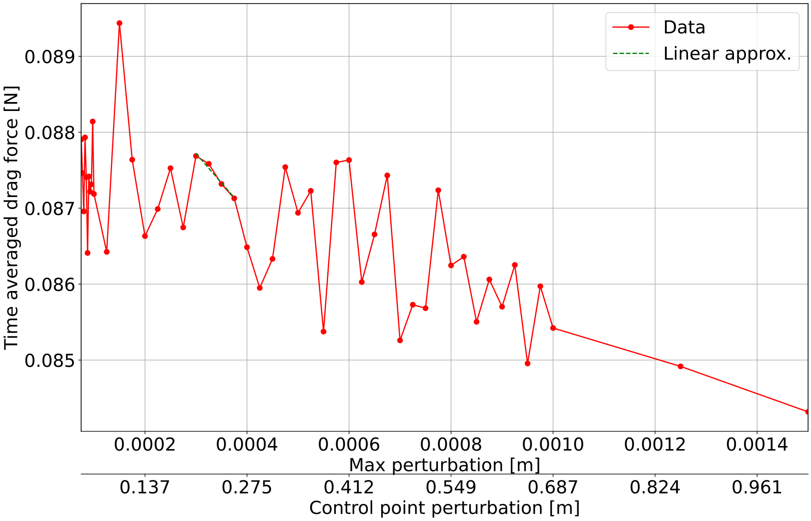


Figure 4.6: Drag force for different radial perturbation sizes with $r = 8 \times 10^{-2}\ m$ and $Re = 1 \times 10^{4}$ at node 3.

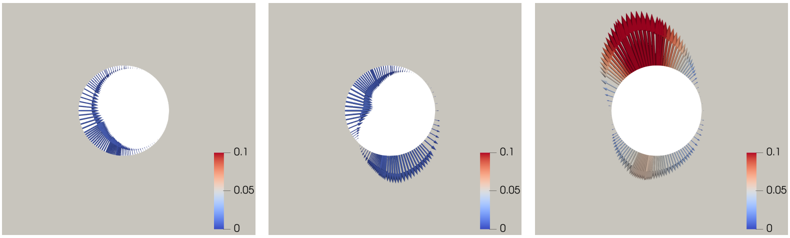
Likewise, time averaged drag force linear regions are identified for all the nodes considered in this verification study and then polynomial fitting is used to calculate the respective FD sensitivities as shown in table 4.1.

Table 4.1: Nodal time averaged drag radial sensitivity reference values.

Node id	Time averaged drag sensitivity $\left[N m^{-1}\right]$
1	9.0696×10^{-3}
2	1.3977×10^{-3}
3	1.1301×10^{-2}
4	9.0091×10^{-3}

4.4.4 Time averaged primal adjoint sensitivities results

Firstly, results of the adjoint sensitivities computed using time averaged primal (see section 4.3.1) are presented. Three different turbulence models are used in this study namely $k-\epsilon$, $k-\omega$ and $k-\omega-sst$ to obtain time averaged solutions. Figure 4.7 illustrates vertex morphed sensitivities calculated with time averaged solution and different RANS turbulence models. The magnitude of each sensitivity arrow is adjusted separately to show their directions, but the same color graph is used in all three cases for comparison.



(a) $k-\epsilon$ (b) $k-\omega$ (c) $k-\omega-sst$

Figure 4.7: Vertex morphed adjoint drag force sensitivities $\left[N m^{-1}\right]$ calculated with time averaged solution and RANS turbulence models.

As depicted in figure 4.7, vertex morphed drag force sensitivities obtained using three different RANS turbulence models show different sensitivity values. Drag force sensitivity directions and magnitudes are changing with the change of RANS turbulence model. Therefore, table 4.2 is used to compare nodal adjoint sensitivities with the reference nodes depicted in figure 4.3.

Table 4.2: Nodal time averaged drag radial sensitivities calculated using time averaged solution with different RANS turbulence models.

Node id	Time averaged drag sensitivity $\left[N m^{-1}\right]$			
	Ref	$k-\epsilon$	$k-\omega$	$k-\omega-sst$
1	9.0696×10^{-3}	-6.5905×10^{-3}	-9.6931×10^{-3}	3.0349×10^{-2}
2	1.3977×10^{-3}	-8.1326×10^{-3}	8.5208×10^{-5}	3.1939×10^{-2}
3	1.1301×10^{-2}	-4.0183×10^{-3}	1.0027×10^{-2}	5.1514×10^{-2}
4	9.0091×10^{-3}	-1.6871×10^{-3}	9.1547×10^{-3}	3.6037×10^{-2}

The drag force sensitivities obtained using RANS $k-\epsilon$ turbulence model show all negative values as depicted in table 4.2 which are in the opposite direction to the reference sensitivities' directions. In the case of RANS $k-\omega$, the direction of sensitivities agrees for node ids 2, 3, and 4. However it does not agree for node 1 which is near the flow separation point in the upper half of the cylinder. The order of magnitude of all the nodes also does not match well with the reference values. On the other hand, the drag sensitivities' directions obtained using RANS $k-\omega-sst$ turbulence model agrees with the reference values. Despite the correct directionality in computed drag sensitivities, the magnitudes do not match well with the reference values. However, RANS $k-\omega-sst$ turbulence model shows better results than RANS $k-\epsilon$ or $k-\omega$. This indicates that, having a more accurate RANS turbulence models (such as Reynolds stress models, 7 equation models, machine learning) may be able to achieve more accurate adjoint drag sensitivity results.

4.4.5 Artificial diffusion results

This section focuses on presenting results of stabilized adjoint using added artificial diffusion (refer section 4.3.2) for the same verification problem explained in section 4.4.1. Influence of θ_{SC} coefficient is presented in the following section.

Effect of stabilization coefficient

Stabilization coefficient (i.e. θ_{SC}) controls relative magnitude of localized artificial diffusion added to the system which makes adjoint problem inconsistent with the primal problem. Despite this, it is critical to add artificial diffusion to stabilize adjoint solution from having exponential growth due to butterfly effect. Therefore, it is critical to add least amount of artificial diffusion just enough to stabilize adjoint solution, thus having least adverse impact on the accuracy and the consistency between adjoint and primal problems.

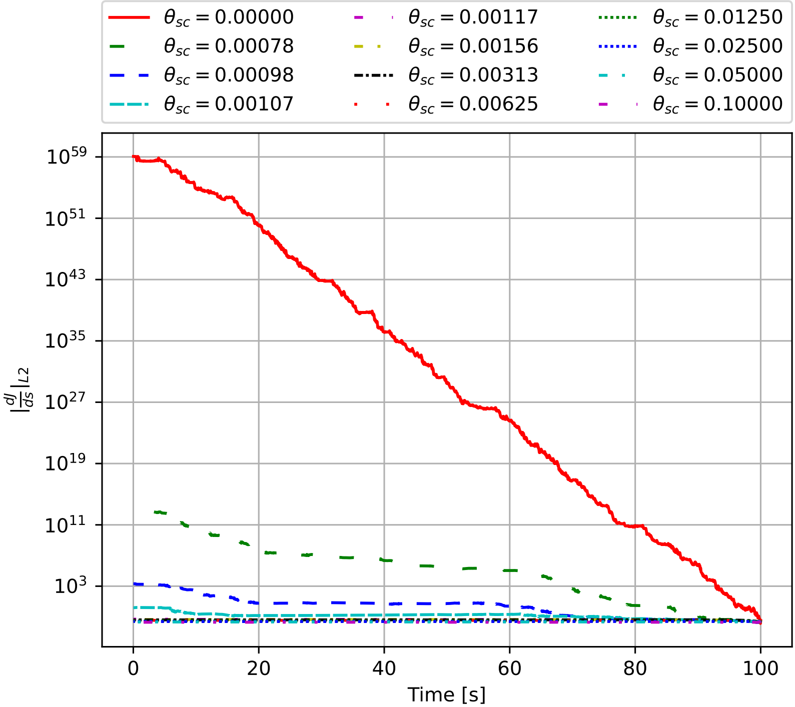


Figure 4.8: Shape sensitivity $L2$ norm variation for different θ_{SC} values used in verification case.

Figure 4.8 illustrates L_2 norm of shape sensitivity variation with time for different θ_{SC} stabilization coefficients obtained using algorithm 8 with $\delta = 1 \times 10^{-4}$, $a = 0.0$, and $b = 0.1$. The plateau range is chosen to be $[n_1, n_2] = [0.0, 50.0]$. It is chosen after conducting an adjoint problem solve with $\theta_{SC} = 1.0$ and then observing the plateau in that adjoint solution. The optimal is found to be $\theta_{SC} = 1.07 \times 10^{-3}$. Figure 4.9 illustrates vertex morphed shape sensitivities for the optimal θ_{SC}. In comparison to results shown in figure 4.7(c), figure 4.9 also shows same pattern in nodal radial shape sensitivities near top and bottom regions of the cylinder. However, frontal and back regions' nodal shape sensitivities differ.

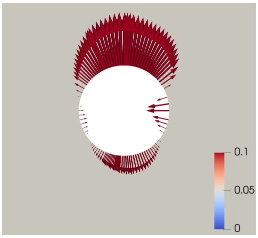


Figure 4.9: Vertex morphed sensitivities calculated from stabilized adjoint solution by adding artificial diffusion.

In order to further quantify the accuracy,table 4.3 shows nodal radial vertex morphed shape sensitivities for selected nodes in the verification case compared against the reference values obtained from finite differencing approach. It can be observed that, all the nodal shape sensitivities from the solution with added artificial diffusion has the same direction as in the reference solution, and their solution order of magnitude also well agrees with the reference solution. Specifically nodes with ids 2, 3, and 4 depicts the least errors among all.

From the results shown in table 4.3, it is evident that adding controlled artificial diffusion is able to produce meaningful nodal shape sensitivities and circumvent exponential growth in adjoint solution.

Table 4.3: Nodal time averaged drag radial sensitivities calculated using added artificial diffusion solution.

Node id	Time averaged drag sensitivity $\left[N m^{-1}\right]$	
	Ref	$\theta_{SC} = 1.07 \times 10^{-3}$
1	9.0696×10^{-3}	3.9364×10^{-3}
2	1.3977×10^{-3}	1.8643×10^{-3}
3	1.1301×10^{-2}	1.0377×10^{-2}
4	9.0091×10^{-3}	9.4237×10^{-3}

4.5 Frequency domain optimization

From the previous discussion, it can be seen that adjoint solution stabilization with added artificial diffusion provides meaningful nodal shape sensitivities. In order to further investigate this method, nodal sensitivities computed via added artificial diffusion are used in frequency domain shape optimization problem, where it is critical to have time dependent QOI, hence transient sensitivities are required with stabilization. Therefore, stabilization with added controlled artificial diffusion stabilization method is used.

4.5.1 Primal problem definition

Analyzing vortex shedding frequency is very often carried out when tall buildings, bridges are constructed (Giosan et al. [52] and Irwin [67]). One of the popular examples to investigate vortex shedding is the flow around a circular cylinder example which is studied experimentally and numerically by Luo et al. [86], Norberg [102], and Oertel Jr [103]. BARC is one of the heavily used structural configurations to analyze wind effects on structures. In this section, both flow over a circular cylinder with $Re = 100$ and BARC problem explained in section 2.5.1 are used in this optimization problem with $k-\omega-sst$ as the turbulence model. In the former case, the same configuration as in BARC is used (refer figure 2.34) where the rectangle is replaced with the cylinder and H is the diameter (i.e. $D = 0.1\,m$) of the cylinder. A mesh convergence study is carried out to obtain the most efficient mesh while having acceptable accuracy. The table 4.4 depicts drag coefficient (i.e. C_D), lift coefficient (i.e. C_L) and Strouhal number (i.e. St) comparison of the final mesh with the literature for the flow over the cylinder case with $Re = 100$.

Table 4.4: Comparison of flow over the cylinder statistical parameters.

Source	C_D	C_L	St
Wang et al. [139]	1.379	0.357	0.17
Present study	1.26	0.29	0.14

4.5.2 Vortex shedding frequency

Vortex shedding frequency is computed by conducting Discrete Fourier Transform (DFT) of the time dependent lift force. Equation (4.34) is used to transform time dependent lift forces to frequency domain where N is the total number of time steps, F^n is the n^{th} time step lift force over the rectangular cylinder in BARC problem.

$$X_k = \sum_{n=0}^{N-1} F^n \left[\cos\left(\frac{2\pi}{N} kn\right) - i \sin\left(\frac{2\pi}{N} kn\right) \right] \tag{4.34}$$

However, this time dependent lift forces are not periodic in time because high $Re = 5 \times 10^4$ in the flow produces a chaotic flow. Therefore it can produce spectral leakage in the frequency domain, thus poor quality results in frequency domain. In order to mitigate that, Hann (Essenwanger [40]) function is used in windowing as depicted in equation (4.35) where K is the number of windowing time steps where $K \leq N$.

$$w^n = \begin{cases} \frac{1}{2}\left[1 - \cos\left(\frac{2\pi(n-N+K)}{K}\right)\right] & if \qquad n \geq N-K \\ 0 & else \end{cases} \tag{4.35}$$

Next, frequency distribution of the windowed time dependent lift force is calculated using equation (4.36).

$$\tilde{X}_k = \sum_{n=0}^{N-1} w^n F^n \left[\cos\left(\frac{2\pi}{N} kn\right) - i \sin\left(\frac{2\pi}{N} kn\right) \right] \tag{4.36}$$

Finally each frequencies' amplitudes are calculated to estimate each frequencies' relative importance which is used as the QOI for this optimization problem as depicted in equation (4.37) where k_1 and k_2 are user defined frequency range indices of interest for optimization problem.

$$J_{freq} = \max_{k \in [k_1, k_2]} \left| \tilde{X}_k \right|^2 \tag{4.37}$$

4.5.3 Adjoint problem definition

The QOI given in equation (4.37) involves taking magnitude of a complex number. Therefore, it is required to calculate nodal shape sensitivities of real and imaginary contributions of the frequency amplitude separately. Hence two adjoint problems are solved for each real and imaginary component as their QOI as depicted in equation (4.38a) and equation (4.38b) using the same primal solution to compute their respective nodal shape sensitivities. K for Hann windowing is chosen such that Hann windowing length corresponds to $3.0\,s$. It is chosen after following a Hann windowing length study.

$$real\left(\tilde{X}_k\right) = \sum_{n=0}^{N-1} w^n F^n \cos\left(\frac{2\pi}{N} kn\right) \tag{4.38a}$$

$$imag\left(\tilde{X}_k\right) = \sum_{n=0}^{N-1} w^n F^n \sin\left(\frac{2\pi}{N} kn\right) \tag{4.38b}$$

The flow over the cylinder problem is considered to be a periodic flow problem which does not require stabilization in adjoint solution. To identify whether the proposed stabilization method correctly detects whether the given problem requires adjoint stabilization or not, this problem is also solved using the same adjoint stabilization algorithm explained in algorithm 8 with $[a, b] = [0.0, 1.0]$. n_1 and n_2 time step indices are chosen such that the slope is calculated between $t \in [0.0\,s, 4.5\,s]$. $\xi_{max} = 50.0$ and $\delta = 1 \times 10^{-4}$ are chosen for the refered algorithm.

The BARC problem is considered to be high $Re = 5 \times 10^4$ chaotic flow problem which requires stabilization in adjoint solution. Therefore, in order to identify parameters required for algorithm 8, primal problem is solved once, and then the adjoint problem with equation (4.38a) as QOI and with $\theta_{SC} = 20.0$ is solved to identify corresponding parameters required for algorithm 8. n_1 and n_2 time step indices are chosen such that slope is calculated between $t \in [0.0\,s, 4.5\,s]$ as in the flow over the cylinder problem. $\xi_{max} = 50.0$ is chosen after observing the L_2 norm variation of the nodal shape sensitivity. $a = 0.0$, $b = 1.0$, and $\delta = 1 \times 10^{-4}$ are other parameters used for adjoint problem stabilization. Figure 4.10 illustrates results of this preliminary study.

4.5.4 Optimization problem

Now we have defined primal problem and adjoint problem with stabilization. This section focuses on defining optimization problem being solved which is depicted in equation (4.39).

$$
\begin{aligned}
&\min_{\underline{s}} \quad J_{freq}\left(\underline{w}(\underline{s}), \underline{s}\right) \\
&\text{subjected to} \\
&\qquad\qquad \underline{R}^n_{\phi} = \underline{0} \quad \forall \phi \in \left\{u^n, v^n, w^n, p^n, k^n, \omega^n\right\} \quad \forall n \in [1, N] \\
&\qquad\qquad G_{centroid}\left(\underline{s}\right) = 0 \\
&\qquad\qquad G_{volume}\left(\underline{s}\right) = G_{volume}\left(\underline{s}_{initial}\right)
\end{aligned}
\tag{4.39}
$$

The goal of this optimization problem is to obtain novel shapes via shape optimization by removing frequencies in a given range. This is achieved by minimizing J_{freq} in a given frequency range as depicted in equation (4.39). Two geometric constraints namely $G_{centroid} = 0$ (refer equation (3.13)) and $G_{volume} = 0$ are used to avoid rectangular cylinder being moved away from the fluid domain and rectangular cylinder being smaller by reducing the size respectively. Equation (4.40) is used to compute G_{volume} constraint where M being number of discretized elements in the domain and V_i being volume of i^{th} discretized element.

$$
G_{volume} = \sum_{i=1}^{M} V_i^2 \tag{4.40}
$$

In order to be efficient in executing shape optimization problem defined in equation (4.39), primal problem is only run once and the solution is

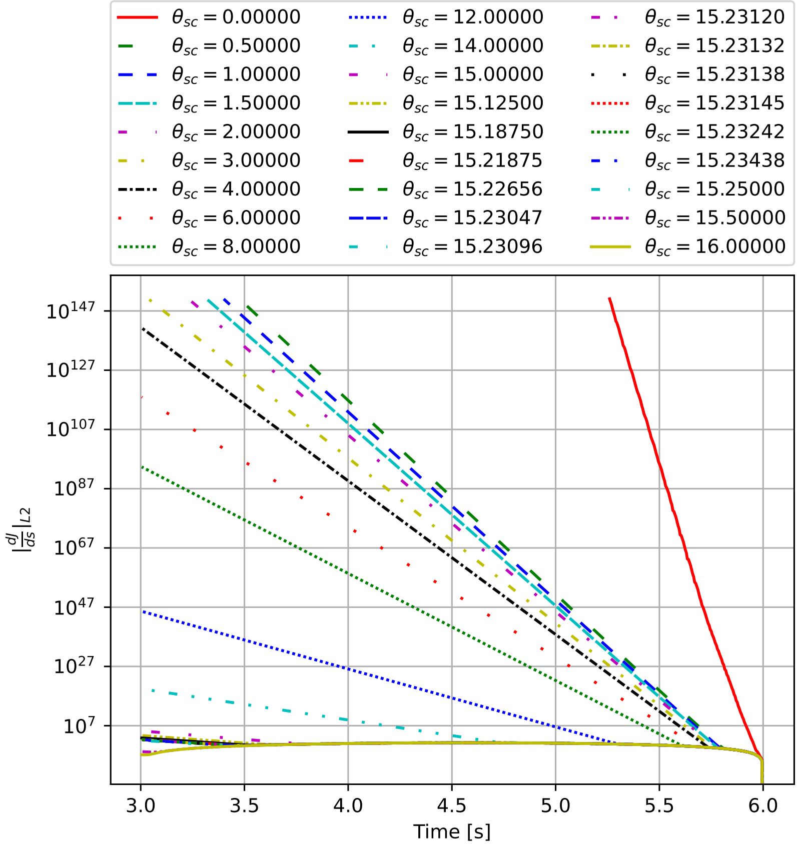


Figure 4.10: Shape sensitivity $L2$ norm variation for different θ_{SC} values used in primal problem in frequency domain optimization.

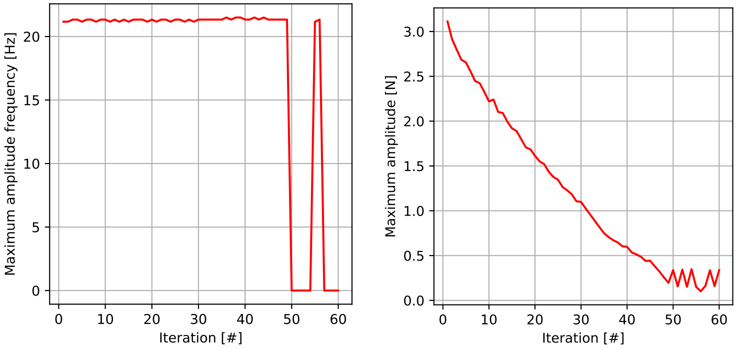
stored using HDF5 check pointing methodology for later use in two adjoint problems. As explained in section 4.5.3, only in the adjoint problem with equation (4.38a) as QOI is used to compute the optimal θ_{SC} using algorithm 8 with coefficients given in section 4.5.3 instead of computing two different optimal θ_{SC} stabilization coefficients for each adjoint problem with QOI equation (4.38a) and equation (4.38b). This is possible because, the observed instability in adjoint solution occurs due to presence of positive Lyapunov coefficients in the primal solution's lift computation. Both adjoint problems use the same time dependent lift forces, hence same stabilization coefficients can be used which make adjoint stabilization coefficient computation efficient.

Optimization procedure is performed using gradient descend algorithm with constant step size. Computed nodal shape sensitivities are vertex morphed to smoothen noisy shape sensitivity field.

4.5.5 Flow over the cylinder results

Until now we have discussed optimization problem definition and its corresponding primal and adjoint problem definitions. The following section will discuss results of the optimization problem of the flow over the cylinder problem. The optimization frequency index range (i.e. $\left[k_1, k_2\right]$) is chosen such that the optimization problem is considering to minimize max amplitude of the frequencies in the range of $[0Hz, 50Hz]$ for the QOI given in equation (4.37).

