Combining Shaping and Flow Control for Performance Optimization

Mai Zhang

Mansfield College

Department of Engineering Science
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Hilary 2016

Acknowledgements

Firstly, I would like to express my foremost gratitude to my supervisor Professor Li He for providing me the precious opportunity to take on this interesting project. I'm very grateful for his patient and dedicated support and advice, generous time for regular meetings, spiritual cheer-up and advice beyond academic issues, during the course of the study. The things learnt from his advice will stay with me forever.

I would also like to express many thanks to Professor Richard Willden and Professor David Gillespie for their dedication of time and support for my transfer viva and confirmation of status.

Much gratitude is also given to Dr. Liwei Chen, Dr. Junsok Yi, Dr. Yuewen Jiang, Dr. Brian Tang for their help in discussion of theory and ideas, sharing their experience in their own field, guidance on programming, etc. I would also like to thank Mr Craig Parsons very much for his kind-hearted and effective IT support.

I would also like to thank my fellow DPhil colleagues in the lab to create an accommodating and professional working environment, and their discussion of problems in the seminar.

Much gratitude also goes to China Scholar Ship council who provides financial support for my study.

Lastly, I'm extremely grateful to my mother and father, especially my mother, for their constant spiritual support from the other side of the planet during all these years of endeavour.

Abstract

Conventionally, shaping and flow control are treated as two separate disciplines. Both have been extensively studied and demonstrated to lead to very effective performance improvement for aerodynamic surfaces in the past. Despite of this, there are very few works that have combined the two together, a stark contrast to the huge number of publications within each discipline respectively. This thesis presents a first-of-the-kind systematic investigation into the combination of the two disciplines for performance improvement in a general sense. To provide efficient high-fidelity flow modelling as required by the flow control analysis, a novel method to efficiently utilize the large-eddy simulation (LES) to analyse the complicated flow is developed, successfully making the LES much more affordable to be used in the design optimizations, while maintaining the fidelity and accuracy of the LES solutions.

A computational study is first carried out with 2D geometries and fast RANS performance evaluation as a starting point for the subsequent 3D investigation with the high-fidelity flow modelling. Five optimization approaches are systematically investigated, which are 1) shape optimization alone; 2) flow control optimization alone; 3) optimizing the shape then optimizing the flow control (sequential I); 4) optimizing the flow control then optimizing the shape (sequential II); 5) simultaneously optimizing the shape and the flow control (concurrent). The results show the combined optimizations of the two disciplines produce higher performance than the maximized performances from the conventional shape optimization alone or flow control optimization alone. In the combined optimizations, the concurrent approach produces higher performance than the two sequential approaches.

Moving the investigation onto 3D realistic configurations, the novel efficient LES method is developed to provide efficient LES-based numerical solution of the complicated

flow control flow field. As only the time-mean performance of the unsteady flow field is of interest, and as the time-mean flow field is time-dependent, an attempt to improve the computational efficiency has been proposed, which is obtain the time-mean LES solution by solving the steady-state equations instead of solving the unsteady LES governing equations. In doing so, the key is to obtain the correct turbulence correlation source terms (e.g. Reynolds stress gradients) for the steady-state equations for each design. A few LES runs are still required at a few initial sample designs to compute the initial set of correct source terms that would enable the steady-state simulations to reproduce the time-mean LES solutions at these sample designs. The methods to iteratively extract the source terms from the sample LES solutions and produce the converged steady-state solutions are developed. Then, various source term propagation methods are developed to adapt the sample source terms for the remaining designs in the design space, so that they can be evaluated with the fast steady-state simulations instead of the expensive time-accurate LES, whilst the steady-state solutions will still have a as high-fidelity and accuracy as the LES. The total computational cost thus can be greatly reduced comparing to the brute-force LES for all the designs.

The developed efficient LES method is successfully implemented in the combined shape and flow control optimization of a 3D realistic blade trailing edge cutback injection configuration. The steady-state performance evaluations fit well with into the LES-predicted trend of variations, illustrating its ability of achieving a significant reduction of computational costs compared to the brute-force LES.

Table of Content

List of Figures

List of Tables

Nomenclature

Arithmetic Symbols

$-$	Time-averaging / filtering
$\sim$	Favre-averaging / Favre-filtering
$\langle\cdot\rangle$	Ensemble averaging / time-spanwise-averaging in quasi 3D geometry
$\lvert\cdot\rvert$	Modular
$\lVert\cdot\rVert$	L^2 -norm
$(\cdot\ ,\ \cdot)$	Inner product for continuous function
$\sum$	Summation
$\prod$	Production

Font Style (except caption, superscript, subscript)

Italic	Scalar variable
Bold Italic	Vector / matrix, vector / matrix of a collection of scalar variables

Latin Symbols

a	POD coefficient
A	Area
b_c, b_k	Polynomial coefficient for the c-th cell or k-th POD coefficient
c	Chord length of blade and airfoil
c_{trim}	Chord length of trimmed airfoil
C_D	Drag coefficient
C_L	Lift coefficient
C_μ	Momentum coefficient for flow control injection
d	Design variable
D	Diameter
D^m	Solution change divided by time/pseudo-time for the m-th source term iteration
$Drag$	Drag force
e	Internal energy
E	Total energy
f	Frequency

$f(\cdot)$	Regression basis function
$\boldsymbol{F}$	Design matrix
$\boldsymbol{F}_i$	Convective flux vector in i-th direction
$\boldsymbol{F}_{v,i}$	Viscous flux vector in i-th direction
$g(\cdot)$	Constraint function for the optimization
$G(\cdot)$	Kernel function for filtering
h	Height of blowing slot \ enthalpy
H	Total enthalpy
$\boldsymbol{I}$	Identity matrix
$I(\cdot)$	Objective function for the optimization
k	(Turbulent) kinetic energy
L	Lift force
$\dot{m}$	Mass flow rate
Ma	Mach number
N	Degree of polynomial
N_{cell}	Number of mesh cell in the computational domain
N_S	Number of samples
N_V	Number of design variables
p	Static pressure
p_t	Total pressure
Pr	Prandtl number
q	Heat flux
$\boldsymbol{Q}$	Vector of non-conservation form of solution variables / matrix of eigenvectors
$\boldsymbol{r}(\cdot)$	Correlation between new and sample design sites \ spatial displacement distance
Re	Reynolds number
R	Gas constant
$\boldsymbol{R}$	Correlation matrix
R_{RMS}	Root-Mean-Square flow solution change of an source term iteration
$R(\cdot)$	Residual
s	Blade cascade pitch length / discrete source term value
$\boldsymbol{s}$	Vector of discrete source term values
S	Source term

S_{ij}	Strain-rate tensor
St	Strouhal number
t	time
u_i	Component of velocity in the i-th spatial direction
U	Magnitude of reference velocity
$\boldsymbol{U}$	Vector of conservation form of solution variables
$\boldsymbol{U}_1$	Matrix of POD bases
w	Weight in objective function
w_{jet}	Energy cost of flow control per unit mass flow rate
x, y, z	x/streamwise coordinate, y coordinate, z/spanwise coordinate
Z	Length of span of the geometry

Greek Symbols

α	Under-relaxation factor / stage coefficient of the Runge-Kutta scheme
β	Inlet flow angle / regression coefficient
γ	Blade cascade stagger angle
$\boldsymbol{\Gamma}$	Precondition matrix in density-based solver
δ	Direction normal to blade chord / Dirac delta
Δ	Change or difference / length scale
$\zeta, \zeta_{th}, \zeta_{eq}$	Aerodynamic loss, thermodynamic loss, equivalent loss
η	Efficiency
η_{ad}	Adiabatic effectiveness
θ	Angle (e.g. blowing angle) / A parameter in the Gaussian correlation
κ	Thermal conductivity
λ	Lagrange multiplier / Eigenvalue
$\boldsymbol{\Lambda}$	Matrix of eigenvalues
μ	Molecular viscosity
μ_d	Viscosity to provide additional damping
ν	Kinematic viscosity
ξ	Chordwise direction
ρ	Density
σ	Singular value