Figure 4.11 illustrates maximum amplitude in the chosen frequency range (refer figure 4.11(b)) and its corresponding frequency (refer figure 4.11(a)). The frequency with maximum amplitude starts with $21.17Hz$. It can be observed that the proposed shape optimization problem with transient adjoint stabilization clearly reduces maximum amplitude of the frequencies within the given range. Until the design iteration 49, it can be observed that the frequency which corresponds to maximum amplitude varies between $21.17Hz$ and $21.33Hz$. The maximum amplitudes at the design iteration 49 for $0Hz$ and $21.33Hz$ are $0.1193N$ and $0.1953N$. At this design iteration, the amplitudes corresponding to $0Hz$ and $21.33Hz$ have reached similar values. Therefore, after design iteration 49, the optimization procedure is identifying $0Hz$ frequency as the frequency with maximum amplitude and computes sensitivities to reduce the amplitude. This can



(a) Frequency variations

(b) Lift force amplitude variations

Figure 4.11: Maximum amplitude frequency and maximum amplitude variations with design iteration in flow over the cylinder frequency range optimization problem.

be observed at design iteration 50 where $0\,Hz$ and $21.33\,Hz$ amplitudes are at $0.3385\,N$ and $0.1498\,N$. Thereafter, the optimization procedure keeps oscillating between $0\,Hz$ frequency and $21.33\,Hz$ frequency since both the frequencies are having similar low amplitudes. This is clearly evident in oscillations found after design iteration 49 in figure 4.11.

Few of the chosen design iterations' cylinder cross sections and their lift force frequency distributions are illustrated in figure 4.12. The grey cross section in each design iteration refers to initial configuration design surface of the optimization problem. It can be seen that, the proposed shape optimization methodology successfully reduce the frequency range maximum amplitude of this problem. The computed θ_{SC} using transient adjoint stabilization coefficient calculation methodology (refer algorithm 8) is always 0 for all the design iterations in this optimization problem. This indicates that the proposed transient adjoint stabilization methodology is capable of identifying when to add controlled artificial diffusion to stabilize transient adjoint solution. It is also evident from figure 4.12 iteration 49, that both amplitudes corresponding to $0\,Hz$ and $21.33\,Hz$ is becoming closer to each other which makes the QOI to oscillate after 49^{th} design iteration.

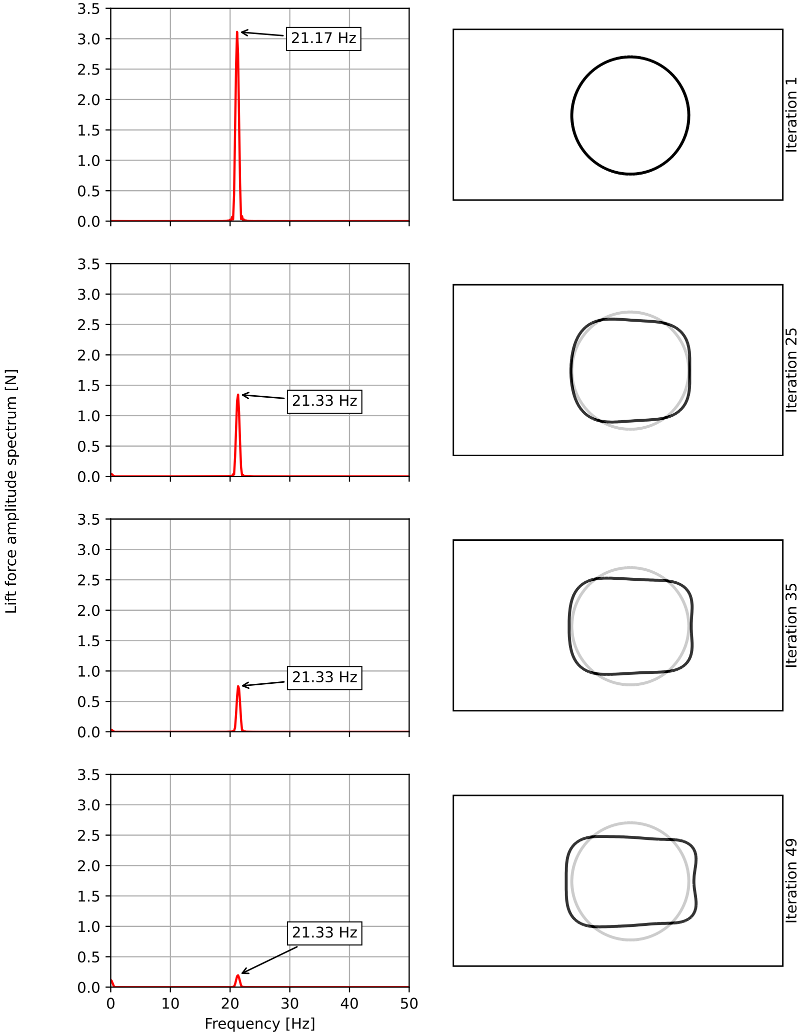


Figure 4.12: Frequency distribution within the interested range and their respective cylinder cross sections for iterations in flow over the cylinder shape optimization problem.

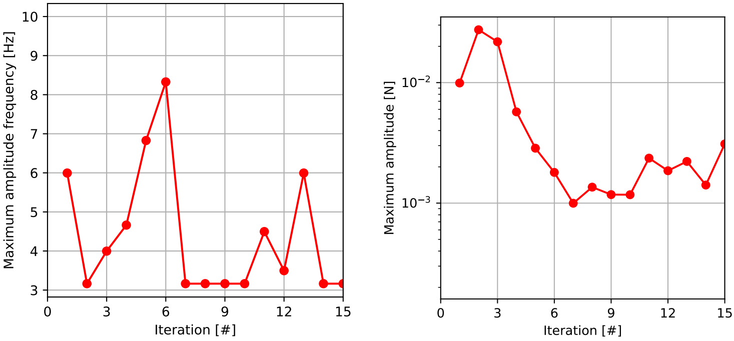
4.5.6 BARC results

Two optimization problems are conducted and results are presented in this section having two different frequency ranges (for corresponding two different frequency index ranges $\left[k_1, k_2\right]$) for minimization of QOI depicted in equation (4.37) used in optimization problem depicted in equation (4.39). First frequency range falls between $[3Hz, 10Hz]$ (i.e. lower frequency range) and second frequency range falls between $[20Hz, 50Hz]$ (i.e. higher frequency range). This allows to observe how different shapes emerge when interested frequency range is changed, thus giving more insight on frequency distribution changes with respect to shape change.

Lower frequency range shape optimization

Figure 4.13(a) illustrates frequency of the maximum amplitude for each design iteration in the low frequency range optimization problem. Figure 4.13(b) shows maximum amplitude for the same design iterations. It can be observed that, the maximum amplitudes in the interested lower frequency range depict a downward trend indicating the proposed optimization procedure is being successful. Hence, this provides clear evidence that, proposed stabilization method with added controlled artificial diffusion is able to produce meaningful nodal shape sensitivities for this optimization problem. Even though maximum amplitude shows a clear decline with increasing iterations, after 7^{th} iteration it shows an oscillatory behavior. This may be due to the fact that, use of a constant step size and constant filter radius may prohibit exploring further optimized designs.

It is observable from figure 4.13(a) that, an increasing trend is present in the frequency corresponding to maximum amplitude (between iteration 1 - 6). This is also in agreement with total energy conservation of the system because evaluated BARC problems are given the same energy input/output and flow conditions for each design iteration. Therefore it requires to increase amplitude of a higher frequency in order to reduce amplitude of a lower frequency in the optimization problem. Evidence from figure 4.13(b) clearly shows that increment in an amplitude of a higher frequency needs to be relatively low than the decrement in an amplitude of a lower frequency which is inline with the energy conservation of the total system. However, a clear drop in frequency corresponding to maximum amplitude is visible after 6^{th} iteration. In order to investigate it further, figure 4.14



(a) Frequency variations

(b) Lift force amplitude variations

Figure 4.13: Maximum amplitude frequency and maximum amplitude variations with design iteration in BARC low frequency range optimization problem.

illustrates frequency distributions and corresponding cross sections for iterations 1, 6, 7, and 15. If observed carefully, cross sections in iteration 6 and iteration 7 do not show significant differences. Even maximum amplitudes do not show a large difference between those two iterations (refer figure 4.13(b)). A possible explanation for this drop is, having two frequencies in the given low frequency range with close enough amplitudes. This can be observed in the frequency distributions for both iterations depicted in figure 4.14. When there are two frequencies which have close enough amplitudes then the proposed QOI in equation (4.37) may switch between them within design iterations causing the sudden drop in the frequency corresponding to maximum amplitude causing the sudden drop in the frequency corresponding to maximum amplitude.

Furthermore, the geometric constraints (i.e. $G_{centroid}$ and G_{volume}) are satisfied with tolerance of 1×10^{-6} in all of the design iterations obtained in solving this optimization problem.

Higher frequency range shape optimization

Lower frequency range shape optimization performed with stabilized adjoint solution exhibits arbitrary BARC cross sections in its design itera-

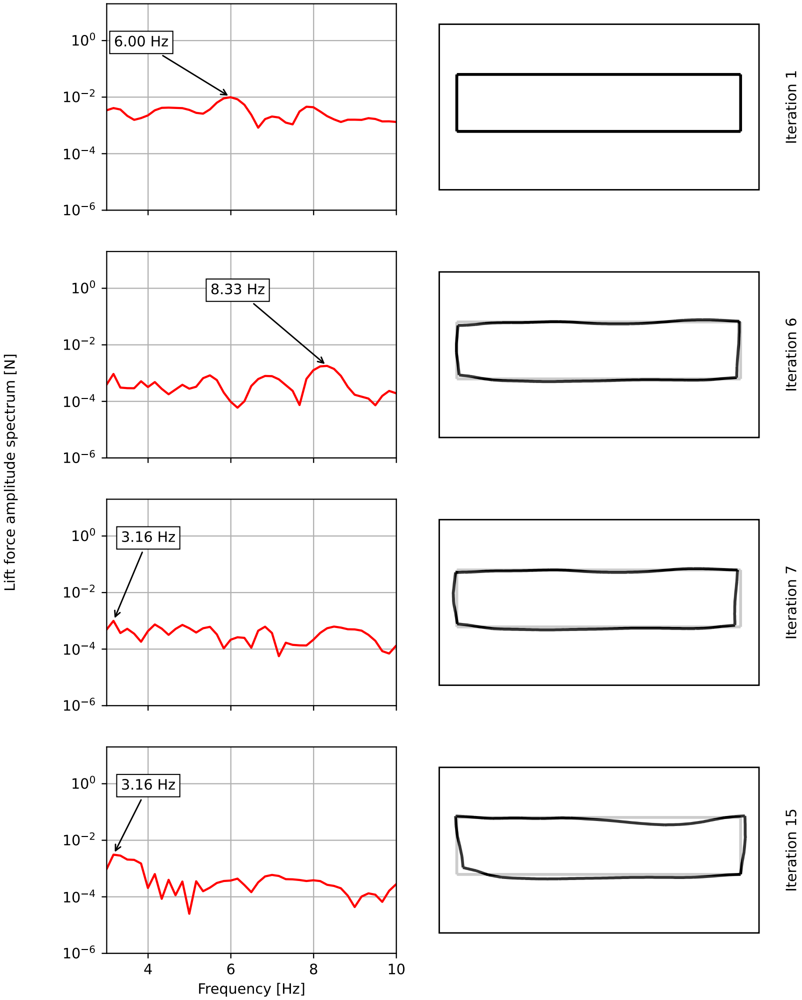
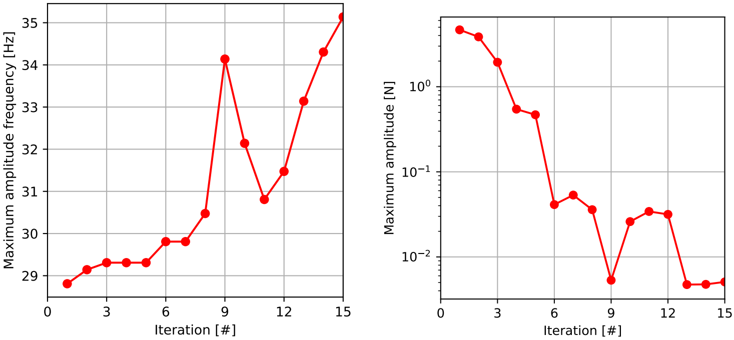


Figure 4.14: Frequency distribution and BARC cross sections within the interested range for iterations in low frequency range shape optimization problem.

tions. Therefore, in order to certify whether these arbitrary BARC cross sections are computed due to aforementioned stabilization used in adjoint solution, or whether it is inherent in the optimization problem itself, another optimization problem is solved with higher frequency range (i.e. $[20\,Hz, 50\,Hz]$). This section presents the results of it.



(a) Frequency variations

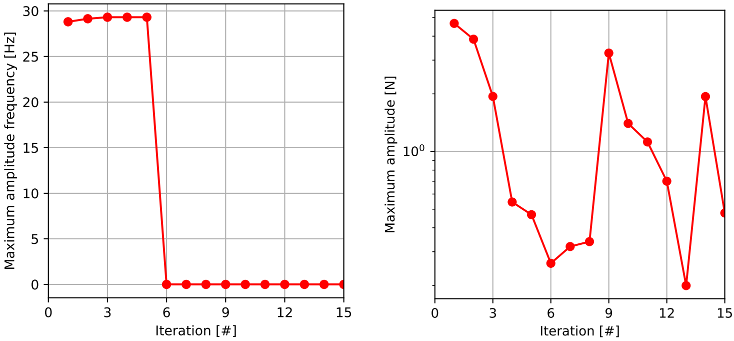
(b) Lift force amplitude variations

Figure 4.15: Maximum amplitude frequency and maximum amplitude variations with design iteration in BARC high frequency range optimization problem for optimized frequency range of $[20\,Hz, 50\,Hz]$.

A reducing trend in the maximum amplitude can be observed from figure 4.15(b) for considered high frequency range optimization problem. This provides further evidence on the ability of stabilization method to produce meaningful results for optimization problems which involve transient QOIs and chaotic flow behaviors. Figure 4.15(a) illustrates frequency of each design iteration corresponding to maximum amplitude. It also shows an increasing trend which re-assures that total energy conservation of the BARC problem in each design iteration is not violated. However, it also shows a sudden drop in the frequency corresponding to maximum amplitude at iteration 10.

In order to further investigate sudden drop in maximum amplitude frequency at iteration 10, figure 4.18 illustrates frequency distributions and their corresponding cross sections for iterations 1, 9, 10, 15. Similar to the sudden drop in the low frequency range optimization problem, drop at iteration 10 is also because of having two frequencies with close enough amplitudes causing QOI defined in equation (4.37) to switch frequencies between iterations. In contrast to the cross sections obtained in low frequency range optimization, high frequency range optimization shows a clear trend in the cross sections. Until the iteration 10 in high frequency range optimization, it shows similar design surface features at downstream section as in the flow over the cylinder case optimization when comparing design surfaces with figure 4.12.

The design surfaces illustrated in figure 4.18 depict non-symmetrical surfaces after 6^{th} iteration. From figure 4.16(a) which depicts frequency corresponding to maximum amplitude in the frequencies within $[0Hz, 50Hz]$, it can be observed that the frequency corresponding to maximum amplitude falls in the range of interest for this high frequency range optimization problem where symmetrical design surfaces are obtained as depicted in figure 4.17. After the 6^{th} iteration, the frequency with most dominant amplitude falls out of the interested range making the optimization process challenging. A similar behaviour is observed in the case of the flow over the cylinder optimization problem illustrated in figure 4.11.



(a) Frequency variations

(b) Lift force amplitude variations

Figure 4.16: Maximum amplitude frequency and maximum amplitude variations with design iteration in BARC high frequency range optimization problem illustrated for frequency range of $[0Hz, 50Hz]$.

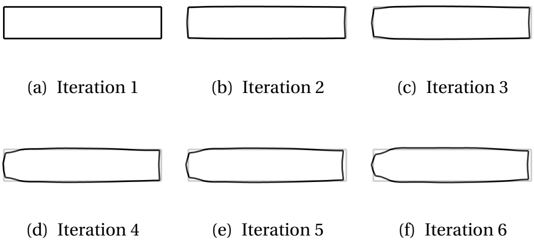


Figure 4.17: Design surface variations in BARC high frequency range optimization problem.

In this experiment also, the geometric constraints (i.e. $G_{centroid}$ and G_{volume}) are satisfied with tolerance of 1×10^{-6} in all of the design iterations obtained in solving this optimization problem.

4.6 Conclusion and Outlook

Most of the problems in CWE are chaotic and transient in nature. These chaotic transient flows are highly sensitive for initial condition perturbations ("Butterfly effect"). Due to butterfly effect, it is meaningless to calculate sensitivities of a QOI in such a flow problem using adjoint approach, because they depict an exponential growth. Therefore, this chapter introduces two novel stabilization methods to control the exponential growth and a methodology to validate adjoint sensitivities with finite differencing sensitivities.

Finite differencing sensitivities are computed for selected few mesh nodes, and the vertex morphing is used to obtain a smooth sensitivity field. A perturbation step size and vertex morphing radius studies have been carried out to determine the best possible range of values for them.

The proposed time averaged primal method uses time averaged quantities of $\underline{u}, p$ to compute RST, and then sensitivities for steady state RANS problem is derived. This removed the problem arising from butterfly effect. But the accuracy of the solution is highly dependent on the turbulence model being used in computing RST. $k-\omega-sst$ illustrated the best results which is agreeing with the directions of the reference finite difference sensitivities,

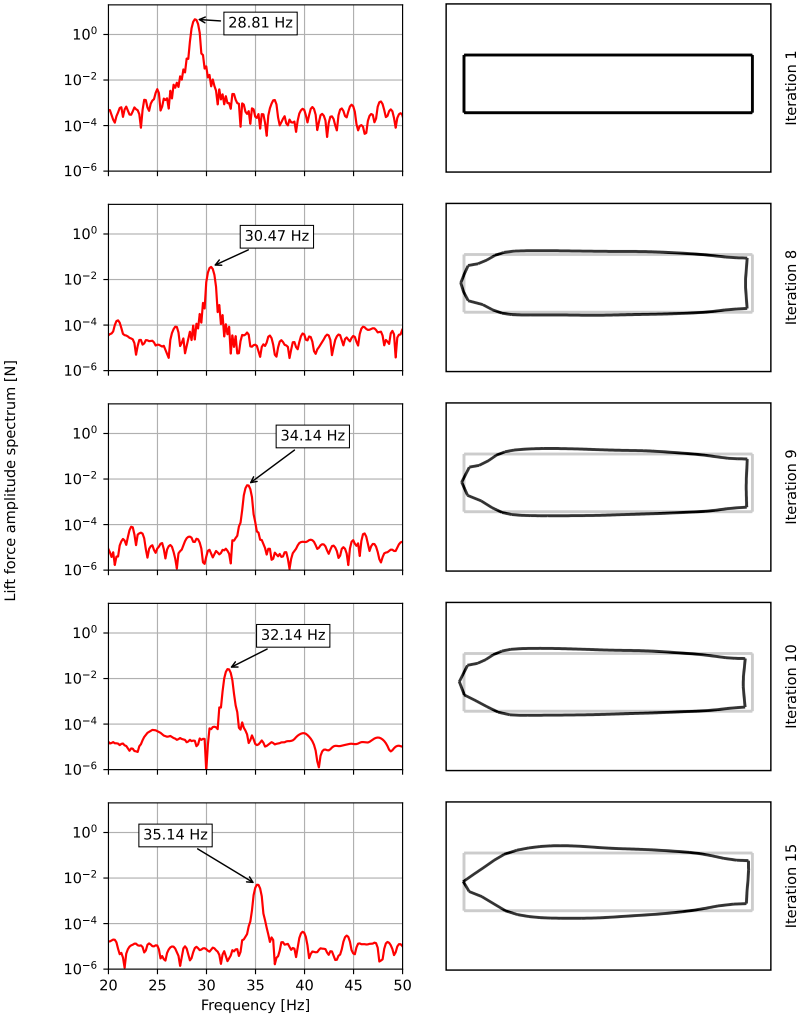


Figure 4.18: Frequency distribution within the interested range and their respective BARC cross sections for iterations in high frequency range shape optimization problem.

where as $k-\epsilon$ and $k-\omega$ failed to produce even the sensitivity directions correctly. However, time averaged primal method was not successful in obtaining shape sensitivities with the same order of magnitude as in the reference solution. Therefore, further studies are recommended with this method to be carried out with more accurate turbulence models.

The time averaged primal method lacks the ability to compute accurate sensitivities if the QOI is having weighted time averages. Therefore, another novel method named artificial diffusion is introduced in this chapter to control aforementioned exponential growth in adjoint solution. This method was successful in producing sensitivities which are agreeing in direction as well as the magnitudes with the reference sensitivities obtained from finite difference method.

Finally artificial diffusion adjoint stabilization method is used in a vortex shedding frequency optimization of BARC problem which is chaotic and transient. Two frequency ranges were used to obtain optimized shapes in two different optimization problems. Low frequency range and high frequency range optimization problems were able to obtain expected optimized solutions further proving the applicability of proposed artificial diffusion method to stabilize adjoint solutions of a transient chaotic problem. However, these optimization procedures illustrated oscillatory behavior in the QOI in later iterations due to the constant step size used in steepest descend algorithm and constant vertex morphing radius used in vertex morphing to obtain smooth sensitivities. Further studies can be conducted with adaptive step sizes and adaptive vertex morphing technologies to further improve obtained optimized design shapes. Another reason for this oscillatory behavior is, the frequency corresponding to maximum amplitude being switched or falling out of the interested frequency range. Therefore, further investigations can be carried out to minimize these frequency jumps by utilizing amplitude integration methods.

CHAPTER 5

Goal Oriented Adaptive Mesh Refinement

One of the main purposes of numerical simulations is to approximate a Quantity of Interest (QOI) (i.e. goal) acting upon a physical system (Cerutti et al. [27], Funke et al. [48], and Ricco et al. [113]). As explained in chapter 1, CWE also requires estimating specific QOIs such as drag, lift, frequency over structures to design stronger structures which can withstand harsh weather conditions or to better utilize wind energy for energy production, etc.. This makes it important to understand sources of error in a numerical simulation because these errors will grow and will propagate adversely towards approximation of the QOI.

Figure 5.1 by Szabó et al. [124] illustrates main sources of errors in numerical methods. In there, errors of idealization are the errors in simplifying complex physical phenomena. This includes simplifications such as PDE simplifications, material modelling simplifications, geometry simplifications. The next source of error is discretization which represents errors present due to the method of discretization of mathematical model after

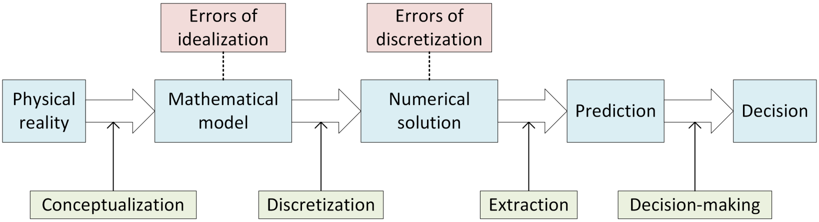


Figure 5.1: Process and associated error sources of a numerical simulation (Szabó et al. [124]).

applying the simplifications in the physical reality. So, it is required to understand the significance of each of these errors if a decision is to be made based on a QOI approximated using numerical methods.