$\boldsymbol{\Sigma}$	Matrix of singular values
τ	Pseudo time
ϕ	General flow variable
$\boldsymbol{\varphi}$	Eigenvector / discrete version of POD basis
Φ	Continuous version of POD basis
ω	vorticity

Superscripts

1	Inlet condition
2	Outlet condition
T	Transpose
$*$	Desired value
m	m-th instance / time-step / pseudo-time-step
0	Initial instance

Subscript

ad	Adiabatic
brk	Breakout surface
c	Coolant / cell index
d	Design condition / damping / design space
f	Face index of a mesh cell
FC	Flow control
h	Hot main flow
i, j, k	Spatial index
jet	Blowing / suction jet
$land$	Land
low	Lower bound
max, min	Maximum, minimum value
$norm$	Normalized value
up	Upper bound
p	p-th polynomial term
r	Reference
t	Total quantity (for pressure, etc.)

te	Trailing edge
$trim$	Trimmed (chord)
$shape$	Shape
v	Viscous
w	Wall
x, y, z, e, m	x, y, z momentum equations, energy equation, continuity equation

Abbreviation

2D, 3D	Two-dimensional, three-dimensional
BCD	Bounded central difference (scheme)
CD	Central difference (scheme)
AOA	Angle of attack
CFD	Computational fluid dynamics
CTC	Cell-to-cell (propagation approach)
DNS	Direct numerical simulation
FC	Flow control
FV	Finite volume (discretization)
Krg	Kriging
LES	Large-eddy simulation
NACA	National advisory committee for aeronautics
NASA	National aeronautics and space administration
NS	Navier-Stokes
NITA	Non-iterative time advancement
POD	Proper-orthogonal decomposition
Quad	Quadratic
RANS	Reynolds-averaged Navier-Stokes
ROM	Reduced-order model
SOU	Second-order upwind (scheme)
URANS	Unsteady Reynolds-averaged Navier-Stokes

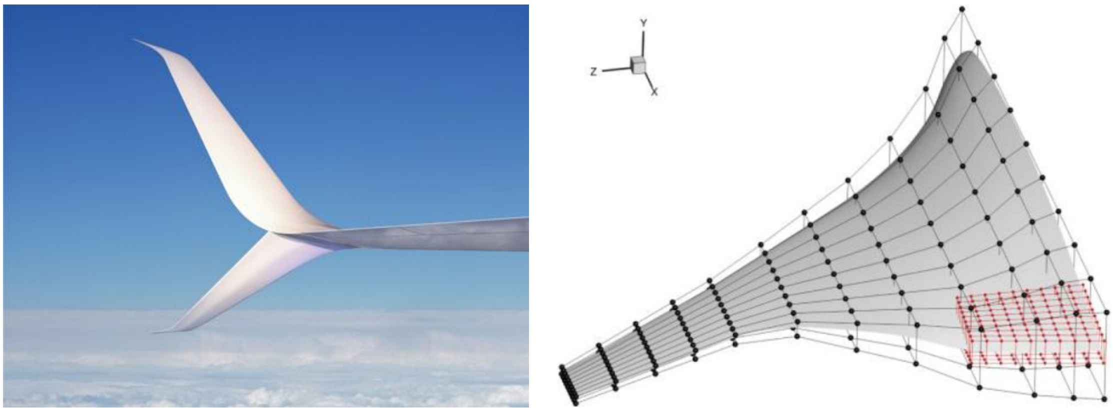
Chapter 1

Introduction

The performance of an aircraft is largely dependent on the performance of its lifting surfaces. Most notably, the wings play a fundamental role in an aircraft's performance. High performance wings lead to high manoeuvrability, short take-off and landing distance, high payload capacity, long cruise distance, etc. In the engines, the performance of the blades is critical to the performance of the engine. High performance engine can produce high thrust to weight ratio, low specific fuel consumption, low noise emission, etc., which also leads to high manoeuvrability, short take-off and landing distance, long cruise distance, high market competitiveness, etc.