Errors due to simplifications need to be addressed by re-evaluating modelling simplifications and assumptions which is out of scope for this study. Therefore, the focus of this study is to develop a methodology to reduce approximation error of a QOI which is evaluated using discretized numerical methods.

Spatial domain discretization error can be reduced by enriching mesh regions locally where the most adverse effects are present on the QOI approximation. It is known to be the highest time consuming process of the whole work flow of applying numerical methods if it is done by the user (refer figure 1.1). Hence, it is important to consider AMR methods which automates this process as much as possible while leaving user intervention to a minimum.

AMR is based on a given error estimator. Gradient based methods by Zienkiewicz et al. [150], Hessian based methods by Loseille et al. [85], and error estimators derived from posteriori methods by Ainsworth et al. [3] are few examples from which the error estimators can be computed. Fidkowski et al. [44] provides an in-depth review of the existing methods for calculating error estimators. Mesh adaptation can be performed with p-adaptation(Rueda-Ramírez et al. [114]), h-adaptation (Babuška et al. [9]), or hp-adaptation (Ahrabi et al. [2]).

In the context of CFD, there exists a large amount of local effects which need to be approximated using spatial discretization. The Hessian based approach aims to minimize interpolation error for a given scalar field. Therefore, it is advantageous to use Hessian approach for AMR in the context of CFD.

This chapter is organized as follows. Firstly, methodology for steady and transient AMR are presented in section 5.1. Afterwards 2D and 3D numerical example configurations are illustrated in section 5.2. Finally results of these numerical experiments are presented in section 5.3.

5.1 Methodology

This section introduces methodology used in goal oriented adaptive mesh refinement for steady and transient problems for CDR. Firstly, methodology for steady problems is explained, followed by the methodology explanation for transient problems. As explained in the introduction of chapter 5, there exists many different ways of adapting a mesh. The focus of this study is to use Hessian based approach for mesh adaptation.

5.1.1 Problem definition

The problem of interest in this study is the CDR transport equation where ϕ can represent a scalar quantity (i.e. turbulent kinetic energy), or vector quantity (i.e. velocity). The general form illustrated in equation (5.1) can be used to explain a wide array of problems governed by PDEs such as Navier-Stokes equations as well. Reader is referred to section 2.1.1 for additional details of the problem definition.

$$R = f - \underline{u} \cdot \frac{\partial \phi}{\partial \underline{x}} + \frac{\partial}{\partial \underline{x}} \left(\nu \frac{\partial \phi}{\partial \underline{x}} \right) - s\phi = 0 \tag{5.1}$$

5.1.2 Hessian Based Metric

Hessian based approach is a subset of metric-based approaches available for mesh adaptation. Here, the metric is called as the Riemannian metric field (i.e. $\underline{\mathcal{M}}$) which is a point wise defined quantity of size $d \times d$ for a domain of $\Omega \subset \mathbb{R}^d$. One advantage of metric-based approaches is, it allows to control mesh anisotropy which is controlling the element size as well as the shape and orientation as explained by works of Wallwork et al. [136].

Hessian based metric for mesh adaptation is studied extensively by Balan et al. [10], Frazza et al. [46]. The premise is as following. Assume w as the exact solution to the given PDE problem in domain Ω. If this is to be approximated in a discretized $\Omega_H \subset \Omega$, then it is required to estimate the interpolation error in Ω_H. Hessian is approximated in element $\mathcal{K} \in \Omega_H$ as $\underline{\mathbf{H}}$, and $\partial\mathcal{K}$ represents edges of the element $\mathcal{K}$. $\mathcal{I}_H$ is the interpolation operator from Ω_H. Wallwork et al. [136] derived a relationship between the $\underline{\mathbf{H}}$ and the interpolation error as depicted in equation (5.2) where $\eta > 0$ is a constant related to the spatial dimension (for additional details of the Hessian derivations, reader is referred to Mataix Ferrándiz [90]).

$$||w - \mathcal{I}_H w||_{L_\infty(\mathcal{K})} \leq \eta \max_{\underline{x} \in \mathcal{K}} \max_{\underline{e} \in \partial\mathcal{K}} \underline{e}^T \underline{\mathbf{H}}(\underline{x}) \underline{e} \tag{5.2}$$

Furthermore, from equation (5.2), a metric tensor is derived as depicted in equation (5.3).

$$\mathcal{M}_{\underline{x}} = \frac{\eta}{|\delta \tilde{w}|} \left|\underline{\mathbf{H}}\right|_{\underline{x}} \tag{5.3}$$

Where, $|\underline{\mathbf{H}}|_{\underline{x}}$ is estimated from eigen decomposition of symmetric positive definite $\underline{\mathbf{H}}|_{\underline{x}} \quad \forall \underline{x} \in \Omega_H$ as depicted in equation (5.4).

$$|\underline{\mathbf{H}}|_{\underline{x}} = \left(\underline{\mathbf{V}} |\Lambda| \underline{\mathbf{V}}^T \right)\Big|_{\underline{x}} \tag{5.4}$$

In the equation (5.3), $|\delta \tilde{w}| > 0$ represents the acceptable error level for a given scalar quantity's $\underline{\mathbf{H}}_{\underline{x}}$ at a mesh grid point $\underline{x}$.

5.1.3 Goal oriented AMR for steady problems

In order to estimate the $\mathcal{M}_{\underline{x}}$ at a given mesh grid point (i.e. node), it is important to identify acceptable error level for that particular node. This acceptable error level can be given as a user input (refer Mataix Ferrándiz [90]). Another way is to estimate this acceptable nodal error level based on the error of the QOI approximation which is the focus of this chapter. Therefore, let's assume the QOI to approximate (i.e. goal) is F as depicted in equation (5.5) where $\underline{w}$ are all the dependent variables of F.

$$F = F\left(\underline{w}\right) \tag{5.5}$$

Now, consider a coarse mesh $\Omega_H \subset \Omega$ as the initial mesh on which it is required to perform AMR and produce a goal oriented adapted mesh. The parameter H represents quantities evaluated in the coarse mesh Ω_H. Then a refined mesh (i.e. Ω_h where $\Omega_H \subset \Omega_h \subset \Omega$) is chosen such that, each coarse mesh element is subdivided into smaller refined elements in a recursive manner. h is used to represent quantities computed on the Ω_h. Due to relatively large number of elements introduced in Ω_h, it is practically not possible to obtain the solution for the physical system governed by PDEs of interest followed by QOI approximation. Therefore, Taylor expansion is used to estimate the QOI prediction error as illustrated in equation (5.6).

$$F\left(\underline{w}^h\right) = F\left(\underline{w}^{\tilde{H}}\right) + \left.\frac{\partial F}{\partial \underline{w}^h}\right|_{\underline{w}^{\tilde{H}}} \left(\underline{w}^h - \underline{w}^{\tilde{H}}\right) + \cdots \tag{5.6}$$

$\underline{w}^{\tilde{H}}$ represents the coarse mesh nodal quantities linearly interpolated on the refined mesh. Following that, QOI approximation error can be calculated as given in equation (5.7). M is the number of elements in Ω_H. In the context of FEM, QOI approximation is done by aggregating each element's contribution of the QOI, hence the QOI approximation error is also computed by aggregation of each element's error contribution.

$$\begin{aligned} \delta F &= \sum_{m=1}^{M} \left[F\left(\underline{w}_m^h\right) - F\left(\underline{w}_m^{\tilde{H}}\right) \right] \\ &= \sum_{m=1}^{M} \left.\frac{\partial F}{\partial \underline{w}_m^h}\right|_{\underline{w}_m^{\tilde{H}}} \delta \underline{w}_m^h \end{aligned} \tag{5.7}$$

Where $\underline{w}_m^{\tilde{H}}$ is computed from equation (5.8) and $\underline{\mathbf{G}}_m$ represents linear interpolation operator for m^{th} element in Ω_H.

$$\underline{w}_m^{\tilde{H}} = \underline{\mathbf{G}}_m \underline{w}_m^H \tag{5.8}$$

Assume expected QOI approximation error is $\delta \tilde{F}$ and number of refined nodes per each coarse mesh element is P, then equation (5.9) can be stated from overestimating the actual QOI approximation error by taking absolute values of each of the coarse element's error contributions.

$$\delta \tilde{F} = \sum_{m=1}^{M} \sum_{p=1}^{P} \left| \left.\frac{\partial F}{\partial \underline{w}_{mp}^h}\right|_{\underline{w}_m^{\tilde{H}}} \right| \left| \delta \underline{w}_{mp}^h \right| \geq \delta F \tag{5.9}$$

$\delta \underline{w}^{h}_{mp}$ is the expected nodal QOI approximation error on the Ω^{h}. The number of nodes in Ω^{h} is excessive therefore storing expected nodal QOI approximation errors on those nodes is prohibitively expensive. $\delta \underline{w}^{h}_{mp}$ also needs to be translated to $\delta \underline{w}^{H}_{m}$ in order to apply AMR on Ω_{H} which is the objective. Therefore, all the $\delta \underline{w}^{h}_{mp}$ in Ω^{h} will be represented by one single value per coarse element $\Omega_{m} \in \Omega^{H}$. If each coarse element is allowed to have the same expected QOI approximation error, then it allows us to simplify equation (5.9) to equation (5.10) where $\left|\delta \underline{w}^{H}_{m}\right|$ represents expected QOI approximation error in m^{th} coarse element in Ω^{H}. Afterwards $\left|\delta \underline{w}^{H}_{m}\right|$ is equally distributed among its nodes in Ω^{H} to calculate nodal expected QOI approximation error.

$$\frac{\delta \tilde{F}}{M} = \left|\delta \underline{w}^{H}_{m}\right| \sum_{p=1}^{P} \left| \left. \frac{\partial F}{\partial \underline{w}^{h}_{mp}} \right|_{\underline{w}^{\tilde{H}}_{m}} \right| \qquad \forall m \in [1, M] \tag{5.10}$$

In order to solve the above problem as a posteriori, adjoint approach is used as depicted in equation (5.11). There, λ^{h}_{mi} represents adjoint solution in Ω^{h}. $\left.\frac{\partial R_{i}}{\partial \underline{w}^{h}_{mp}}\right|_{\underline{u}^{\tilde{H}}_{m}}$ represents governing PDE's partial derivatives on Ω^{h} evaluated with values from Ω^{H} linearly interpolated on Ω^{h}.

$$\left. \frac{\partial F}{\partial \underline{w}^{h}_{mp}} \right|_{\underline{w}^{\tilde{H}}_{m}} = -\lambda^{h}_{mi} \left. \frac{\partial R_{i}}{\partial \underline{w}^{h}_{mp}} \right|_{\underline{u}^{\tilde{H}}_{m}} \tag{5.11}$$

A variety of methods are used to obtain λ^{h}_{mi} and $\left.\frac{\partial R_{i}}{\partial \underline{w}^{h}_{mp}}\right|_{\underline{u}^{\tilde{H}}_{m}}$. A study carried out by Venditti et al. [132] uses second order interpolation for λ^{h}_{mi} and linear interpolation for $\left.\frac{\partial R_{i}}{\partial \underline{w}^{h}_{mp}}\right|_{\underline{u}^{\tilde{H}}_{m}}$, whereas Power et al. [109] introduces a novel interpolations scheme to calculate λ^{h}_{mi}, and uses linear interpolation to estimate $\left.\frac{\partial R_{i}}{\partial \underline{w}^{h}_{mp}}\right|_{\underline{u}^{\tilde{H}}_{m}}$. For this study, linear interpolation is employed to estimate λ^{h}_{mi} as well as $\left.\frac{\partial R_{i}}{\partial \underline{w}^{h}_{mp}}\right|_{\underline{u}^{\tilde{H}}_{m}}$ in order to be computationally less expensive.

$$\left|\delta \underline{w}^{H}_{m}\right| = \frac{\delta \tilde{F}}{M}\left[\sum_{p=1}^{P}\left|\lambda^{\tilde{H}}_{mi}\left.\frac{\partial R_i}{\partial \underline{w}^{h}_{mp}}\right|_{\underline{w}^{\tilde{H}}_{m}}\right|\right]^{-1} \tag{5.12}$$

Element refinement procedure

This section describes the element refinement procedure used to produce Ω^h from Ω^H required by section 5.1.3. The easiest way to generate Ω^h is to subdivide each $\Omega^H_e \in \Omega^H$ to smaller elements and interpolate quantities from Ω^H_e to Ω^h_e. This is explained in detail in algorithm 9.

Algorithm 9 Element refinement procedure.

```
procedure RefineElement($\Omega^h, \underline{x}, \theta_{rl}$)
    if $\theta_{rl} = 0$ then
        Add refined nodes at $\underline{x}$ to $\Omega^h$
        Add $\Omega_e$ created by $\underline{x}$ to $\Omega^h$
        Add refined conditions to $\Omega^h$ representing $\Gamma_{\Omega_e}$
        Store nodal interpolation vectors in each node of $\Omega_e$
    else
        for $i \leq Q$ do
            Calculate nodal positions of refined element $\underline{x}_i$ by sudividing nodal positions given by $\underline{x}$
            RefineElement($\Omega_h, \underline{x}_i, \theta_{rl} - 1$)
```

Number of subdivisions carried out in each Ω^H_e is represented by Q in algorithm 9. Figure 5.1 represents single triangle element refinement carried out with $\theta_{rl} = 1$ in left and $\theta_{rl} = 2$ in right where $Q = 4$. This will result in a Ω^h with $4^{\theta_{rl}} M$ number of refined elements for given M number of coarse elements in Ω^H. This can lead to prohibitively expensive amount of elements for higher θ_{rl}. Therefore, in this study, only one Ω^H_e is refined (i.e. Ω^h_e) and stored in memory. When refined mesh quantities for a given Ω^H_e is required, then Ω^h_e is moved to its respective location of Ω^H_e, and the values are interpolated, and surface conditions on $\Gamma_{\Omega^H_e}$ is made active if they are in Γ.

Figure 5.3 illustrates tetrahedral element refinement carried out in $3D$. This is done by dividing Ω^H_e to four corner tetrahedrons, and then the

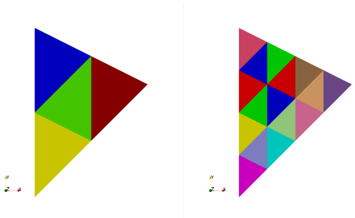


Figure 5.2: Triangle element refinement in 2D [Left: $\theta_{rl} = 1$, Right: $\theta_{rl} = 2$].

inner octagon is further subdivided into four tetrahedrons along the main diagonal, which results in a subdivision with $Q = 8$. This will result in Ω_h with $8^{\theta_{rl}} M$ number of elements. Due to the high computational memory cost as explained before, in $3D$ also only one refined element is created and then values and nodal positions are interpolated from the values of Ω_e^H when quantities form Ω^h is requested.

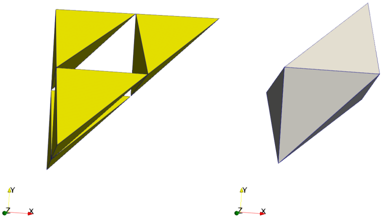


Figure 5.3: Tetrahedral element refinement in 3D [Left: Refined elements of outer most tetrahedral elements, Right: Refined elements in innermost octagon].

Steady state goal oriented adaptive mesh refinement procedure

This section introduces the algorithm (refer figure 5.4) used in AMR for steady state problems. First refined Ω_e^h is created and stored using first $\Omega_e^H \in \Omega^H$ using algorithm 9. Then, primal steady problem is solved using Ω^H followed by adjoint problem solve in the same coarse mesh. Afterwards each Ω_e^H is interpolated to Ω_e^h to calculate $\left|\delta \underline{w}_m^H\right|$ using "Estimate Nodal

Interpolation Error" illustrated in figure 5.4 $\forall m \in [1, M]$. Finally stored $\left|\delta \underline{w}_m^H\right|$ in nodes are used to perform AMR on Ω^H. This process is repeated until user specified N_{max} iterations are reached.

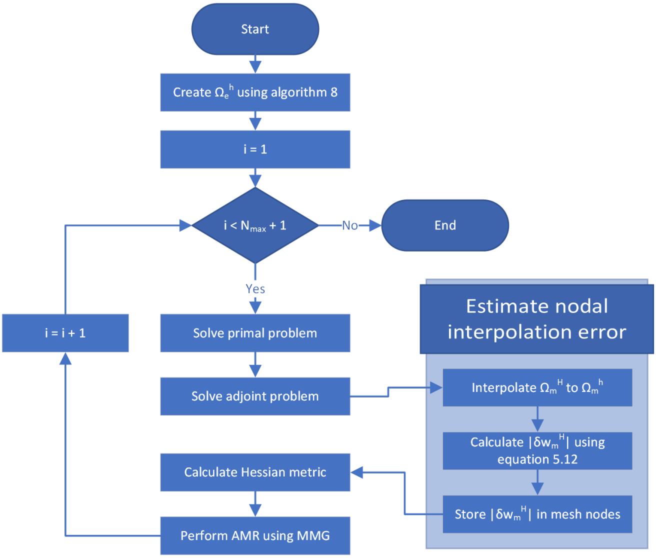

Start
Create Ω_e^h using algorithm 8
i = 1
i < N_{max} + 1
No
End
Yes
i = i + 1
Solve primal problem
Solve adjoint problem
Calculate Hessian metric
Perform AMR using MMG
Estimate nodal interpolation error
Interpolate Ω_m^H to Ω_m^h
Calculate $|\delta w_m^H|$ using equation 5.12
Store $|\delta w_m^H|$ in mesh nodes

Figure 5.4: AMR procedure for steady state problems.

5.1.4 Goal oriented AMR for transient problems

The next class of problems to be considered in this study is transient CDR transport equation as depicted in equation (5.13).

$$R = f - \frac{\partial \phi}{\partial t} - \underline{u} \cdot \frac{\partial \phi}{\partial \underline{x}} + \frac{\partial}{\partial \underline{x}}\left(\nu \frac{\partial \phi}{\partial \underline{x}}\right) - s\phi = 0 \tag{5.13}$$

Time discretization is performed using 2^{nd} order accurate Bossak method (Wood et al. [147]). Time derivatives are derived using equation (5.14).

$$Z = \dot{\phi}^n - \frac{1}{\gamma \Delta t}\left(\phi^n - \phi^{n-1}\right) - \frac{\gamma - 1}{\gamma}\phi^{\dot{n}-1} = 0 \tag{5.14}$$

Statistical quantities are of an interest in transient problems rather than the time instantaneous values in CWE. Therefore, time averaged QOI function is used in this study as depicted in equation (5.15), where N is the number of time steps.

$$F = \frac{1}{N} \sum_{n=1}^{N} f^n \left(\underline{w}^n, \underline{\dot{w}}^n, \underline{\dot{w}}^{n-1} \right) \tag{5.15}$$

As described in the section 5.1.3, Taylor expansion is used to calculate function approximation error as shown in equation (5.16) where $\underline{\tilde{w}}_m^{\tilde{H},n} = \left[\underline{w}_m^{\tilde{H},n}, \underline{w}_m^{\tilde{H},n-1}, \underline{\dot{w}}_m^{\tilde{H},n}, \underline{\dot{w}}_m^{\tilde{H},n-1} \right]$.

$$\begin{aligned} \delta F = \frac{1}{N} \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{p=1}^{P} \Bigg[& \left. \frac{\partial f^n}{\partial \underline{w}_{mp}^{h,n}} \right|_{\underline{\tilde{w}}_m^{\tilde{H},n}} \delta \underline{w}_{mp}^{h,n} + \left. \frac{\partial f^n}{\partial \underline{\dot{w}}_{mp}^{h,n}} \right|_{\underline{\tilde{w}}_m^{\tilde{H},n}} \delta \underline{\dot{w}}_{mp}^{h,n} \\ & + \left. \frac{\partial f^n}{\partial \underline{\dot{w}}_{mp}^{h,n-1}} \right|_{\underline{\tilde{w}}_m^{\tilde{H},n}} \delta \underline{\dot{w}}_{mp}^{h,n-1} \Bigg] \end{aligned} \tag{5.16}$$

It is assumed to have the same expected error from each n^{th} time step and each m^{th} coarse element. Δt is assumed to be chosen such that it has negligible error contribution. Finally we like to over-estimate expected QOI approximation error. As a result, equation (5.17) can be derived by using equation (5.16) and equation (5.14) where $\delta \tilde{F}$ is the expected QOI approximation error.

$$\begin{aligned}
&\frac{\delta\tilde{F}}{M} = \\
&\sum_{p=1}^{P}\left[a_{mp}^{h,n}\left|\delta\underline{w}_{mp}^{h,n}\right| + b_{mp}^{h,n}\left(\left|\frac{\gamma-1}{\gamma}\right|\left|\delta\underline{\dot{w}}_{mp}^{h,n-1}\right| + \left|\frac{\delta\underline{w}_{mp}^{h,n-1}}{\gamma\Delta t}\right|\right)\right] \geq \frac{\delta F}{M} \\
&\quad \forall n \in [1,N] \quad \textit{and} \quad \forall m \in [1,M] \\
&\textit{where} \\
&a_{mp}^{h,n} = \left|\left(\frac{\partial f^{n}}{\partial\underline{w}_{mp}^{h,n}}\bigg|_{\underline{\tilde{w}}_{m}^{\tilde{H},n}} + \frac{1}{\gamma\Delta t}\left\{\frac{\partial f^{n}}{\partial\underline{\dot{w}}_{mp}^{h,n}}\bigg|_{\underline{\tilde{w}}_{m}^{\tilde{H},n}} + \frac{\partial f^{n+1}}{\partial\underline{\dot{w}}_{mp}^{h,n}}\bigg|_{\underline{\tilde{w}}_{m}^{\tilde{H},n+1}}\right\}\right)\right| \\
&b_{mp}^{h,n} = \left|\left(\frac{\partial f^{n}}{\partial\underline{\dot{w}}_{mp}^{h,n}}\bigg|_{\underline{\tilde{w}}_{m}^{\tilde{H},n}} + \frac{\partial f^{n+1}}{\partial\underline{\dot{w}}_{mp}^{h,n}}\bigg|_{\underline{\tilde{w}}_{m}^{\tilde{H},n+1}}\right)\right|
\end{aligned} \tag{5.17}$$