1.1 Shape Optimization

The performance of the aerodynamic devices is fundamentally determined by their shapes. Therefore, the shape optimization is a critical discipline to maintain the high performance and has been studied extensively (e.g. Painchaud-Ouellet et al., 2006; Zing and Elias, 2006; Epstein et al., 2005; Rallabhandi and Mavris, 2007; etc.). Over the years, various shape optimization research efforts have produced huge impact for improving the performance. For example, the supercritical airfoil reduces the wave drag and the design becomes common for modern passenger aircrafts. Carefully optimized winglets help airliners to reduce drag and fuel consumption and also noise emission. In turbomachinery, the design of the controlled diffusion blading profile increases the efficiency of the compressor blades, due to the elimination of the passage shock and shock-induced boundary layer separation when the blades are operated at high speed conditions. Some practical examples of shaping are shown in Figure 1-1.

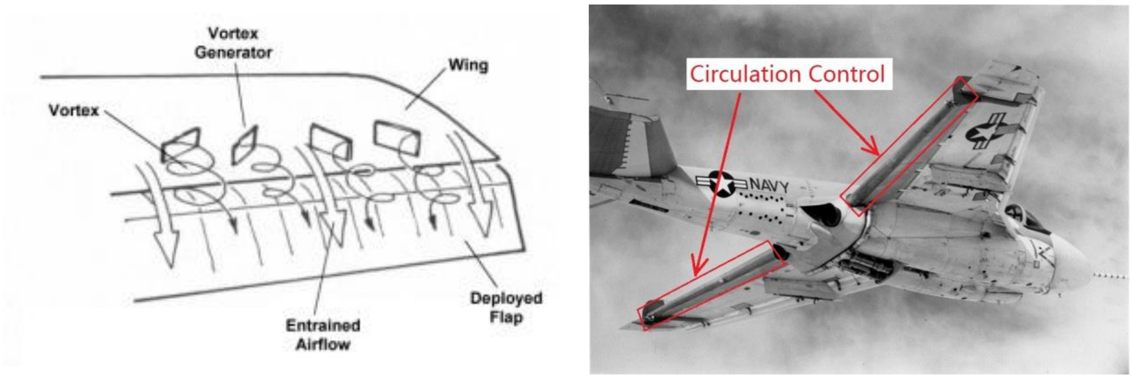


(a) **(b)**

Figure 1-1 Examples of shape optimization: (a) optimized winglet shape; (b) shaping the half blended-wing-body aircraft with control points (Lyu and Martins, 2014)

1.2 Flow Control

Aside from shaping, flow control is another important discipline for performance improvement and has also been extensively developed, particularly in the past two decades (see Gad-el-Hak, 2001). In the aeronautical applications, flow control is usually applied as local fluidic forcing on the aerodynamic surface to achieve desired changes in the flow field to improve the performance. Typical flow control examples include~~s~~ vortex generators, boundary layer blowing and suction, trailing edge blowing, circulation control, plasma actuators, etc. Over the years of research, flow controls have resulted in significant performance enhancement, such as in separation control, noise reduction, flutter damping, lift increment, drag reduction, etc. Some examples of practical applications of flow control are shown in Figure 1-2.



(a) **(b)**

Figure 1-2 Flow control examples: (a) vortex generators; (b) circulation control on the wing trailing edge of the A-6 aircraft (Englar et al., 1979)

1.3 Combining Shaping and Flow Control Optimization

Although both shaping and flow control have been proven to be very effective in improving the performance of lift surfaces for external and internal flows (wing and blades), the effort in combining the shape optimization with flow control optimization is still very rare. Compared with the very large number of papers published in each discipline respectively, there are only few papers so far that have touched both disciplines at the same time (e.g. Pehlivanoglu and Yagiz, 2011; Nielsen and Jones, 2011) . Though these works have indicated the possibility of combining the shaping with flow control for performance improvement, the geometry and the flow control configurations are all simplified two-dimensional ones. It seems to be difficult to find conclusive findings in open literature regarding the relation between shaping and flow control, and possible benefits when they are combined.

A major challenge to combine the shaping with flow control is the conflict between the computational efficiency of solutions as required for the shape optimization and the high fidelity but time-consuming flow modelling/solution methods as required by the flow control.