Due to high computational storage cost involved in storing $\delta\underline{w}_{mp}^{h,n}$, $\delta\underline{\dot{w}}_{mp}^{h,n}$, and $\delta\underline{\dot{w}}_{mp}^{h,n-1}$, each of these quantities are to be represented by one value per coarse element respectively $\delta\underline{w}_{m}^{H,n}$, $\delta\underline{\dot{w}}_{m}^{H,n}$, and $\delta\underline{\dot{w}}_{m}^{H,n-1}$. Thus equation (5.17) can be rearranged as depicted in equation (5.18).

$$\begin{aligned}
\frac{\delta\tilde{F}}{M} = &\left|\delta\underline{w}_{m}^{H,n}\right|\underbrace{\sum_{p=1}^{P}a_{mp}^{h,n}}_{a_{m}^{n,h}} \\
&+\left(\left|\frac{\gamma-1}{\gamma}\right|\left|\delta\underline{\dot{w}}_{m}^{H,n-1}\right| + \left|\frac{1}{\gamma\Delta t}\right|\left|\delta\underline{w}_{m}^{H,n-1}\right|\right)\underbrace{\sum_{p=1}^{P}b_{mp}^{h,n}}_{b_{m}^{n,h}}
\end{aligned} \tag{5.18}$$

$\left|\delta \underline{\dot{w}}_m^{h,n}\right|$ is calculated using equation (5.19) where $\delta \underline{w}_m^{H,0} = 0$ and $\delta \underline{\dot{w}}_m^{H,0} = 0$ $\forall m \in [1, M]$.

$$\left|\delta \underline{\dot{w}}_m^{H,n}\right| = \left|\frac{1}{\gamma \Delta t}\right| \left(\left|\delta \underline{w}_m^{H,n}\right| + \left|\delta \underline{w}_m^{H,n-1}\right|\right) + \left|\frac{\gamma - 1}{\gamma}\right| \left|\delta \underline{\dot{w}}_m^{H,n-1}\right| \tag{5.19}$$

Terms $a_m^{h,n}$ and $b_m^{h,n}$ needs to be calculated in Ω^h using the adjoint solution for the defined lagrangian as depicted in equation (5.20) where λ^n is R^n's lagrangian multiplier and μ^n is the lagrangian multiplier of Z^n.

$$L^{\tilde{H}} = \sum_{n=1}^{N} \sum_{m=1}^{M} \left[\frac{1}{N} f^n + \sum_{i=1}^{P} \left(\lambda_{mi}^{h,n} R_i^n + \mu_{mi}^{h,n} Z_i^n \right) \right] \Bigg|_{\underline{\tilde{w}}_m^{\tilde{H},n}} \tag{5.20}$$

Adjoint problem for lagrangian definition in equation (5.20) is given by equation (5.21) where $\lambda_{mp}^{h,N+1} = 0$ and $\mu_{mp}^{h,N+1} = 0$ $\forall m \in [1, M]$ and $\forall p \in [1, P]$.

$$\begin{aligned}
\lambda_{mi}^{h,n} \left[\frac{\partial R_i^n}{\partial \underline{w}_{mp}^{h,n}} + \frac{1}{\gamma \Delta t} \frac{\partial R_i^n}{\partial \underline{\dot{w}}_{mp}^{h,n}} \right] \Bigg|_{\underline{\tilde{w}}_m^{\tilde{H},n}} = & - \frac{\partial f^n}{\partial \underline{w}_{mp}^{h,n}} \Bigg|_{\underline{\tilde{w}}_m^{\tilde{H},n}} \\
& - \frac{1}{\gamma \Delta t} \frac{\partial f^n}{\partial \underline{\dot{w}}_{mp}^{h,n}} \Bigg|_{\underline{\tilde{w}}_m^{\tilde{H},n}} \\
& - \frac{1}{\gamma \Delta t} \frac{\partial f^{n+1}}{\partial \underline{\dot{w}}_{mp}^{h,n}} \Bigg|_{\underline{\tilde{w}}_m^{\tilde{H},n+1}} \\
& - \frac{1}{\gamma \Delta t} \lambda_{mi}^{h,n+1} \frac{\partial R_i^{n+1}}{\partial \underline{\dot{w}}_{mp}^{h,n}} \Bigg|_{\underline{\tilde{w}}_m^{\tilde{H},n+1}} \\
& - \frac{1}{\gamma^2 \Delta t} \mu_{mp}^{h,n+1}
\end{aligned} \tag{5.21}$$

Where $\mu_{mp}^{h,n}$ is defined in equation (5.22).

$$\begin{aligned}\mu_{mp}^{h,n} = & -\left.\frac{\partial f^{n}}{\partial \underline{\dot{w}}_{mp}^{h,n}}\right|_{\underline{\tilde{w}}_m^{\tilde{H},n}} - \left.\frac{\partial f^{n+1}}{\partial \underline{\dot{w}}_{mp}^{h,n}}\right|_{\underline{\tilde{w}}_m^{\tilde{H},n+1}} - \lambda_{mi}^{h,n}\left.\frac{\partial R_i^{n}}{\partial \underline{\dot{w}}_{mp}^{h,n}}\right|_{\underline{\tilde{w}}_m^{\tilde{H},n}} \\ & - \lambda_{mi}^{h,n+1}\left.\frac{\partial R_i^{n+1}}{\partial \underline{\dot{w}}_{mp}^{h,n}}\right|_{\underline{\tilde{w}}_m^{\tilde{H},n+1}} + \frac{\gamma-1}{\gamma}\mu_{mp}^{h,n+1}\end{aligned} \tag{5.22}$$

From equation (5.21) and equation (5.22), $a_{mp}^{h,n}$ and $b_{mp}^{h,n}$ terms can be derived. They may require prohibitively expensive computational cost to compute because $\lambda_{mi}^{h,n}$ needs to be computed by solving equation (5.21) in Ω^h. Consequently, linearly interpolated $\lambda_{mi}^{\tilde{H},n} \approx \lambda_{mi}^{h,n}$ is computed from its counterpart $\lambda_{mi}^{H,n}$. $\lambda_{mi}^{H,n}$ is evaluated by solving same equation (5.21) in Ω^H. Finally we can derive $a_{mp}^{h,n}$ and $b_{mp}^{h,n}$ terms as depicted in equation (5.23) and equation (5.24) respectively.

$$\begin{aligned}a_{mp}^{h,n} = & -\lambda_{mi}^{\tilde{H},n}\left.\left[\frac{\partial R_i^{n}}{\partial \underline{w}_{mp}^{h,n}} + \frac{1}{\gamma \Delta t}\frac{\partial R_i^{n}}{\partial \underline{\dot{w}}_{mp}^{h,n}}\right]\right|_{\underline{\tilde{w}}_m^{\tilde{H},n}} \\ & - \frac{1}{\gamma \Delta t}\lambda_{mi}^{\tilde{H},n+1}\left.\frac{\partial R_i^{n+1}}{\partial \underline{\dot{w}}_{mp}^{h,n}}\right|_{\underline{\tilde{w}}_m^{\tilde{H},n+1}} - \frac{1}{\gamma^2 \Delta t}\mu_{mp}^{h,n+1}\end{aligned} \tag{5.23}$$

$$b_{mp}^{h,n} = \mu_{mp}^{n,h} - \lambda_{mi}^{\tilde{H},n}\left.\frac{\partial R_i^{n}}{\partial \underline{\dot{w}}_{mp}^{h,n}}\right|_{\underline{\tilde{w}}_m^{\tilde{H},n}} - \lambda_{mi}^{\tilde{H},n+1}\left.\frac{\partial R_i^{n+1}}{\partial \underline{\dot{w}}_{mp}^{h,n}}\right|_{\underline{\tilde{w}}_m^{\tilde{H},n+1}} + \frac{\gamma-1}{\gamma}\mu_{mp}^{h,n+1} \tag{5.24}$$

Once $\left|\delta \underline{w}_m^{H,n}\right|$ is computed from equation (5.18) for each time step, then they are aggregated to one coarse elemental quantity as depicted in equation (5.25).

$$\frac{1}{\left|\delta \underline{w}_m^{H}\right|} = \sum_{n=1}^{N}\frac{1}{\left|\delta \underline{w}_m^{H,n}\right|} \tag{5.25}$$

Transient goal oriented adaptive mesh refinement procedure

Until now we have discussed formulations which are used to calculate required nodal interpolation error of QOI's dependent quantities. As discussed above, the transient adjoint problem needs to be solved backwards in time because only $\lambda_{mp}^{\tilde{H},N+1} = 0$ and $\mu_{mp}^{\tilde{H},N+1} = 0$ are defined in equation (5.21). On the other hand, $\left|\delta \underline{w}_m^{H,n}\right|$ needs to be calculated by marching forward in time using equation (5.18) because only $\left|\delta \underline{w}_m^{H,0}\right| = 0$ and $\left|\delta \underline{\dot{w}}_m^{H,0}\right| = 0$ is defined. Therefore following algorithm is used in this study to estimate nodal interpolation error.

Algorithm 10 describes the procedure followed in this study for AMR. Firstly, total time span responsible for $n \in [1, N]$ time steps is subdivided into smaller $\tilde{N}$ time spans of equal length such that $\left[N_1, N_2\right] \cup \left[N_2, N_3\right] \cdots = [1, N]$. In each smaller time span, the primal and adjoint problems are solved in Ω_i^H where i being the i^{th} subdivision. Afterwards, QOI function interpolation error is estimated using equation (5.25). This is followed by calculation of Hessian matrix for each dof. Thereafter, anisotropic mesh adaptation is carried out to generate next smaller time span's mesh (i.e. Ω_{i+1}^H).

Once the next time span's mesh is generated, then nodal values from old mesh are interpolated/mapped to the new mesh using three different methods. First one being the "nearest neighbor" approach where it calculates nodal values on Ω_{i+1}^H based on each nodes' nearest neighbor in Ω_i^H. The second mapping methodology is called "nearest element", which identifies the element (in Ω_i^H) in which each node in Ω_{i+1}^H lies, and uses linear shape functions to interpolate. The third approach for mapping is called "smoothed mapping" which is a novel mapping technique introduced in this thesis. It uses the same "nearest element" approach. However, mapping is not done on the time instance where the AMR happens. It uses solution field at θ_{ms} time steps before the AMR time step (this is saved using HDF format (The HDF Group [127]) check pointing method) to map Ω_i^H fields to Ω_{i+1}^H. Thereafter, forward time marching is done until it reaches time instance where AMR is performed with an adjusted time step. Adjusted time step is calculated by computing the maximum CFL number in the mapped refined mesh. The maximum CFL is used to estimate the required time step to achieve user desired CFL. Then the same procedure is repeated for next time span.

Algorithm 10 Nodal interpolation error estimation.

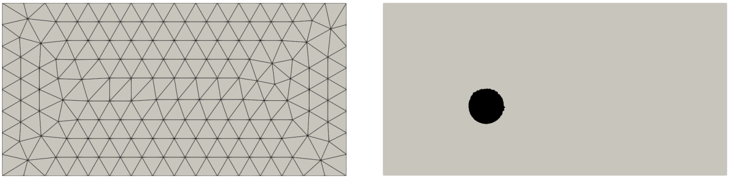
```
procedure TransientAdaptiveMeshRefinement($\Omega^H, \theta_{rl}, \tilde{N}$)
    Create $\Omega^h$ using algorithm 9.
    for $i \le \tilde{N}$ do
        $n \leftarrow N_i$
        $\underline{\bar{w}}^{H,n-1} \leftarrow 0$
        for $n \le N_{i+1}$ do
            Solve primal (ref. equation (5.13)) for $n^{th}$ time step in $\Omega^H_i$
            $\underline{\bar{w}}^{H,n} \leftarrow Mean\left(\underline{w}^{H,n}, \underline{\bar{w}}^{H,n-1}\right)$ (refer appendix A)
            Store $\underline{\bar{w}}^{H,n}$, $\underline{w}^{H,n}$, $\underline{\dot{w}}^{H,n}$
            $n \leftarrow n+1$
        $n \leftarrow N_{i+1}$
        for $n \ge N_i$ do
            Load $\underline{w}^{H,n}$, $\underline{\dot{w}}^{H,n}$
            Solve adjoint (ref. equation (5.21)) for $n^{th}$ time step in $\Omega^H_i$.
            Calculate $a^{h,n}_m$, $b^{h,n}_m$ using equations 5.23, equation (5.24)
            Store $a^{h,n}_m$ and $b^{h,n}_m$
            $n \leftarrow n-1$
        $n \leftarrow N_i$
        $\left|\delta \underline{w}^{H,n-1}_m\right| \leftarrow 0 \quad \forall m \in [1, M]$
        $\left|\delta \underline{\dot{w}}^{H,n-1}_m\right| \leftarrow 0 \quad \forall m \in [1, M]$
        $\left|\delta \underline{w}^{H}_m\right| \leftarrow 1 \times 10^{100} \quad \forall m \in [1, M]$
        for $n \le N_{i+1}$ do
            Load $a^{h,n}_m$, $b^{h,n}_m$
            Calculate $\left|\delta \underline{w}^{H,n}_m\right|$, $\left|\delta \underline{\dot{w}}^{H,n}_m\right|$ using equations 5.18, 5.14
            Update $\left|\delta \underline{w}^{H}_m\right|$ using equation (5.25)
            $n \leftarrow n+1$
        Calculate $\underline{\mathbf{H}}$ using $\underline{\bar{w}}^{H,N_{i+1}}$
        Calculate $\mathcal{M}$ using $\underline{\mathbf{H}}$ and $\left|\delta \underline{w}^{H}_m\right|$
        $\Omega^H_{i+1} \leftarrow$ Perform AMR on $\Omega^H_i$
        Map $\Omega^H_i$ data to $\Omega^H_{i+1}$
        Adjust $\Delta t$
        $n \leftarrow N_i$
        for $n \le N_{i+1}$ do
            Solve primal (ref. equation (5.13)) for $n^{th}$ time step in $\Omega^H_{i+1}$
            $n \leftarrow n+1$
        $i \leftarrow i+1$
```

5.2 Numerical Experiments

So far this paper has focused on formulations and procedures of the AMR study. The following section will introduce numerical experiments on which proposed AMR is applied. Firstly $2D$ scalar experiments are introduced, then followed by a $3D$ example.

5.2.1 Circular convection

This experiment is selected to investigate proposed AMR procedure on steady scalar transport equations. It is taken from Hubbard [62]. Primal problem details are explained in section 2.2.1. Initial mesh for this problem is selected to be coarse with triangular elements as depicted in figure 5.5(a). One of the reasons to have such a coarsened mesh is to highlight the effect of proposed AMR procedure.



(a) Initial mesh

(b) QOI function domain area

Figure 5.5: Circular convection initial mesh and QOI function domain area.

QOI function (i.e. F) for this numerical experiment is depicted in equation (5.26), where the domain Ω_F is illustrated in figure 5.5(b).

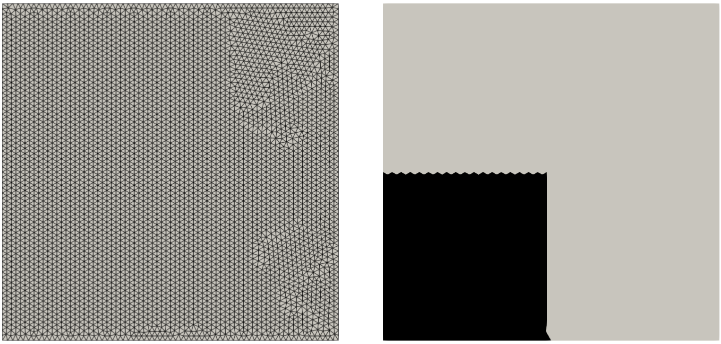
$$F = \int_{\Omega_F} \phi \, d\Omega \tag{5.26}$$

5.2.2 Diffusion only

Diffusion only initial value experiment is investigated to analyze proposed transient AMR procedure. This experiment is also a $2D$ scalar transport problem as depicted in equation (5.27).

$$\frac{\partial \phi}{\partial t} = \frac{\partial}{\partial \underline{x}}\left(\frac{\partial \phi}{\partial \underline{x}}\right) \quad \text{in} \quad \Omega = (0,1) \times (0,1) \quad \text{for} \quad t \in (0,40] \tag{5.27}$$

QOI function used in this experiment is same as the circular convection experiment which is depicted in equation (5.26). Initial mesh and Ω_F are illustrated in figure 5.6(a) and figure 5.6(b) respectively.



(a) Initial mesh

(b) QOI function domain area

Figure 5.6: Diffusion only initial mesh and QOI function domain area.

Equation (5.28) depicts a single hump used to initialize this experiment. A is the magnitude, (x_0, y_0) is the center coordinates of the hump, and r_0 is the radius of the hump.

$$\phi = \begin{cases} A^{\frac{1+cos(\pi r)}{4}} & \mathit{if} \qquad r = \frac{\sqrt{(x-x_0)^2+(y-y_0)^2}}{r_0} \leq 1.0 \\ 0 & \mathit{otherwise} \end{cases} \tag{5.28}$$
$$\forall x, y \in (0,1)^2 \quad \mathit{and} \quad t = 0$$

Two initial value conditions are considered in this experiment. First experiment consists of a single hump centered in the domain (i.e. $x_0 = 0.5$, $y_0 = 0.5$ with $A = 10.0$ and $r_0 = 0.15$). Second experiment consists of two humps centered at $x_0 = 0.25$, $y_0 = 0.5$ and $x_0 = 0.75$, $y_0 = 0.5$ with $A = 10.0$ and $r_0 = 0.05$.

5.2.3 2D flow over a cylinder

This experiment is carried out to investigate proposed AMR procedure's performance with reference to highly non-linear 2D steady-state problem. It is based on the work done by Catalano et al. [26]. The experiment consist of rectangular computational domain (i.e. Ω) with a cylinder with a diameter $D = 1\,m$. Upstream distance in Ω is $22D$, downstream is $17D$, and $24D$ is the distance to the far field. Inlet is applied with constant velocity of $u = 10.0\,ms^{-1}$, turbulent intensity of 0.2. Far field is assigned with slip conditions. Outlet is assigned with fixed pressure conditions with $P = 0$. Linear log-law wall functions are used on the cylinder. The Reynolds number corresponding to this experiment is 1×10^6. Drag over the cylinder is used as the QOI function for AMR procedure.

5.2.4 3D flow over a cylinder

This experiment is based on work of Bayraktar et al. [14]. The experiment with transient inlet corresponds to $Re = 100$ is chosen to investigate proposed transient AMR procedure. The computational domain for this problem is made exact as the reference experiment, and the boundary conditions are also made exact.

Two types of inlets are investigated with this experiment. First one with the steady inlet as depicted in equation (5.29a) then followed by transient inlet experiment as depicted in equation (5.29b).

$$u(0, y, z, t) = 16 \times 2.25 \times \frac{yz(0.41 - y)(0.41 - z)}{0.41^4} \tag{5.29a}$$

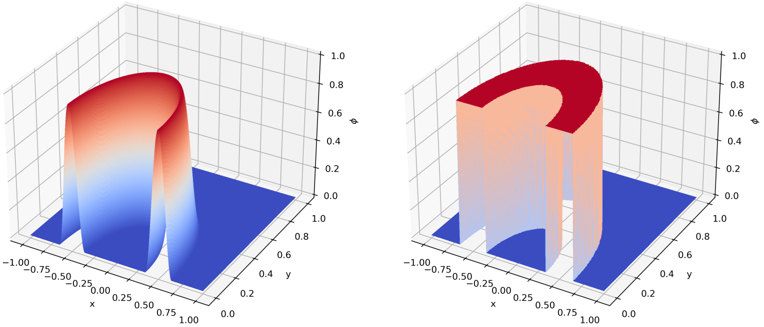
$$u(0, y, z, t) = 16 \times 2.25 \times \sin\left(\frac{\pi t}{8}\right) \frac{yz(0.41 - y)(0.41 - z)}{0.41^4} \tag{5.29b}$$

5.3 Results of Numerical Experiments and Discussion

In this section we present and systematically analyze results of numerical experiments. Firstly, 2D scalar steady transport circular convection results are presented and compared against the exact solutions. Then followed by scalar 2D transient diffusion only experiment results are presented. Afterwards 2D steady flow over cylinder experiment with $k-\omega-sst$ turbulence model's results are presented. Finally transient flow over cylinder results are presented.

5.3.1 Circular convection

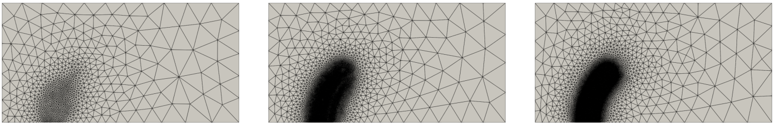
Figure 5.7(a) and figure 5.7(b) illustrate exact solutions for the circular convection problem with smooth solution (i.e G_1 in equation (2.19)) and with discontinuous solution (i.e. G_2 in equation (2.19)) respectively. This problem is investigated to understand the performance of the proposed AMR method for steady state problems. This study is carried out with



(a) Smooth

(b) Discontinuous

Figure 5.7: Exact solutions of circular convection problem.



(a) Iteration 2

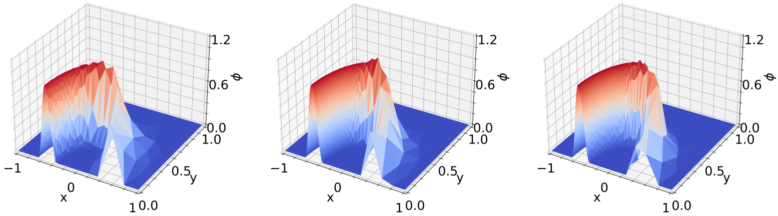
(b) Iteration 3

(c) Iteration 4

Figure 5.8: AMR study meshes in each iteration with $\theta_{rl} = 2$ and $\delta\tilde{F} = 5 \times 10^{-2}$ for circular convection problem with smooth solution.

different θ_{rl} and $\delta\tilde{F}$ to investigate the effect on the adapted mesh in mesh iteration study. The QOI functional is chosen as given in equation (5.26).

Figure 5.8 illustrate adapted meshes after each iteration for smooth solution problem with $\theta_{rl} = 2$ and $\delta\tilde{F} = 5 \times 10^{-2}$. As the applied convective velocity in the domain is circular (refer equation (2.17)), it is in the clockwise direction for $\left(x,y\right)\in(-1,0)\times(0,1)$ range. It is encouraging to compare adapted meshes in figure 5.8 with its initial mesh illustrated in figure 5.5(a). While the initial mesh is made coarse, figure 5.9 illustrates the proposed AMR procedure identifies regions correctly which are critical for objective evaluation. In this case the critical region being the circular region which is being refined as in the figures figure 5.8, non-critical region being $x>0$ coarsened out even more than the initial mesh. This mesh coarsening results in overshoots in the solution as depicted in figure 5.9 because the proposed AMR procedure identifies these regions as the regions which do not have strong influence over the objective evalutation, thus the illustrated overshoots do not affect the objective evaluation.

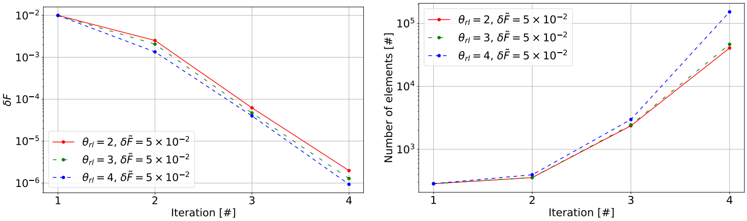


(a) Iteration 2 (b) Iteration 3 (c) Iteration 4

Figure 5.9: ϕ distribution in circular convection smooth solution AMR study meshes in each iteration with $\theta_{rl}=2$ and $\delta\tilde{F}=5\times10^{-2}$.

Having discussed how the proposed AMR method is producing mesh refinement which is capable of optimizing computational cost via coarsening non-critical regions and refining critical regions for QOI function evaluation, it is also important to investigate effect of θ_{rl} on the overall procedure.

Figure 5.10(a) illustrates actual error for each adapted mesh iteration for different θ_{rl} values. It can be observed that with adapted mesh based on higher θ_{rl} have lower error than with lower θ_{rl}. In addition, figure 5.10(b) shows, higher the θ_{rl} higher the number of elements. However, the calculated QOI function approximation error for different θ_{rl} do not have significant difference. Therefore, it is safer to assume in practical applica-



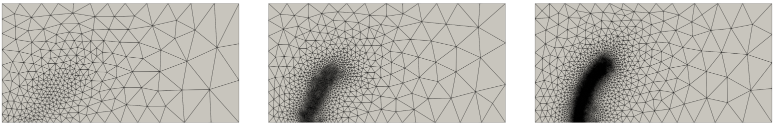
(a) Error

(b) Number of elements

Figure 5.10: Circular convection smooth AMR study meshes' solution errors and number of elements.

tions $\theta_{rl} = 2$ would suffice. Another observation from figure 5.10(a) is that actual error (i.e. δF) is smaller than the expected QOI approximation error (i.e. $\delta\tilde{F}$). This is due to the fact that in equation (5.9) the expected QOI approximation is over estimated. This indicates that in order to achieve a desired level of QOI approximation accuracy, it is possible to use a higher level of expected QOI approximation error which will result in lower cost for more complex problems.

Resultant meshes of the discontinuous solution are presented in figure 5.11. This numerical experiment is carried out to investigate how well the proposed AMR performs near discontinuities. As depicted in the figure, it correctly identifies critical regions for QOI functional approximation as in the smooth solution.



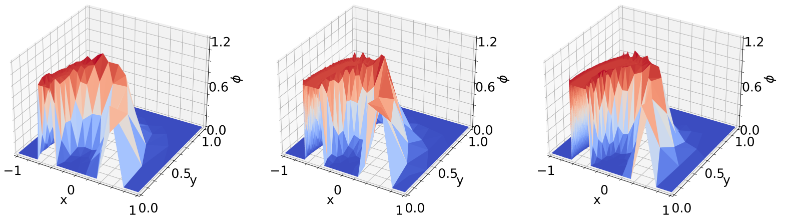
(a) Iteration 2

(b) Iteration 3

(c) Iteration 4

Figure 5.11: AMR study meshes in each iteration $\theta_{rl} = 2$ and $\delta\tilde{F} = 5\times10^{-1}$ for circular convection problem with discontinuous solution.

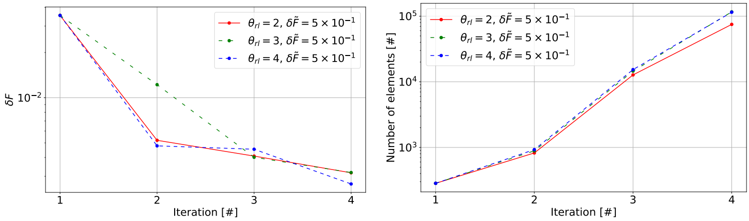
Figure 5.12 illustrates ϕ distribution for different adapted meshes used in solving discontinuous boundary condition problem. Oscillations can be observed near the discontinuity due to limitations in CDR transport equation formulations.



(a) Iteration 2 (b) Iteration 3 (c) Iteration 4

Figure 5.12: ϕ distribution in circular convection discontinuous solution AMR study meshes in each iteration $\theta_{rl} = 2$ and $\delta\tilde{F} = 5 \times 10^{-1}$.

Figure 5.13(a) illustrates actual errors calculated on θ_{rl} study for discontinuous solution problem. As a result of oscillations at the discontinuity, oscillations can also be observed in actual errors as well. However, the number of elements in each mesh iteration for discrete solution problem depicted in figure 5.13(b) follows the same trend as in the smooth solution problem.

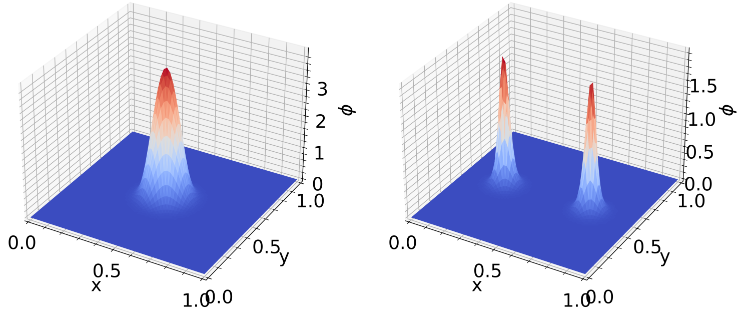


(a) Error (b) Number of elements

Figure 5.13: Circular convection discontinuous AMR study meshes' solution errors and number of elements.

5.3.2 Diffusion only

The exact initialization used in this experiment is visualized in figure 5.14 where figure 5.14(a) illustrates initialization with only a single hump and figure 5.14(b) illustrates initialization with two humps.

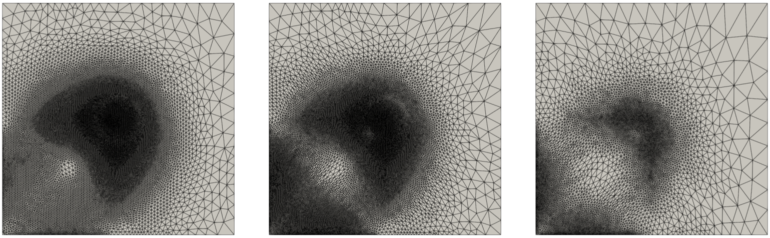


(a) Single hump at $x_0 = 0.5, y_0 = 0.5, r_0 = 0.15$ (b) Double humps with $r_0 = 0.05$ at $(x_0, y_0) = (0.25, 0.5)$ and $(x_0, y_0) = (0.75, 0.5)$

Figure 5.14: Initial values of diffusion only experiment with $A = 10$.

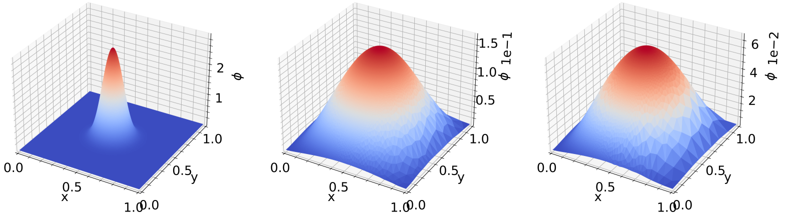
Firstly, resultant meshes of diffusion only AMR experiment with single hump are illustrated in figure 5.15. While the computed $\underline{\mathbf{H}}$ is symmetrical along $x = 0.5$ and $y = 0.5$ axises due to initial values being symmetrical, the resultant meshes are not symmetrical. This is because Ω_F for the QOI function evaluation is not symmetrical. The AMR procedure refined mesh near the region of Ω_F. Therefore, it can be stated that the proposed AMR procedure successfully identifies the regions which are significant for the QOI function evaluation and refines near them correctly.

Figure 5.16 illustrates ϕ distribution in different initial time steps. The observed solution for ϕ is reduced with time advancement due to the diffusion. This leads to coarsening of the mesh with time advancement as depicted in figure 5.15 because QOI function evaluation is carried out with significantly diffused ϕ producing low values and the expected approximation error can be reached with reduced number of elements.



(a) $t = 0.002\,s$ (b) $t = 0.052\,s$ (c) $t = 0.102\,s$

Figure 5.15: Diffusion only with single hump solution AMR study meshes in different time steps $\theta_{rl} = 2$ and $\delta\tilde{F} = 5\times10^{-1}$.

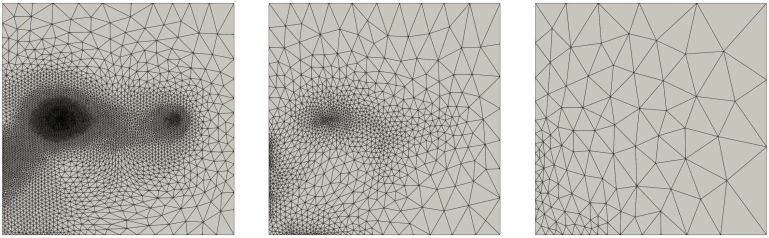


(a) $t = 0.002\,s$ (b) $t = 0.052\,s$ (c) $t = 0.102\,s$

Figure 5.16: ϕ distribution in single hump diffusion only solution AMR for different time steps $\theta_{rl} = 2$ and $\delta\tilde{F} = 5\times10^{-1}$.

The meshes of different time steps for double hump diffusion only experiment are illustrated in figure 5.17. It is evident that at $t = 0.002\,s$ (refer figure 5.17(a)) QOI function evaluation over Ω_F has contributions from both humps where more contributions are from the left most hump and relatively less contributions from right most hump, thereby higher refinement is carried out near the leftmost hump, and relatively lower refinement is carried out near the right most hump.

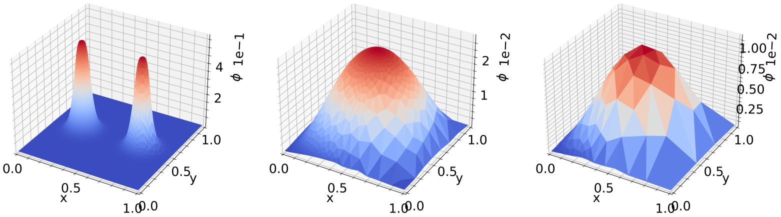
The mesh is coarsened with time marching as in the single hump experiment due to the fact that ϕ is being diffused as explained earlier. However, this experiment illustrates relatively higher coarsening of the mesh for same time steps than the single hump experiment. This can be attributed



(a) $t = 0.002\,s$ (b) $t = 0.052\,s$ (c) $t = 0.102\,s$

Figure 5.17: AMR study meshes in different time steps $\theta_{rl} = 2$ and $\delta\tilde{F} = 5\times10^{-1}$ for diffusion only problem with double hump solution.

to initial values being different as depicted in figure 5.14. There, single hump is initialized with comparatively larger values over larger Ω_F than its double hump experiment. This is also evident in the ϕ distributions presented in figure 5.18 for different time steps. The overall magnitude of the ϕ distribution is smaller in double hump experiment than in the single hump leading to a coarser mesh in double hump experiment. It is because, if the ϕ distribution is smaller, then domain integral over Ω_F is smaller then the expected approximation error can be reached with reduced number of elements.



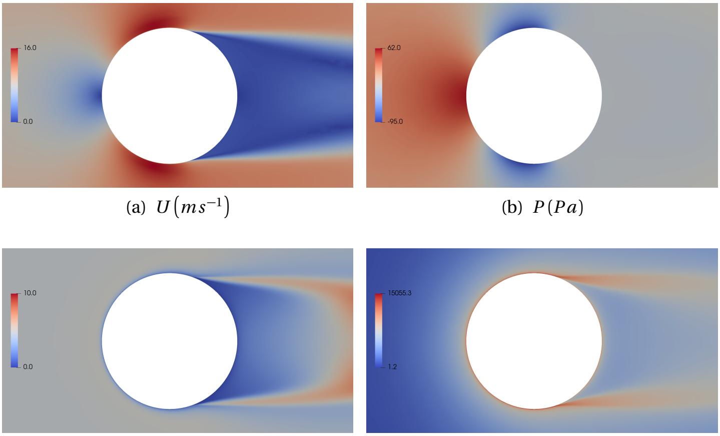
(a) $t = 0.002\,s$ (b) $t = 0.052\,s$ (c) $t = 0.102\,s$

Figure 5.18: ϕ distribution in double hump diffusion only solution AMR for different time steps $\theta_{rl} = 2$ and $\delta\tilde{F} = 5\times10^{-1}$.

5.3.3 2D flow over a cylinder

This experiment exhibits performance of proposed AMR procedure for highly non-linear steady state problems. It is done based on the specifications given in Catalano et al. [26], following the domain size description, and boundary descriptions. $k-\omega-sst$ turbulence model is used to obtain steady state flow field and then to perform the goal oriented AMR.

Figure 5.19 illustrates the solutions for aforementioned 2D flow over a cylinder problem. Figure 5.19(a) depicts velocity magnitude in the vicinity of the cylinder. Pressure distribution is shown in figure 5.19(b), turbulent kinetic energy(i.e. k) is shown in figure 5.19(c), and turbulent specific energy dissipation rate (i.e. ω) is shown in figure 5.19(d) in log scale.



(a) $U\left(ms^{-1}\right)$

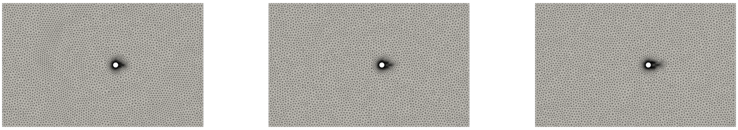
(b) $P(Pa)$

(c) $k\left(m^2s^{-2}\right)$

(d) $\omega\left(s^{-1}\right)$ in log scale

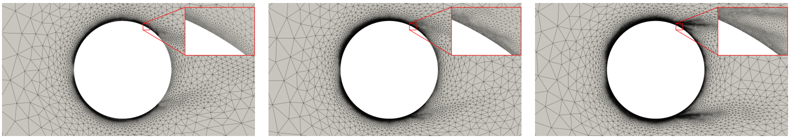
Figure 5.19: Solution of 2d flow over a cylinder experiment.

Figure 5.20 illustrates meshes obtained after each mesh refinement study for this experiment. Figure 5.21 illustrates the magnified view of the same iteration to better understand the behavior of AMR methodology. The mesh outside the vicinity of cylinder is kept almost same in all the iterations due to the allowed maximum element size restriction in the mesh refinement study. It is done to have a good quality mesh in the whole domain so the drag coefficients can be compared with the reference values from Catalano et al. [26].



(a) Iteration 2 (b) Iteration 3 (c) Iteration 4

Figure 5.20: Meshes for each iteration in 2d flow over cylinder AMR study.

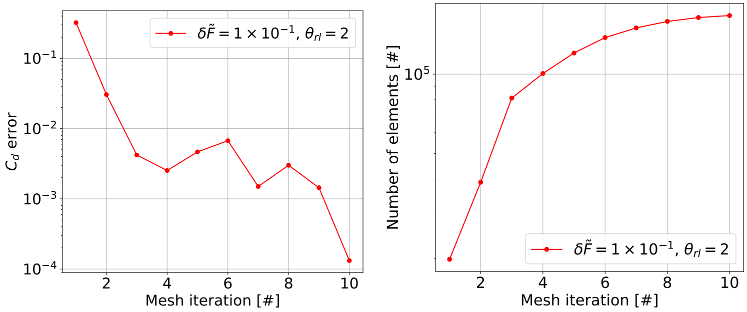


(a) Iteration 2 (b) Iteration 3 (c) Iteration 4

Figure 5.21: Magnified view of meshes for each iteration in 2d flow over cylinder AMR study.

From figure 5.21, it is evident that the mesh refinement is carried out near the front half of the cylinder. This refinement is continued from the stagnation point till the separation point (refer figure 5.19). This is because in order to calculate drag accurately, the numerical simulation needs to compute the momentum equation correctly which requires turbulent viscosity which is dependent on U, k, and ω. They exhibit prominent variations in region along the cylinder from stagnation point till the separation point, hence the refinement of mesh. It is also evident that each iteration refines elements in the vicinity of cylinder, with relatively higher refinement from stagnation point till the separation point. This can be attributed to the fact that, proposed AMR does not impose a lower bound on the error calculation and the solution field is not vanishing, therefore AMR will keep on refining the mesh.

Finally resultant C_d from each iteration is compared against RANS results for the same experiment found in Catalano et al. [26]. Figure 5.22(a) shows the error calculated w.r.t. the reference result after each iteration. The proposed AMR procedure is able to achieve desired QOI function approximation error $\delta\tilde{F}$ within 1 iteration. It is also evident that the error has a reducing trend with the increase in mesh iteration re-enforcing the fact that the proposed AMR procedure is developed without a lower bound for desired QOI function approximation error. Figure 5.22(b) illustrates the number of elements in each mesh refinement iteration.



(a) C_d error w.r.t. ref Catalano et al. [26]

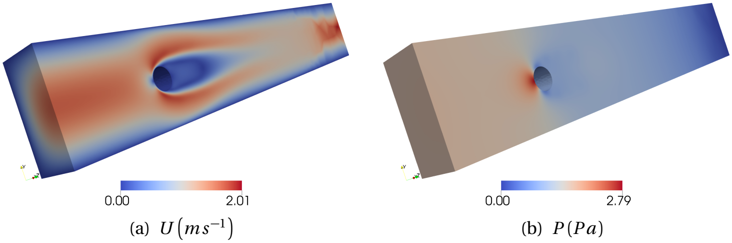
(b) Number of elements in each iteration

Figure 5.22: Steady state AMR study on 2d flow over cylinder experiment.

5.3.4 3D flow over a cylinder

Results of this experiment are investigated to understand proposed transient AMR procedure's ability to perform if an highly non-linear transient 3D problem is presented. Figure 5.23 illustrates velocity field distribution (refer figure 5.23(a)) and pressure field distribution (refer figure 5.23(b)) for transient inlet condition at $t = 2.17\,s$.

Figure 5.24 illustrates C_d distribution over the time with different mapping methods elaborated in section 5.1. Reference data is taken from Bayraktar et al. [14]. It can be observed that directly mapping current time step as in "nearest neighbor" and "nearest element" approaches produce oscillations at the mapping time step which are reduced over time. Oscillations are there because, C_d is computed by calculating reaction forces on each surface node on the cylinder. Reactions are computed by computing residual of the momentum equation. Since interpolation does not guarantee



(a) $U\left(ms^{-1}\right)$

(b) $P\left(Pa\right)$

Figure 5.23: 3D flow over a cylinder velocity and pressure distribution for transient inlet at $t = 2.17\,s$.

satisfaction of the residuals, hence the spikes are visible. This is relatively less significant in the case with "smoothed mapping" approach introduced in this thesis work. That is because, mapping is done in a previous time step (i.e. θ_{ms} steps before current step) and forward problem is run until the current time step. This allows solution field to build the solution while satisfying residuals for smoothing steps. "Smoothed mapping" is investigated with different θ_{ms} steps to identify effect of number of steps for prediction of QOI function value. It is evident that θ_{ms} does not have a strong influence over the final QOI function approximation.

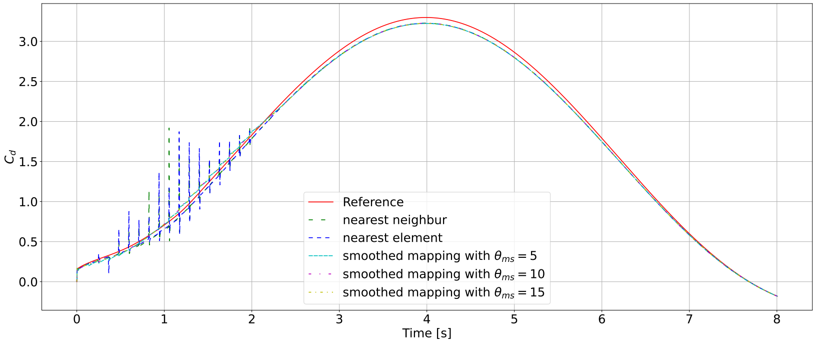


Figure 5.24: 3D flow over a cylinder C_d variation over the time for different mapping techniques.

Figure 5.25 illustrates QOI function approximation error variation with time averaging time window. "Nearest neighbor" and "nearest element" mapping techniques show steady oscillating behavior until the end of AMR period which is caused by the C_d spikes evident in figure 5.24. Afterwards, it shows almost steady error convergence. However, "smoothed mapping" techniques with different θ_{ms} values illustrate a different behavior. This is because, initially the C_d variation with time is below the reference (refer $t \in [0, 1]$ in figure 5.24). Then it increases beyond the reference value creating the first valley in figure 5.25. After the AMR period, C_d is again go below the reference value, consequently resulting in the second valley as depicted in figure 5.25.

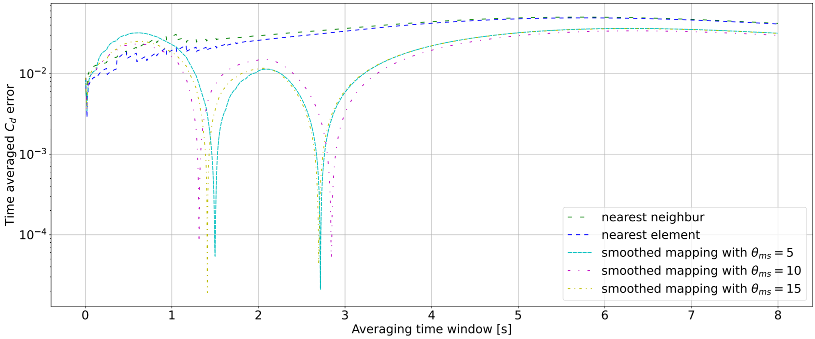


Figure 5.25: 3D flow over a cylinder time averaged C_d error variation over the time for different mapping techniques.

Even though results from mapping methodologies depicted in figure 5.24 show spikes when transient AMR is performed, the effect of spikes on overall QOI computation dies down with time. This is because, time averaged QOI is used in this case. It is important to make users aware that, in a high Re chaotic transient problem, these spikes may carry out artificial effects for a longer time than the experiment results shown in here. Therefore, it is important to understand the requirement to have a better mapping methodology to reduce these spikes. From figure 5.25, it is evident that, "smoothed mapping" technique provides better approximation of the QOI function for a given $\delta\tilde{F}$ value and shows less deviations from the reference value at positions where transient AMR is performed. Therefore, "smoothed mapping" technique is used hereafter.

Afterward, the transient AMR procedure is investigated for different $\delta\tilde{F}$ values. Then calculated C_d error with reference is illustrated in figure 5.26. Actual error (i.e. δF) is reducing with reduction in expected QOI approximation error (i.e. $\delta\tilde{F}$), with $\delta F \leq \delta\tilde{F}$ which illustrates transient AMR procedure's ability to obtain solutions satisfying the given approximation error. To illustrate this further, different meshes (obtained via anisotropic meshing methodology) for first three different transient AMR steps performed with different $\delta\tilde{F}$ values are depicted in table 5.1. It can be seen in all the cases, it identifies cylinder region as most important for drag computation, therefore it is refined more relative to other regions. In the case of $\delta\tilde{F} = 0.01$, transient AMR procedure identifies walls closer to the cylinder also has an effect towards computation of drag over the cylinder to obtained given $\delta\tilde{F}$, thus making refinement there as well.

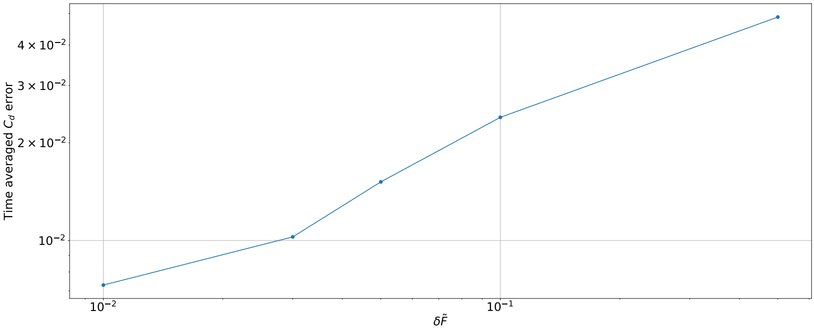


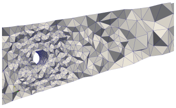
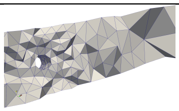
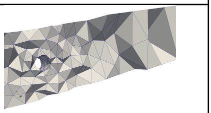
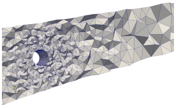
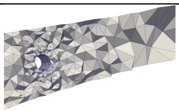
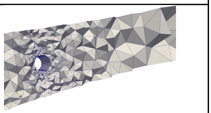
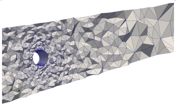
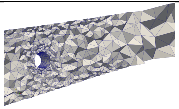
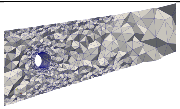
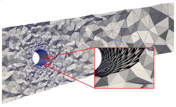
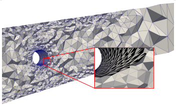
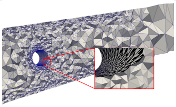
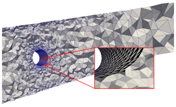
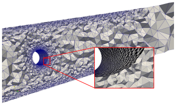
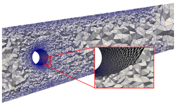
Figure 5.26: 3D flow over a cylinder C_d variation for different $\delta\tilde{F}$.

5.4 Conclusions and Outlook

Most of the time in CWE, numerical wind tunnel tests are performed to calculate one or many QOIs. The flow behavior of a CWE problem is transient and chaotic. Therefore, this chapter focused on developing methodologies to obtain optimized meshes to achieve given accuracy for a given QOI.

Firstly, AMR method for steady state problems were developed. Then they are evaluated against 2D scalar transport problems having analytical solu-

Table 5.1: Meshes after transient AMR procedure for different $\delta\tilde{F}$ values.

$\delta\tilde{F}$	Transient AMR step		
	1	2	3
0.5			
0.1			
0.05			
0.03			
0.01			

tions. Comparison with analytical solutions illustrated that proposed AMR method for steady state problems can achieve expected or better accuracy levels for given QOIs. Thereafter, the same methodology was tested against a highly non-linear 2D flow over a problem with $k-\omega-sst$ turbulence model. It also showed that, the given accuracies can be achieved with proposed AMR methodology.

Afterwards, transient AMR methodology was developed. Special care has to be taken when developing transient AMR methodology otherwise, the naive implementation requires prohibitively expensive memory when computing expected error values for given QOI. Finally proposed transient AMR methodology is used to solve a 2D scalar transient transport problem which has an analytical answer. Comparison with the analytical results showed, proposed transient AMR method also capable of achieving expected or better accuracies in QOI evaluations.

While developing transient AMR methodology, one crucial aspect also investigated which is mapping between coarse and refined meshes. Two different existing mapping methodologies and one novel "smoothed" mapping methodology were tested in a 3D low Re transient problem. It showed novel "smoothed" mapping methodology was able to perform better than other two mapping methodologies. It is important to note here that, other two mapping methodologies showed spikes at the points where transient AMR is performed, whereas "smoothed" mapping methodology only showed small deviations. This is important because, in a high Re chaotic transient problem, these artifacts may travel through time for a longer time affecting the overall solution. This makes it important to either make the time between two different transient AMR points longer, or to further investigate into better mapping methodologies which preserve conservation laws.

CHAPTER 6

CONCLUSIONS AND OUTLOOK

Predicting wind field around urban cities/structures poses many challenges in engineering. Computational Wind Engineering (CWE) is one branch of the engineering sciences which focuses on modelling wind effects around cities/structures in a numerical wind tunnel. It also poses many challenges because of the time investment required to properly define and solve the problem (i.e. defining assumptions, making mesh, solving). Therefore, this thesis focused on methodologies to reduce overall time taken for a numerical wind tunnel test. Chapter 1 gives a brief introduction to many challenges faced in preparing a numerical wind tunnel test for a CWE problem.

Chapter 2 introduces turbulence modelling in Finite Element Method (FEM) to reduce computational time required in evaluating a CWE problem. A novel Residual-based Flux-Corrected (RFC) stabilization method was introduced in this chapter to overcome difficulties faced in obtaining positivity preserving solutions for turbulence quantities used in two equations Reynolds Averaged Navier-Stokes (RANS) turbulence modelling. This novel RFC stabilization methodology was developed while keeping the possibility to easily extend it for adjoint sensitivity analysis, applicability in ALE frameworks for FSI analysis, and robustness. Firstly, the proposed stabilization method was tested against 2D scalar transport equations which have their analytical solutions. It showed that, novel RFC stabilization method was able to produce positivity preserving solutions and was giving better system matrix conditioning which is beneficial to be used in iterative linear solvers. Afterwards, it was tested against benchmark problems, obtained results were compared against their respective DNS or experimental data. It also showed, the proposed novel RFC stabilization method performs as expected giving accurate solution fields. Thereafter, developed stabilization method was extended to solve Unsteady Reynolds Averaged Navier-Stokes (URANS) problems and tested against a Fluid-Structure Interaction (FSI) benchmark case. Finally, the same stabilization method was used with different turbulence models namely RANS $k-\epsilon$, RANS $k-\omega-sst$ and URANS $k-\omega-sst$ to evaluate applicability of turbulence modelling in a CWE problem and compared against Large Eddy Simulation (LES) results. The results of this investigation showed, that URANS $k-\omega-sst$ turbulence model performed better and produced more accurate results. However, it also had short comings in replicating vortex shedding frequencies as in the reference solution.

The chapter 3 focused on developing steady state adjoint sensitivity analysis which is highly modular/extensible and efficient. The proposed developments in this chapter were then extended to obtain sensitivities for three different turbulence models namely $k-\epsilon$, $k-\omega$, and $k-\omega-sst$ with and without using the frozen turbulence assumption. Finally these developed sensitivities were used in different chaotic and non-chaotic flow optimization problems having drag as the minimization Quantities of Interest (QOI), with lift as a constraint. The results suggest that, having different turbulence models lead to different local minima in optimization procedures. The best results were shown in $k-\omega-sst$ turbulence model without the frozen turbulence assumption, which indicates the most accurate turbulence modelling methodology may produce the best optimized design when used in optimization problems.

Afterwards, chapter 4 extends proposed developments in chapter 2 and chapter 3 for transient sensitivity analysis. One major problem in transient sensitivity analysis is the butterfly effect, which causes chaotic transient systems to be highly sensitive for initial conditions. This causes computed transient adjoint solution to has an exponential growth in time, thus making them useless. Therefore, this chapter focused on introducing two novel approaches to stabilize these transient adjoint solutions. Time averaged primal was the first stabilization methodology which has proven to be providing stabilized adjoint sensitivities. However, the computed stabilized adjoint sensitivities showed dependence on the turbulence model used in estimating Reynolds Stress Tensor (RST) when time averaged solutions were used. This illustrated that, if a more accurate turbulence model was used, then more accurate adjoint sensitivities can be obtained. This warrants further investigations in implementing better accurate turbulence modelling methodologies. The next novel stabilization method was named artificial diffusion. It adds controlled artificial diffusion to mitigate exponential growth in transient adjoint solution. It was successful not only in stabilizing transient adjoint solution, but also in providing meaningful shape sensitivities which were having reduced errors with the sensitivities computed via finite difference method. Afterwards, this stabilization method was utilized in an optimization problem. This optimization problem involved reducing amplitudes of maximum vortex shedding frequencies in a given range for a chaotic transient Benchmark on the Aerodynamics of a Rectangular 5:1 Cylinder (BARC) problem. The proposed artificial diffusion stabilization problem was able to provide meaningful sensitivities, making the optimization problem to have the QOI reduced.

Chapter 5 developed goal oriented Adaptive Mesh Refinement (AMR) methodologies for steady state and transient problems using adjoint approach to obtain error sensitivities for a given QOI. First, methodology for steady state problems was developed. Then it was tested against 2D scalar transport problems and a highly non-linear high Re flow problem using $k-\omega-sst$ as the turbulence model. Results from these studies depicted that, the proposed steady state AMR methodology can provide meshes which lead to expected accuracy levels in QOI. Then this steady state methodology was extended for transient AMR methodology. In here, special care has been taken to reduce overall memory consumption due to requirement in refined mesh to obtain QOI error sensitivities. Two different existing mapping methodologies and one novel "smoothed" mapping methodology were also used in this chapter to map coarse mesh field solution to refined mesh field when using AMR. Two existing mapping methodologies showed spikes in QOI at points when transient AMR were performed. These spikes were then diffused with time since time averaged QOI was used. However, the proposed novel "smoothed" mapping methodology showed only small deviations at points where transient AMR were performed indicating its better performance in mapping. Finally proposed transient AMR methodology was tested against a 2D scalar transient transport equation problem and a 3D flow problem. Results from both illustrated that proposed transient AMR methodology were able to produce meshes which can lead to expected accuracy levels in the used QOI.

Finally it can be stated that, this thesis successfully addressed the problem of requiring substantial amount of time to perform a numerical wind tunnel test in CWE by breaking down the problem in small sections such as reducing time consumption via developed turbulence modelling, reducing time spent on design analysis via developed shape optimization methodologies, and reducing time spent on designing and making spatial discretizations by developed steady state and transient AMR methodologies. All of these developments were thoroughly tested and validated using examples from literature and internal work. The results of these testing illustrated, that the proposed developments are performing as expected, thus reducing the overall time consumption in performing a numerical wind tunnel test.

As for the outlook, even though proof of concept for the proposed novel RFC stabilization is validated for non-turbulent FSI problem, the method can be used to solve turbulent FSI problems as well. Therefore, further studies can be carried out to evaluate novel RFC stabilization method’s effectiveness in solving turbulent FSI problems. Furthermore, future studies on the proposed time averaged transient adjoint stabilization method warrants for more accurate turbulence models such as machine learning based, three equations based, four equations based, etc. Therefore, novel RFC stabilization with developed modular sensitivity analysis methodology can be extended for aforementioned high accuracy turbulence models to investigate their ability to provide sensible sensitivity values for chaotic flow problems. Thereafter, further research can be undertaken to extend novel RFC stabilization method and its modular sensitivity analysis for analyzing wind-thermal related problems by implementing energy equation which will open whole new branch of transient sensitivity analysis for wind problems.

APPENDIX A

STATISTICAL QUANTITIES

The definitions for statistical quantities for a given ϕ^n is given below.

A.1 Mean

Mean of ϕ^n is computed as

$$\overline{\phi}^N = \frac{1}{N}\sum_{n=1}^{N}[\phi^n] \tag{A.1}$$

A continuous mean can be computed for a transient forward time marching problem as

$$\overline{\phi}^{N+1} = \frac{N\overline{\phi}^N + \phi^{N+1}}{N+1} \quad \textit{where} \quad \overline{\phi}^0 = 0 \quad \textit{and} \quad N \geq 0 \tag{A.2}$$

A.2 Variance

Variance of ϕ^n is computed as

$$Var(\phi)^N = \frac{1}{N}\sum_{n=1}^{N}\left[\left(\phi^n\right)^2\right] - \left(\overline{\phi}^N\right)^2 \tag{A.3}$$

A continuous variance can be computed for a transient forward time marching problem as

$$Var(\phi)^{N+1} = \frac{N\left[Var(\phi)^N + \left(\overline{\phi}^N\right)^2\right] + \left(\phi^{N+1}\right)^2}{N+1} - \left(\overline{\phi}^{N+1}\right)^2 \tag{A.4}$$

A.3 Standard Deviation

Standard deviation of ϕ^n is computed as follows

$$\sigma\left(\phi\right)^N = \sqrt{Var(\phi)^N} \tag{A.5}$$

Appendix B

Discrete Transient Adjoint Formulation

B.1 Primal Problem

Primal problem's discrete residual is defined as in equation (B.1) using Bossak time integration scheme.

$$\underline{R}^n = \underline{F}^n - \underline{\mathbf{C}}^n \underline{w}^n - \underline{\mathbf{M}}^n \left[\left(1-\alpha_b\right) \underline{\dot{w}}^n + \alpha_b \underline{\dot{w}}^{n-1} \right] \tag{B.1}$$

Where $\underline{F}^n$, $\underline{\mathbf{C}}^n \underline{w}^n$, $\underline{\mathbf{M}}^n$ are discrete time instantaneous force vector, convective matrix and mass matrix. $\underline{w}^n$ is the time instantaneous solution vector and $\underline{\dot{w}}^n$ is the time derivative of time instantaneous solution vector. α_b is the Bossak constant. $\underline{\dot{w}}^n$ is obtained by using the equation (B.2) where γ_b is the newmark constant calculated from $\gamma_b = 0.5 - \alpha_b$.

$$\underline{H}^n = \underline{\dot{w}}^n - \frac{1}{\Delta t \gamma_b}\left(\underline{w}^n - \underline{w}^{n-1}\right) - \frac{\gamma_b - 1}{\gamma_b}\underline{\dot{w}}^{n-1} = \underline{0} \tag{B.2}$$

B.2 Quantity of Interest

Time averaged (or weighted averaged) QOI is illustrated in equation (B.3) where s is the shape parameter.

$$\overline{J} = \frac{1}{N}\sum_{n=1}^{N} J\left(\underline{w}^{n}, s\right) = \sum_{n=1}^{N} J^{n} \tag{B.3}$$

B.3 Adjoint Problem

Hence the Lagrangian can be formulated as in equation (B.4).

$$L = \sum_{n=1}^{N}\left[L^{n}\right] = \sum_{n=1}^{N}\left[J^{n} + \left(\underline{\lambda}_{1}^{n}\right)^{T}\underline{R}^{n} + \left(\underline{\lambda}_{2}^{n}\right)^{T}\underline{H}^{n}\right] \tag{B.4}$$

Equation (B.5) illustrates the total derivative w.r.t s of the formulated Lagrangian.

$$\begin{aligned}
\frac{dL}{ds} = \sum_{n=1}^{N}\left[\frac{dL}{ds}^{n}\right] = & \sum_{n=1}^{N}\left[\frac{\partial J}{\partial s}^{n} + \left(\underline{\lambda}_{1}^{n}\right)^{T}\frac{\partial \underline{R}}{\partial s}^{n} + \left(\underline{\lambda}_{2}^{n}\right)^{T}\frac{\partial \underline{H}}{\partial s}^{n}\right] \\
& + \sum_{n=1}^{N}\left[\frac{\partial J^{n}}{\partial \underline{w}^{n}} + \left(\underline{\lambda}_{1}^{n}\right)^{T}\frac{\partial \underline{R}^{n}}{\partial \underline{w}^{n}} + \left(\underline{\lambda}_{2}^{n}\right)^{T}\frac{\partial \underline{H}^{n}}{\partial \underline{w}^{n}}\right]\frac{d\underline{w}}{ds}^{n} \\
& + \sum_{n=1}^{N}\left[\left(\underline{\lambda}_{1}^{n}\right)^{T}\frac{\partial \underline{R}^{n}}{\partial \underline{\dot{w}}^{n}} + \left(\underline{\lambda}_{2}^{n}\right)^{T}\frac{\partial \underline{H}^{n}}{\partial \underline{\dot{w}}^{n}}\right]\frac{d\underline{\dot{w}}}{ds}^{n} \\
& + \sum_{n=1}^{N}\left[\left(\underline{\lambda}_{2}^{n}\right)^{T}\frac{\partial \underline{H}^{n}}{\partial \underline{w}^{n-1}}\right]\frac{d\underline{w}}{ds}^{n-1} \\
& + \sum_{n=1}^{N}\left[\left(\underline{\lambda}_{1}^{n}\right)^{T}\frac{\partial \underline{R}^{n}}{\partial \underline{\dot{w}}^{n-1}} + \left(\underline{\lambda}_{2}^{n}\right)^{T}\frac{\partial \underline{H}^{n}}{\partial \underline{\dot{w}}^{n-1}}\right]\frac{d\underline{\dot{w}}}{ds}^{n-1}
\end{aligned} \tag{B.5}$$

Equation (B.5) can be simplified as depicted in equation (B.6) using $\frac{d\underline{w}}{ds}^{0} = \underline{0}$, $\frac{d\underline{\dot{w}}}{ds}^{0} = \underline{0}$, $\underline{\lambda}_{1}^{N+1} = \underline{0}$, $\underline{\lambda}_{2}^{N+1} = \underline{0}$.

$$\frac{dL}{ds}^{n} = \frac{\partial J}{\partial s}^{n} + \left(\underline{\lambda}_{1}^{n}\right)^{T}\frac{\partial \underline{R}}{\partial s}^{n} + \left(\underline{\lambda}_{2}^{n}\right)^{T}\frac{\partial \underline{H}}{\partial s}^{n} + \underline{\tilde{R}}_{1}^{T}\frac{d\underline{w}}{ds}^{n} + \underline{\tilde{R}}_{2}^{T}\frac{d\underline{\dot{w}}}{ds}^{n} \tag{B.6}$$

$\underline{\tilde{R}}_1^T$ and $\underline{\tilde{R}}_2^T$ is defined in equation (B.7a) and equation (B.7b).

$$\underline{\tilde{R}}_1^T = \frac{\partial J^n}{\partial \underline{w}^n} + \left(\underline{\lambda}_1^n\right)^T \frac{\partial \underline{R}^n}{\partial \underline{w}^n} + \left(\underline{\lambda}_2^n\right)^T \frac{\partial \underline{H}^n}{\partial \underline{w}^n} + \left(\underline{\lambda}_2^{n+1}\right)^T \frac{\partial \underline{H}^{n+1}}{\partial \underline{w}^n} \tag{B.7a}$$

$$\underline{\tilde{R}}_2^T = \left(\underline{\lambda}_1^n\right)^T \frac{\partial \underline{R}^n}{\partial \underline{\dot{w}}^n} + \left(\underline{\lambda}_2^n\right)^T \frac{\partial \underline{H}^n}{\partial \underline{\dot{w}}^n} + \left(\underline{\lambda}_1^{n+1}\right)^T \frac{\partial \underline{R}^{n+1}}{\partial \underline{\dot{w}}^n} + \left(\underline{\lambda}_2^{n+1}\right)^T \frac{\partial \underline{H}^{n+1}}{\partial \underline{\dot{w}}^n} \tag{B.7b}$$

In the discrete adjoint approach, $\underline{\tilde{R}}_1 = \underline{0}$ and $\underline{\tilde{R}}_2 = \underline{0}$ is assumed to avoid expensive evaluation of total state derivatives (i.e. $\frac{d\underline{w}^n}{ds}, \frac{d\underline{\dot{w}}^n}{ds}$) when calculating adjoint solutions (i.e. $\underline{\lambda}_1^n, \underline{\lambda}_2^n$). Simplifying equation (B.7b) will result in the definition for $\underline{\lambda}_2^n$ as depicted in equation (B.8).

$$\underline{\lambda}_2^n = -\left(\frac{\partial \underline{R}^n}{\partial \underline{\dot{w}}^n}\right)^T \underline{\lambda}_1^n - \left(\frac{\partial \underline{R}^{n+1}}{\partial \underline{\dot{w}}^n}\right)^T \underline{\lambda}_1^{n+1} + \frac{\gamma_b - 1}{\gamma_b} \underline{\lambda}_2^{n+1} \tag{B.8}$$

Applying equation (B.8) to equation (B.7a) will result in the final transient adjoint system of equations as depicted in equation (B.9).

$$\left[\frac{\partial \underline{R}^n}{\partial \underline{w}^n} + \frac{1}{\Delta t \gamma_b} \frac{\partial \underline{R}^n}{\partial \underline{\dot{w}}^n}\right]^T \underline{\lambda}_1^n = -\left(\frac{\partial J^n}{\partial \underline{w}^n}\right)^T - \frac{1}{\Delta t \gamma_b} \left(\frac{\partial \underline{R}^{n+1}}{\partial \underline{\dot{w}}^n}\right)^T \underline{\lambda}_1^{n+1} - \frac{1}{\Delta t \gamma_b^2} \underline{\lambda}_2^{n+1} \tag{B.9}$$

List of Figures

List of Tables

Bibliography

[1] G Ahmadi. “A two-equation turbulence model for compressible flows based on the second law of thermodynamics.” In: *Journal of Non-Equilibrium Thermodynamics* 14.1 (1989), pp. 45–60. DOI: 10.1515/jnet.1989.14.1.45.

[2] B. R. Ahrabi, W. K. Anderson, and J. C. Newman III. “An adjoint-based hp-adaptive stabilized finite-element method with shock capturing for turbulent flows.” In: *Computer Methods in Applied Mechanics and Engineering* 318 (2017), pp. 1030–1065. DOI: 10.1016/j.cma.2017.02.001.

[3] M. Ainsworth and J. T. Oden. “A posteriori error estimation in finite element analysis.” In: *Computer methods in applied mechanics and engineering* 142.1-2 (1997), pp. 1–88. DOI: 10.1016/S0045-7825(96)01107-3.

[4] W. K. Anderson and V Venkatakrishnan. “Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation.” In: *Computers & Fluids* 28.4-5 (1999), pp. 443–480. DOI: 10.1016/S0045-7930(98)00041-3.

[5] M. Andre, M. Peéntek, K.-U. Bletzinger, and R. Wüchner. “Aeroelastic simulation of the wind-excited torsional vibration of a parabolic trough solar collector.” In: *Journal of Wind Engineering and Industrial Aerodynamics* 165 (2017), pp. 67–78. DOI: 10.1016/j.jweia.2017.03.005.

[6] I. ANSYS. *Wall Boundary Conditions*. 2009. URL: https://www.afs.enea.it/project/neptunius/docs/fluent/html/th/node68.htm.

[7] T. Apel, O. Benedix, D. Sirch, and B. Vexler. “A priori mesh grading for an elliptic problem with Dirac right-hand side.” In: *SIAM journal on numerical analysis* 49.3 (2011), pp. 992–1005. DOI: 10.1137/090778018.

[8] L. Arriola and J. M. Hyman. “Sensitivity Analysis for Uncertainty Quantification in Mathematical Models.” In: *Mathematical and Statistical Estimation Approaches in Epidemiology*. Springer Netherlands, 2009, pp. 195–247. DOI: 10.1007/978-90-481-2313-1_10.

[9] I. Babuška and W. C. Rheinboldt. “A-posteriori error estimates for the finite element method.” In: *International Journal for Numerical Methods in Engineering* 12.10 (1978), pp. 1597–1615. DOI: 10.1002/nme.1620121010.

[10] A. Balan, M. A. Park, and W. K. Anderson. “Adjoint-based Anisotropic Mesh Adaptation for a Stabilized Finite-Element Flow Solver.” In: *AIAA Aviation 2019 Forum*. 2019, p. 2949. DOI: 10.2514/6.2019-2949.

[11] B. Baldwin and H. Lomax. “Thin-layer approximation and algebraic model for separated turbulentflows.” In: *16th Aerospace Sciences Meeting*. 1978. DOI: 10.2514/6.1978-257.

[12] G. R. Barrenechea, V. John, and P. Knobloch. “Analysis of Algebraic Flux Correction Schemes.” In: *SIAM Journal on Numerical Analysis* 54.4 (2016), pp. 2427–2451. DOI: 10.1137/15M1018216.

[13] D. Baumgärtner, A. Viti, A. Dumont, G. Carrier, and K.-U. Bletzinger. “Comparison and combination of experience-based parametrization with Vertex Morphing in aerodynamic shape optimization of a forward-swept wing aircraft.” In: *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2016, p. 3368. DOI: 10.2514/6.2016-3368.

[14] E. Bayraktar, O. Mierka, and S. Turek. “Benchmark computations of 3D laminar flow around a cylinder with CFX, OpenFOAM and FeatFlow.” In: *International Journal of Computational Science and Engineering* 7.3 (2012), pp. 253–266. DOI: 10.1504/IJCSE.2012.048245.

[15] M. Bielecki, M. Karcz, W. Radulski, and J. Badur. "Thermo-mechanical coupling between the flow of steam and deformation of the valve during start-up of the 200 MW turbine." In: *Task Quarterly* 5 (2001), pp. 125–140.

[16] K.-U. Bletzinger. "Shape optimization." In: *Encyclopedia of Computational Mechanics Second Edition* (2017), pp. 1–42. DOI: 10.1002/9781119176817.ecm2109.

[17] P. J. Blonigan, S. A. Gomez, and Q. Wang. "Least squares shadowing for sensitivity analysis of turbulent fluid flows." In: *52nd Aerospace Sciences Meeting*. 2014, p. 1426. DOI: 10.2514/6.2014-1426.

[18] P. J. Blonigan and Q. Wang. "Multiple shooting shadowing for sensitivity analysis of chaotic dynamical systems." In: *Journal of Computational Physics* 354 (2018), pp. 447–475. DOI: 10.1016/j.jcp.2017.10.032.

[19] P. T. Boggs, A. Althsuler, A. R. Larzelere, E. J. Walsh, R. L. Clay, and M. F. Hardwick. "DART system analysis." In: (Aug. 2005). DOI: 10.2172/876325.

[20] T Bouhal, Y Agrouaz, T Kousksou, T El Rhafiki, Y Zeraouli, et al. "Performance optimization of a two-phase closed thermosyphon through CFD numerical simulations." In: *Applied Thermal Engineering* 128 (2018), pp. 551–563. DOI: 10.1016/j.applthermaleng.2017.09.049.

[21] A. N. Brooks and T. J. Hughes. "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations." In: *Computer Methods in Applied Mechanics and Engineering* 32.1-3 (1982), pp. 199–259. DOI: 10.1016/0045-7825(82)90071-8.

[22] L. Bruno, M. V. Salvetti, and F. Ricciardelli. "Benchmark on the aerodynamics of a rectangular 5: 1 cylinder: an overview after the first four years of activity." In: *Journal of Wind Engineering and Industrial Aerodynamics* 126 (2014), pp. 87–106. DOI: 10.1016/j.jweia.2014.01.005.

[23] E. Burman and A. Ern. "Nonlinear diffusion and discrete maximum principle for stabilized Galerkin approximations of the convection--diffusion-reaction equation." In: *Computer Methods in Applied Mechanics and Engineering* 191.35 (2002), pp. 3833–3855. DOI: 10.1016/S0045-7825(02)00318-3.

[24] E. G. D. do Carmo and G. B. Alvarez. "A new stabilized finite element formulation for scalar convection--diffusion problems: the streamline and approximate upwind/Petrov--Galerkin method." In: *Computer Methods in Applied Mechanics and Engineering* 192.31-32 (2003), pp. 3379–3396. DOI: 10.1016/S0045-7825(03)00292-5.

[25] P. Catalano and M. Amato. "An evaluation of RANS turbulence modelling for aerodynamic applications." In: *Aerospace Science and Technology* 7.7 (2003), pp. 493–509. DOI: 10.1016/S1270-9638(03)00061-0.

[26] P. Catalano, M. Wang, G. Iaccarino, and P. Moin. "Numerical simulation of the flow around a circular cylinder at high Reynolds numbers." In: *International journal of heat and fluid flow* 24.4 (2003), pp. 463–469. DOI: 10.1016/S0142-727X(03)00061-4.

[27] J. Cerutti, G. Cafiero, and G. Iuso. "Aerodynamic drag reduction by means of platooning configurations of light commercial vehicles: A flow field analysis." In: *International Journal of Heat and Fluid Flow* 90 (2021), p. 108823. DOI: 10.1016/j.ijheatfluidflow.2021.108823.

[28] N. Chandramoorthy, P. Fernandez, C. Talnikar, and Q. Wang. "An Analysis of the Ensemble Adjoint Approach to Sensitivity Analysis in Chaotic Systems." In: *23rd AIAA Computational Fluid Dynamics Conference*. 2017, p. 3799. DOI: 10.2514/6.2017-3799.

[29] M. Chater, A. Ni, P. J. Blonigan, and Q. Wang. "Least squares shadowing method for sensitivity analysis of differential equations." In: *SIAM Journal on Numerical Analysis* 55.6 (2017), pp. 3030–3046. DOI: 10.1137/15M1039067.

[30] M. Chater, A. Ni, and Q. Wang. "Simplified least squares shadowing sensitivity analysis for chaotic ODEs and PDEs." In: *Journal of Computational Physics* 329 (2017), pp. 126–140. DOI: 10.1016/j.jcp.2016.10.035.

[31] D. Chenais. “On the existence of a solution in a domain identification problem.” In: *Journal of Mathematical Analysis and Applications* 52.2 (1975), pp. 189–219. DOI: 10.1016/0022-247X(75)90091-8.

[32] K.-Y. Chien. “Predictions of channel and boundary-layer flows with a low-Reynolds-number turbulence model.” In: *AIAA journal* 20.1 (1982), pp. 33–38. DOI: 10.2514/3.51043.

[33] R. Codina. “A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection-diffusion equation.” In: *Computer Methods in Applied Mechanics and Engineering* 110.3-4 (1993), pp. 325–342. DOI: 10.1016/0045-7825(93)90213-H.

[34] R. Codina. “Pressure stability in fractional step finite element methods for incompressible flows.” In: *Journal of Computational Physics* 170.1 (2001), pp. 112–140. DOI: 10.1006/jcph.2001.6725.

[35] J. Cotela Dalmau and R. Rossi. *Applications of turbulence modeling in civil engineering*. International Centre for Numerical Methods in Engineering (CIMNE), 2016.

[36] P. Dadvand, R. Rossi, M. Gil, X. Martorell, J. Cotela, E. Juanpere, S. Idelsohn, and E. Onate. “Migration of a generic multi-physics framework to HPC environments.” In: *Computers and Fluids* 80 (2013). Selected Contributions of the 23rd International Conference on Parallel Fluid Dynamics ParCFD2011, pp. 301 –309. DOI: 10.1016/j.compfluid.2012.02.004.

[37] P. Dadvand, R. Rossi, and E. Onate. “An object-oriented environment for developing finite element codes for multi-disciplinary applications.” In: *Archives of computational methods in engineering* 17.3 (2010), pp. 253–297. DOI: 10.1007/s11831-010-9045-2.

[38] T. Dhert, T. Ashuri, and J. R. R. A. Martins. “Aerodynamic shape optimization of wind turbine blades using a Reynolds-averaged Navier–Stokes model and an adjoint method.” In: *Wind Energy* 20.5 (2017), pp. 909–926. DOI: 10.1002/we.2070.

[39] R. P. Dickinson and R. J. Gelinas. “Sensitivity analysis of ordinary differential equation systems—a direct method.” In: *Journal of computational physics* 21.2 (1976), pp. 123–143. DOI: `10.1016/0021-9991(76)90007-3`.

[40] O. M. Essenwanger. “Elements of statistical analysis.” In: (1986).

[41] G. Eyink, T. Haine, and D. Lea. “Ruelle’s linear response formula, ensemble adjoint schemes and Lévy flights.” In: *Nonlinearity* 17.5 (2004), p. 1867. DOI: `10.1088/0951-7715/17/5/016`.

[42] C. S. Ferreira and B. Geurts. “Aerofoil optimization for vertical-axis wind turbines.” In: *Wind Energy* 18.8 (2015), pp. 1371–1385.

[43] L. Ferres. “Memory management in C: The heap and the stack.” In: *Universidad de Concepcion* (2010).

[44] K. J. Fidkowski and D. L. Darmofal. “Review of output-based error estimation and mesh adaptation in computational fluid dynamics.” In: *AIAA journal* 49.4 (2011), pp. 673–694. DOI: `10.2514/1.J050073`.

[45] A. R. Firoozjaee and M. H. Afshar. “Steady-state solution of incompressible Navier-Stokes equations using discrete least-squares meshless method.” In: *International Journal for Numerical Methods in Fluids* 67.3 (2011), pp. 369–382. DOI: `10.1002/fld.2370`.

[46] L. Frazza, A. Loseille, and F. Alauzet. “Anisotropic mesh adaptation for turbomachinery applications.” In: *23rd AIAA Computational Fluid Dynamics Conference*. 2017, p. 3299. DOI: `10.2514/6.2017-3299`.

[47] D. Fröhlich, M. Gangwisch, and A. Matzarakis. “Effect of radiation and wind on thermal comfort in urban environments - Application of the RayMan and SkyHelios model.” In: *Urban Climate* 27 (2019), pp. 1–7. DOI: `10.1016/j.uclim.2018.10.006`.

[48] S. W. Funke, S. C. Kramer, and M. D. Piggott. “Design optimisation and resource assessment for tidal-stream renewable energy farms using a new continuous turbine approach.” In: *Renewable energy* 99 (2016), pp. 1046–1061. DOI: `10.1016/j.renene.2016.07.039`.

[49] A. C. Galeão and E. G. D. Do Carmo. “A consistent approximate upwind Petrov-Galerkin method for convection-dominated problems.” In: *Computer Methods in Applied Mechanics and Engineering* 68.1 (1988), pp. 83–95. DOI: 10.1016/0045-7825(88)90108-9.

[50] G. Gan. “A parametric study of Trombe walls for passive cooling of buildings.” In: *Energy and buildings* 27.1 (1998), pp. 37–43. DOI: 10.1016/S0378-7788(97)00024-8.

[51] A. Geiser, I. Antonau, and K.-U. Bletzinger. “Aggregated Formulation of geometric Constraints for Node-Based Shape Optimization with Vertex Morphing.” In: *EUROGEN 2021* 3 (2021), p. 269. DOI: 10.7712/140121.7952.18383.

[52] I. Giosan and P Eng. “Vortex shedding induced loads on free standing structures.” In: *Structural Vortex Shedding Response Estimation Methodology and Finite Element Simulation* 42 (2013).

[53] M Gourma, N Jia, and C Thompson. “Adaptive mesh refinement for two-phase slug flows with an a priori indicator.” In: *International journal of multiphase flow* 49 (2013), pp. 83–98. DOI: 10.1016/j.ijmultiphaseflow.2012.08.003.

[54] H. Grotjans, F. R. Menter, and K. D. Papailiou. “Wall Functions for General Application CFD Codes, 4th, Vol 1 pts 1-2; Proceedings of the fourth European computational fluid dynamics conference.” In: *COMPUTATIONAL FLUID DYNAMICS -EUROPEAN CONFERENCE-, Vol 1 pts 1-2; Proceedings of the fourth European computational fluid dynamics conference, 4th, Vol 1 pts 1-2; Proceedings of the fourth European computational fluid dynamics conference*. Vol. 1. 2. Chichester: Wiley, 1998, pp. 1112–1117. ISBN: 0471985791.

[55] M. D. Gunzburger. “Sensitivities, adjoints and flow optimization.” In: *International Journal for Numerical Methods in Fluids* 31.1 (1999), pp. 53–78. DOI: 10.1002/(SICI)1097-0363(19990915)31:1%3C53::AID-FLD955%3E3.0.CO;2-Z.

[56] C. Harris, D Roekaerts, F. Rosendal, F. Buitendijk, P. Daskopoulos, A. Vreenegoor, and H Wang. “Computational fluid dynamics for chemical reactor engineering.” In: *Chemical Engineering Science*

51.10 (1996), pp. 1569–1594. DOI: 10.1016/0009-2509(96)00021-8.

[57] A. Harten. “High resolution schemes for hyperbolic conservation laws.” In: *Journal of Computational Physics* 135.2 (1997), pp. 260–278. DOI: 10.1016/0021-9991(83)90136-5.

[58] R. M. Hicks and P. A. Henne. “Wing design by numerical optimization.” In: *Journal of Aircraft* 15.7 (1978), pp. 407–412.

[59] U. Högström. “Von Karman’s constant in atmospheric boundary layer flow: Reevaluated.” In: *Journal of the Atmospheric Sciences* 42.3 (1985), pp. 263–270. DOI: 10.1175/1520-0469(1985)042%3C0263:VKCIAB%3E2.0.CO;2.

[60] S. Hoyas and J. Jiménez. “Reynolds number effects on the Reynolds-stress budgets in turbulent channels.” In: *Physics of Fluids* 20.10 (2008), p. 101511. DOI: 10.1063/1.3005862.

[61] S. Hoyas and J. Jiménez. “Scaling of the velocity fluctuations in turbulent channels up to Re τ= 2003.” In: *Physics of Fluids* 18.1 (2006), p. 011702. DOI: 10.1063/1.2162185.

[62] M. Hubbard. “Non-Oscillatory Third Order Fluctuation Splitting Schemes for Steady Scalar Conservation Laws.” In: *J. Comput. Phys.* 222.2 (Mar. 2007), 740–768. DOI: 10.1016/j.jcp.2006.08.007.

[63] T. J. Hughes and M. Mallet. “A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems.” In: *Computer Methods in Applied Mechanics and Engineering* 58.3 (1986), pp. 305 –328. DOI: 10.1016/0045-7825(86)90152-0.

[64] T. J. Hughes and M. Mallet. “A new finite element formulation for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems.” In: *Computer Methods in Applied Mechanics and Engineering* 58.3 (1986), pp. 329 –336. DOI: 10.1016/0045-7825(86)90153-2.

[65] S. R. Idelsohn, E. Onate, N. Calvo, and F. Del Pin. “The meshless finite element method.” In: *International Journal for Numerical Methods in Engineering* 58.6 (2003), pp. 893–912. DOI: 10.1002/nme.798.

[66] F. Ilinca and D. Pelletier. “Positivity Preservation and Adaptive Solution for the k-epsilon Model of Turbulence.” In: *AIAA Journal* 36.1 (1998), pp. 44–50. DOI: `10.2514/2.350`.

[67] P. A. Irwin. “Vortices and tall buildings: A recipe for resonance.” In: *Phys. Today* 63.9 (2010), pp. 68–69.

[68] A. Jameson, L. Martinelli, and N. A. Pierce. “Optimum aerodynamic design using the Navier--Stokes equations.” In: *Theoretical and computational fluid dynamics* 10.1 (1998), pp. 213–237. DOI: `10.1007/s001620050060`.

[69] H. Jasak. “Error analysis and estimation for the finite volume method with applications to fluid flows.” PhD thesis. Imperial College London (University of London), 1996.

[70] A. Jha and V. John. “A study of solvers for nonlinear AFC discretizations of convection--diffusion equations.” In: *Computers & Mathematics with Applications* 78.9 (2019), pp. 3117–3138. DOI: `10.1016/j.camwa.2019.04.020`.

[71] J. Johansen and N. N. Sørensen. “Aerodynamic investigation of winglets on wind turbine blades using CFD.” In: (2006).

[72] V. John and E. Schmeyer. “Finite element methods for time-dependent convection--diffusion--reaction equations with small diffusion.” In: *Computer Methods in Applied Mechanics and Engineering* 198.3-4 (2008), pp. 475–494. DOI: `10.1016/j.cma.2008.08.016`.

[73] V. Joshi and R. K. Jaiman. “A positivity preserving variational method for multi-dimensional convection--diffusion--reaction equation.” In: *Journal of Computational Physics* 339 (2017), pp. 247–284. DOI: `10.1016/j.jcp.2017.03.005`.

[74] S. Jovic and D. M. Driver. “Backward-facing step measurements at low Reynolds number, Re (sub h)= 5000.” In: (1994).

[75] A Korobenko, M.-C. Hsu, I Akkerman, J Tippmann, and Y Bazilevs. “Structural mechanics modeling and FSI simulation of wind turbines.” In: *Mathematical Models and Methods in Applied Sciences* 23.02 (2013), pp. 249–272. DOI: `10.1142/S0218202513400034`.

[76] D. Kuzmin. "Algebraic flux correction for finite element discretizations of coupled systems." In: *Computational Methods for Coupled Problems in Science and Engineering II, CIMNE, Barcelona* (2007), pp. 653–656.

[77] "Algebraic Flux Correction I." In: *Flux-Corrected Transport: Principles, Algorithms, and Applications*. Ed. by D. Kuzmin, R. Löhner, and S. Turek. Dordrecht: Springer Netherlands, 2012, pp. 145–192. ISBN: 978-94-007-4038-9. DOI: 10.1007/978-94-007-4038-9_6.

[78] D. Kuzmin, R. Löhner, and S. Turek. *Flux-corrected Transport: Principles, Algorithms, and Applications*. Springer, 2012. ISBN: 978-94-007-4038-9.

[79] D. Kuzmin, O. Mierka, and S. Turek. *On the implementation of the k-epsilon turbulence model in incompressible flow solvers based on a finite element discretization*. Univ. Dortmund, 2007. DOI: 10.1504/IJCSM.2007.016531.

[80] B. E. Launder and D. B. Spalding. "The numerical computation of turbulent flows." In: (1983), pp. 96–116. DOI: 10.1016/B978-0-08-030937-8.50016-7.

[81] C. Le, T. Bruns, and D. Tortorelli. "A gradient-based, parameter-free approach to shape optimization." In: *Computer Methods in Applied Mechanics and Engineering* 200.9-12 (2011), pp. 985–996. DOI: 10.1016/j.cma.2010.10.004.

[82] D. J. Lea, M. R. Allen, and T. W. Haine. "Sensitivity analysis of the climate of a chaotic system." In: *Tellus A: Dynamic Meteorology and Oceanography* 52.5 (2000), pp. 523–532. DOI: 10.3402/tellusa.v52i5.12283.

[83] E. N. Lorenz. "Deterministic Nonperiodic Flow." In: *Journal of Atmospheric Sciences* 20.2 (1963), pp. 130 –141. DOI: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.

[84] E. N. Lorenz. "Deterministic nonperiodic flow." In: *Journal of atmospheric sciences* 20.2 (1963), pp. 130–141. DOI: 10.1175/1520-0469(1963)020%3C0130:DNF%3E2.0.CO;2.

[85] A. Loseille and F. Alauzet. “Optimal 3D highly anisotropic mesh adaptation based on the continuous mesh framework.” In: *Proceedings of the 18th International Meshing Roundtable.* Springer, 2009, pp. 575–594. DOI: 10.1007/978-3-642-04319-2_33.

[86] K. Luo, J. Fan, W. Li, and K. Cen. “Transient three-dimensional simulation of particle dispersion in flows around a circular cylinder Re= 140--260.” In: *Fuel* 88.7 (2009), pp. 1294–1301.

[87] Z. Lyu, G. K. Kenway, and J. R. Martins. “RANS-based aerodynamic shape optimization investigations of the common research model wing.” In: *52nd Aerospace Sciences Meeting.* 2014, p. 0567. DOI: 10.2514/6.2014-0567.

[88] M. Mahmood, N. Hayat, M. H. Bashir, A. R. Khan, K. N. Ahmad, and S. Khan. “CFD applications in various heat exchangers design: A review.” In: *Applied Thermal Engineering* 32 (2012), pp. 1 –12. DOI: 10.1016/j.applthermaleng.2011.09.001.

[89] C. M. Mak, J. Niu, C. Lee, and K. Chan. “A numerical simulation of wing walls using computational fluid dynamics.” In: *Energy and Buildings* 39.9 (2007), pp. 995–1002. DOI: 10.1016/j.enbuild.2006.10.012.

[90] V. Mataix Ferrándiz. “Innovative mathematical and numerical models for studying the deformation of shells during industrial forming processes with the Finite Element Method.” In: (2020).

[91] S. Mathew, K. Pandey, and A. Kumar.V. “Analysis of wind regimes for energy estimation.” In: *Renewable Energy* 25.3 (2002), pp. 381–399. DOI: 10.1016/S0960-1481(01)00063-5.

[92] X. Méla and K. Petersen. “Dynamical properties of the Pascal adic transformation.” In: *Ergodic Theory and Dynamical Systems* 25.1 (2005), pp. 227–256. DOI: 10.1017/S0143385704000173.

[93] F. R. Menter. “Review of the shear-stress transport turbulence model experience from an industrial perspective.” In: *International Journal of Computational Fluid Dynamics* 23.4 (2009), pp. 305–316. DOI: 10.1080/10618560902773387.

[94] F. R. Menter. “Two-equation eddy-viscosity turbulence models for engineering applications.” In: *AIAA journal* 32.8 (1994), pp. 1598–1605. DOI: 10.2514/3.12149.

[95] C. Merkle. “Time-accurate unsteady incompressible flow algorithms based on artificial compressibility.” In: *8th Computational Fluid Dynamics Conference*. 1987, p. 1137. DOI: 10.2514/6.1987-1137.

[96] B. Mohammadi and O. Pironneau. *Applied shape optimization for fluids*. Oxford university press, 2010. ISBN: 9780191574214.

[97] D. P. Mok. *Partitionierte Lösungsansätze in der Strukturdynamik und der Fluid-Struktur-Interaktion*. 2001. ISBN: 3-00-007974-2.

[98] A. C. Molina, G Bartoli, and T De Troyer. “Generation of uniform turbulence profiles in the wind tunnel for urban VAWT testing.” In: *Wind Energy Exploitation in Urban Environment*. Springer International Publishing, 2018, pp. 27–43. ISBN: 978-3-319-74944-0.

[99] R. D. Moser, J. Kim, and N. N. Mansour. “Direct numerical simulation of turbulent channel flow up to Re τ= 590.” In: *Physics of Fluids* 11.4 (1999), pp. 943–945. DOI: 10.1063/1.869966.

[100] R. Najian Asl. “Shape optimization and sensitivity analysis of fluids, structures, and their interaction using Vertex Morphing parametrization.” Dissertation. München: Technische Universität München, 2019.

[101] F Nieto, J. Owen, D. Hargreaves, and S Hernández. “Bridge deck flutter derivatives: Efficient numerical evaluation exploiting their interdependence.” In: *Journal of Wind Engineering and Industrial Aerodynamics* 136 (2015), pp. 138–150. DOI: 10.1016/j.jweia.2014.11.006.

[102] C. Norberg. “Fluctuating lift on a circular cylinder: review and new measurements.” In: *Journal of Fluids and Structures* 17.1 (2003), pp. 57–96.

[103] H Oertel Jr. “Wakes behind blunt bodies.” In: *Annual Review of Fluid Mechanics* 22.1 (1990), pp. 539–562.

[104] S. A. Orszag. “Analytical theories of turbulence.” In: *Journal of Fluid Mechanics* 41.2 (1970), 363–386. DOI: 10.1017/S0022112070000642.

[105] C. Othmer. “A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows.” In: *International journal for numerical methods in fluids* 58.8 (2008), pp. 861–877. DOI: 10.1002/fld.1770.

[106] D. Papadimitriou and K. Giannakoglou. “A continuous adjoint method with objective function derivatives based on boundary integrals, for inviscid and viscous flows.” In: *Computers & Fluids* 36.2 (2007), pp. 325–341. DOI: 10.1016/j.compfluid.2005.11.006.

[107] M. Peric. “A finite volume method for the prediction of three-dimensional fluid flow in complex ducts.” PhD thesis. Imperial College London (University of London), 1985.

[108] A. Perry, S Henbest, and M. Chong. “A theoretical and experimental study of wall turbulence.” In: *Journal of Fluid Mechanics* 165 (1986), pp. 163–199. DOI: 10.1017/S002211208600304X.

[109] P. Power, M. Piggott, F Fang, G. Gorman, C. Pain, D. Marshall, A. Goddard, and I. Navon. “Adjoint goal-based error norms for adaptive mesh ocean modelling.” In: *Ocean modelling* 15.1-2 (2006), pp. 3–38. DOI: 10.1016/j.ocemod.2006.05.001.

[110] J. Prospathopoulos, E. Politis, and P. Chaviaropoulos. “Modelling wind turbine wakes in complex terrain.” In: *Proceedings EWEC 2008, Brussels, Belgium* (2008).

[111] O. Reynolds. “IV. On the dynamical theory of incompressible viscous fluids and the determination of the criterion.” In: *Philosophical Transactions of the Royal Society of London. (A.)* 186 (1895), pp. 123–164. DOI: 10.1098/rsta.1895.0004.

[112] A. Ricci, M. Burlando, A. Freda, and M. P. Repetto. “Wind tunnel measurements of the urban boundary layer development over a historical district in Italy.” In: *Building and Environment* 111 (2017), pp. 192–206. DOI: 10.1016/j.buildenv.2016.10.016.

[113] P. Ricco, M. Skote, and M. A. Leschziner. “A review of turbulent skin-friction drag reduction by near-wall transverse forcing.” In: *Progress in Aerospace Sciences* 123 (2021), p. 100713. DOI: 10.1016/j.paerosci.2021.100713.

[114] A. M. Rueda-Ramírez, J. Manzanero, E. Ferrer, G. Rubio, and E. Valero. “A p-multigrid strategy with anisotropic p-adaptation based on truncation errors for high-order discontinuous Galerkin methods.” In: *Journal of Computational Physics* 378 (2019), pp. 209–233. DOI: 10.1016/j.jcp.2018.11.009.

[115] B. Sanderse, S. van der Pijl, and B. Koren. “Review of computational fluid dynamics for wind turbine wake aerodynamics.” In: *Wind Energy* 14.7 (2011), pp. 799–819. DOI: 10.1002/we.458.

[116] M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher. “Benchmark computations of laminar flow around a cylinder.” In: *Flow simulation with high-performance computers II*. Springer, 1996, pp. 547–566. DOI: 10.1007/978-3-322-89849-4_39.

[117] F. Shakib, T. J. Hughes, and Z. Johan. “A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations.” In: *Computer Methods in Applied Mechanics and Engineering* 89.1-3 (1991), pp. 141–219. DOI: 10.1016/0045-7825(91)90041-4.

[118] J. Smagorinsky. “General circulation experiments with the primitive equations: I. The basic experiment.” In: *Monthly Weather Review* 91.3 (Mar. 1963), pp. 99–164. DOI: 10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2.

[119] P. Spalart and S. Allmaras. “A one-equation turbulence model for aerodynamic flows.” In: *30th Aerospace Sciences Meeting and Exhibit*. 1992, p. 439. DOI: 10.2514/6.1992-439.

[120] R Spence and J. Amaral-Teixeira. “A CFD parametric study of geometrical variations on the pressure pulsations and performance characteristics of a centrifugal pump.” In: *Computers & Fluids* 38.6 (2009), pp. 1243–1257. DOI: 10.1016/j.compfluid.2008.11.013.

[121] R. Strehl, A. Sokolov, D. Kuzmin, D. Horstmann, and S. Turek. “A positivity-preserving finite element method for chemotaxis problems in 3D.” In: *Journal of Computational and Applied Mathematics* 239 (2013), pp. 290–303. DOI: 10.1016/j.cam.2012.09.041.

[122] S. H. Strogatz. *Nonlinear dynamics and chaos with student solutions manual: With applications to physics, biology, chemistry, and engineering.* CRC press, 2018. ISBN: 9780429399640.

[123] U. Svensson and K. Häggkvist. “A two-equation turbulence model for canopy flows.” In: *Journal of Wind Engineering and Industrial Aerodynamics* 35 (1990), pp. 201–211. DOI: 10.1016/0167-6105(90)90216-Y.

[124] B. Szabó et al. *Introduction to finite element analysis: formulation, verification and validation.* Vol. 35. John Wiley & Sons, 2011. ISBN: 978-1-119-99348-3.

[125] C. Talnikar, Q. Wang, and G. M. Laskowski. “Unsteady adjoint of pressure loss for a fundamental transonic turbine vane.” In: *Journal of Turbomachinery* 139.3 (2017), p. 031001. DOI: 10.1115/1.4034800.

[126] C. Taylor and P. Hood. “A numerical solution of the Navier-Stokes equations using the finite element technique.” In: *Computers & Fluids* 1.1 (1973), pp. 73–100. DOI: 10.1016/0045-7930(73)90027-3.

[127] The HDF Group. *Hierarchical data format version 5.* 2000-2010. DOI: 10.11578/dc.20180330.1. URL: http://www.hdfgroup.org/HDF5.

[128] P. Tucker, S. Menon, C. Merkle, J. Oefelein, and V. Yang. “Validation of high-fidelity CFD simulations for rocket injector design.” In: *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit.* 2008, p. 5226. DOI: 10.2514/6.2008-5226.

[129] S. Turek, D. Kuzmin, and R. Löhner. *Flux-corrected Transport: Principles, Algorithms, and Applications (Scientific Computation).* Springer Netherlands, 2012. ISBN: 978-94-007-4038-9.

[130] K. Uehara, S. Murakami, S. Oikawa, and S. Wakamatsu. "Wind tunnel experiments on how thermal stratification affects flow in and above urban street canyons." In: *Atmospheric Environment* 34.10 (2000), pp. 1553–1562. DOI: 10.1016/S1352-2310(99)00410-0.

[131] J. G. V. Vázquez. "Nonlinear Analysis of Orthotropic Membrane and Shell Structures Including Fluid-Structure Interaction." PhD thesis. Universitat Politècnica de Catalunya (UPC), 2007.

[132] D. A. Venditti and D. L. Darmofal. "Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows." In: *Journal of Computational Physics* 187.1 (2003), pp. 22–46. DOI: 10.1016/S0021-9991(03)00074-3.

[133] R. Verfürth. "A posteriori error estimation and adaptive mesh-refinement techniques." In: *Journal of Computational and Applied Mathematics* 50.1-3 (1994), pp. 67–83. DOI: 10.1016/0377-0427(94)90290-9.

[134] H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method.* Pearson Education, 2007. ISBN: 978-0131274983.

[135] T. Von Kármán. "Mechanical Similarity and Turbulence." In: *News from the Society of Sciences in Göttingen, Mathematical-Physical Class* 1930 (1930), pp. 58–76.

[136] J. G. Wallwork, N. Barral, D. A. Ham, and M. D. Piggott. "Anisotropic goal-oriented mesh adaptation in Firedrake." In: *28th Intl Meshing Roundtable* (2019).

[137] J. Wang and G. Liu. "A point interpolation meshless method based on radial basis functions." In: *International Journal for Numerical Methods in Engineering* 54.11 (2002), pp. 1623–1648. DOI: 10.1002/nme.489.

[138] Q. Wang. "Convergence of the least squares shadowing method for computing derivative of ergodic averages." In: *SIAM Journal on Numerical Analysis* 52.1 (2014), pp. 156–170. DOI: 10.1137/130917065.

[139] Z. Wang, J. Fan, and K. Cen. “Immersed boundary method for the simulation of 2D viscous flow based on vorticity--velocity formulations.” In: *Journal of Computational Physics* 228.5 (2009), pp. 1504–1520.

[140] S. Warnakulasuriya, M. Andre, R. Wüchner, and K.-U. Bletzinger. “Stabilization of a Time-Dependent Discrete Adjoint Solver for Chaotic Incompressible Flows.” In: *PAMM* 18.1 (2018), e201800124. DOI: 10.1002/pamm.201800124.

[141] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. “A tensorial approach to computational continuum mechanics using object-oriented techniques.” In: *Computers in Physics* 12.6 (1998), pp. 620–631. DOI: 10.1063/1.168744.

[142] D. C. Wilcox. “Formulation of the kw turbulence model revisited.” In: *AIAA Journal* 46.11 (2008), pp. 2823–2838. DOI: 10.2514/1.36541.

[143] D. C. Wilcox. “Reassessment of the scale-determining equation for advanced turbulence models.” In: *AIAA Journal* 26.11 (1988), pp. 1299–1310. DOI: 10.2514/3.10041.

[144] D. C. Wilcox. “Turbulence modeling.” In: *DCW Industries* (1993).

[145] A. Winterstein, M. Péntek, K.-U. Bletzinger, and R. Wüchner. “Fluid-structure-control interaction (FSCI): General algorithm and its application to the reduction of wind-induced vibrations of civil engineering structures.” In: *VIII International Conference on Coupled Problems in Science and Engineering*. 2019.

[146] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano. “Determining Lyapunov exponents from a time series.” In: *Physica D: Nonlinear Phenomena* 16.3 (1985), pp. 285–317. DOI: 10.1016/0167-2789(85)90011-9.

[147] W. Wood, M Bossak, and O. Zienkiewicz. “An alpha modification of Newmark’s method.” In: *International Journal for Numerical Methods in Engineering* 15.10 (1980), pp. 1562–1566. DOI: 10.1002/nme.1620151011.

[148] Y. Yu, Z. Lyu, Z. Xu, and J. R. Martins. “On the influence of optimization algorithm and initial design on wing aerodynamic shape optimization.” In: *Aerospace Science and Technology* 75 (2018), pp. 183 –199. DOI: 10.1016/j.ast.2018.01.016.

[149] X. Zhang, X.-H. Liu, K.-Z. Song, and M.-W. Lu. “Least-squares collocation meshless method.” In: *International Journal for Numerical Methods in Engineering* 51.9 (2001), pp. 1089–1100. DOI: 10.1002/nme.200.

[150] O. C. Zienkiewicz and J. Z. Zhu. “A simple error estimator and adaptive procedure for practical engineering analysis.” In: *International journal for numerical methods in engineering* 24.2 (1987), pp. 337–357. DOI: 10.1002/nme.1620240206.

[151] M. R. Zorrilla. “Towards the virtual wind tunnel for civil engineering applications.” In: (2020).

[152] A. Zymaris, D. Papadimitriou, K. Giannakoglou, and C Othmer. “Continuous adjoint approach to the Spalart--Allmaras turbulence model for incompressible flows.” In: *Computers & Fluids* 38.8 (2009), pp. 1528–1538. DOI: 10.1016/j.compfluid.2008.12.006.

Bisherige Titel der Schriftenreihe

Band	Titel
1	Frank Koschnick, *Geometrische Lockingeffekte bei Finiten Elementen und ein allgemeines Konzept zu ihrer Vermeidung*, 2004.
2	Natalia Camprubi, *Design and Analysis in Shape Optimization of Shells*, 2004.
3	Bernhard Thomee, *Physikalisch nichtlineare Berechnung von Stahlfaserbetonkonstruktionen*, 2005.
4	Fernaß Daoud, *Formoptimierung von Freiformschalen - Mathematische Algorithmen und Filtertechniken*, 2005.
5	Manfred Bischoff, *Models and Finite Elements for Thin-walled Structures*, 2005.
6	Alexander Hörmann, *Ermittlung optimierter Stabwerkmodelle auf Basis des Kraftflusses als Anwendung plattformunabhängiger Prozesskopplung*, 2006.
7	Roland Wüchner, *Mechanik und Numerik der Formfindung und Fluid-Struktur-Interaktion von Membrantragwerken*, 2006.
8	Florian Jurecka, *Robust Design Optimization Based on Metamodeling Techniques*, 2007.
9	Johannes Linhard, *Numerisch-mechanische Betrachtung des Entwurfsprozesses von Membrantragwerken*, 2009.
10	Alexander Kupzok, *Modeling the Interaction of Wind and Membrane Structures by Numerical Simulation*, 2009.
11	Bin Yang, *Modified Particle Swarm Optimizers and their Application to Robust Design and Structural Optimization*, 2009.
12	Michael Fleischer, *Absicherung der virtuellen Prozesskette für Folgeoperationen in der Umformtechnik*, 2009.
13	Amphon Jrusjrungkiat, *Nonlinear Analysis of Pneumatic Membranes - From Subgrid to Interface*, 2009.
14	Alexander Michalski, *Simulation leichter Flächentragwerke in einer numerisch generierten atmosphärischen Grenzschicht*, 2010.
15	Matthias Firl, *Optimal Shape Design of Shell Structures*, 2010.
16	Thomas Gallinger, *Effiziente Algorithmen zur partitionierten Lösung stark gekoppelter Probleme der Fluid-Struktur-Wechselwirkung*, 2011.
17	Josef Kiendl, *Isogeometric Analysis and Shape Optimal Design of Shell Structures*, 2011.
18	Joseph Jordan, *Effiziente Simulation großer Mauerwerksstrukturen mit diskreten Rissmodellen*, 2011.

Band	Titel
19	Albrecht von Boetticher, *Flexible Hangmurenbarrieren: Eine numerische Modellierung des Tragwerks, der Hangmure und der Fluid-Struktur-Interaktion*, 2012.
20	Robert Schmidt, *Trimming, Mapping, and Optimization in Isogeometric Analysis of Shell Structures*, 2013.
21	Michael Fischer, *Finite Element Based Simulation, Design and Control of Piezoelectric and Lightweight Smart Structures*, 2013.
22	Falko Hartmut Dieringer, *Numerical Methods for the Design and Analysis for Tensile Structures*, 2014.
23	Rupert Fisch, *Code Verification of Partitioned FSI Environments for Lightweight Structures*, 2014.
24	Stefan Sicklinger, *Stabilized Co-Simulation of Coupled Problems Including Fields and Signals*, 2014.
25	Madjid Hojjat, *Node-based parametrization for shape optimal design*, 2015.
26	Ute Israel, *Optimierung in der Fluid-Struktur-Interaktion - Sensitivitätsanalyse für die Formoptimierung auf Grundlage des partitionierten Verfahrens*, 2015.
27	Electra Stavropoulou, *Sensitivity analysis and regularization for shape optimization of coupled problems*, 2015.
28	Daniel Markus, *Numerical and Experimental Modeling for Shape Optimization of Offshore Structures*, 2015.
29	Pablo Suárez, *Design Process for the Shape Optimization of Pressurized Bulkheads as Components of Aircraft Structures*, 2015.
30	Armin Widhammer, *Variation of Reference Strategy - Generation of Optimized Cutting Patterns for Textile Fabrics*, 2015.
31	Helmut Masching, *Parameter Free Optimization of Shape Adaptive Shell Structures*, 2016.
32	Hao Zhang, *A General Approach for Solving Inverse Problems in Geophysical Systems by Applying Finite Element Method and Metamodel Techniques*, 2016.
33	Tianyang Wang, *Development of Co-Simulation Environment and Mapping Algorithms*, 2016.
34	Michael Breitenberger, *CAD-integrated Design and Analysis of Shell Structures*, 2016.
35	Önay Can, *Functional Adaptation with Hyperkinematics using Natural Element Method: Application for Articular Cartilage*, 2016.
36	Benedikt Philipp, *Methodological Treatment of Non-linear Structural Behavior in the Design, Analysis and Verification of Lightweight Structures*, 2017.
37	Michael Sean Andre, *Aeroelastic Modeling and Simulation for the Assessment of Wind Effects on a Parabolic Trough Solar Collector*, 2018.

Band	Titel
38	Andreas Apostolatos, *Isogeometric Analysis of Thin-Walled Structures on Multipatch Surfaces in Fluid-Structure Interaction*, 2018.
39	Altuğ Emiroğlu, *Multiphysics Simulation and CAD Integrated Shape Optimization in Fluid-Structure Interaction*, 2019.
40	Mehran Saeedi, *Multi-Fidelity Aeroelastic Analysis of Flexible Membrane Wind Turbine Blades*, 2017.
41	Reza Najian Asl, *Shape optimization and sensitivity analysis of fluids, structures, and their interaction using Vertex Morphing parametrization*, 2019.
42	Ahmed Abodonya, *Verification Methodology for Computational Wind Engineering Prediction of Wind Loads on Structures*, 2020.
43	Anna Maria Bauer, *CAD-integrated Isogeometric Analysis and Design of Lightweight Structures*, 2020.
44	Andreas Winterstein, *Modeling and Simulation of Wind-Structure Interaction of Slender Civil Engineering Structures Including Vibration Mitigation Systems*, 2020.
45	Franz-Josef Ertl, *Vertex Morphing for Constrained Shape Optimization of Three-dimensional Solid Structures*, 2020.
46	Daniel Baumgärtner, *On the Grid-based Shape Optimization of Structures with Internal Flow and the Feedback of Shape Changes into a CAD Model*, 2020.
47	Mohamed Khalil, *Combining Physics-based models and machine learning for an Enhanced Structural Health Monitoring*, 2021.
48	Long Chen, *Gradient Descent Akin Method*, 2021.
49	Aditya Ghantasala, *Coupling Procedures for Fluid-Fluid and Fluid-Structure Interaction Problems Based on Domain Decomposition Methods*, 2021.
50	Ann-Kathrin Goldbach, *The Cad-Integrated Design Cycle for Structural Membranes*, 2021.
51	Iñigo Pablo López Canalejo, *A Finite-Element Transonic Potential Flow Solver with an Embedded Wake Approach for Aircraft Conceptual Design*, 2022.
52	Mayu Sakuma, *An Application of Multi-Fidelity Uncertainty Quantification for Computational Wind Engineering*, 2022.