In shaping, the optimization efficiency is the major concern, which leads to the majority of past developments focusing on the shape parameterization, optimization algorithms, formulation of the optimization problems (e.g. single versus multiple operating points), etc. On the flow simulation side, however, as the ‘optimised’ flows are often relatively simple and well behaved, they can be relatively easily modelled with adequate accuracy by the matured RANS models. The steady RANS methods are usually chosen in shaping due to the low computational cost in performance evaluation, given a large number of possible configurations to be evaluated in a typical optimization problem.

In flow control, however, due to the complexities of the flow field involved, the required numerical flow modelling is a bigger challenge. Thus, high-fidelity unsteady simulations, such as LES are often required, whose computational cost tend to be three to four orders of magnitude higher than the RANS used for shape optimization. The reasons are, first, the baseline flow fields that need to be controlled are usually complicated, such as those with flow separation in which large scale unsteadiness and a wide range of turbulence scales are present. Moreover, the applications of the flow control may bring small-scale, unconventional, three-dimensional, or unsteady flow structures to the flow field (e.g. in local flow ejection). These flow conditions tend to be beyond the capability of the conventional RANS turbulence models.

1.4 Present Research Objectives

The present work is motivated by the well-established effectiveness on the performance improvement by shaping and flow control respectively on the one hand, but clear lack of previous efforts in combining the two on the other hand. There are two principal issues of interest: (i) is there any positive benefit of combining the shaping and flow control? and (ii) if so, what would be a required flow modelling fidelity, and what can we do to advance the modelling techniques to towards providing such a modelling to enable the combined shaping and flow control optimization accurately and efficiently?

Given these basic considerations, the first objective of the present work is to explore whether there are practically meaningful scenarios in which combining shaping with flow control optimization can produce further performance improvement. The influence of different sequencing of the two disciplines in the combined optimization, i.e. the sequential optimizations versus the concurrent optimization will also be studied.

The second objective is to resolve the conflict between the computational speed/efficiency and the modelling fidelity required by the performance evaluation of flow control, when combining the two disciplines in 3D realistic optimization cases. As many of the flow control solutions point to the need for LES, a new method to increase the efficiency of LES analysis in the combined optimization is required to make the overall computational cost affordable.

1.5 Thesis Outline

Chapter 1, as presented describes the background, identifies the research problems and presents the objectives for this thesis.

Chapter 2 reviews the state of the art of shaping and flow control, with an emphasis on their effectiveness in improving the performance, as well as the main limitations in the two disciplines. The basic optimization methods are presented to serve this purpose. Various previous works in developing efficient methods for unsteady flow analysis are discussed as the baseline reference point for the development of the new method for efficient LES solutions for design optimization.

Chapter 3 presents the basic methods for the optimization and numerical flow simulations. Different optimization approaches for the combined optimizations are identified to investigate the effects of combining the shaping and the flow control optimization, as well as to investigate the different ways of the combination. The optimization systems compatible with all the optimization approaches are developed. Then, the flow solver, and the baseline RANS and LES solution methods are described.

Chapter 4 presents a series of 2D case studies of combining shaping and flow control optimizations, using the low fidelity but fast RANS simulations. This chapter aims to systematically examine the effect of combining the two disciplines and the different ways to combine them. The outcomes of these case studies further justify the substantial investigation on 3D realistic configurations, and consequently the requirement to develop the efficient LES method.

Chapter 5 validates the solver’s LES capability on some canonical flow cases. The importance of using LES to simulate the complex flows is demonstrated on the realistic configurations.

Chapter 6 outline the overall strategy of the efficient LES method, and presents the Part I development of the efficient LES method, which is to compute the source terms to reproduce the time-mean LES solutions with steady-state solution.

Chapter 7 discusses the Part II development the efficient LES method, focusing on the propagation of the source terms for the steady-state solution at the new designs to return the solution accuracy comparable to the time-mean LES solution, without requiring anymore LES evaluations on the new designs.

Chapter 8 presents the procedure to integrate the efficient LES method into the optimization. The demonstrative optimization case combines the shaping and flow control optimization of a 3D realistic configuration.

Chapter 9 presents the concluding remarks for the thesis and gives some general suggestions for possible future works.

Chapter 2

Literature Review

This chapter will review the present state of research in shaping and flow control, with the aim to show the typical applications, effectiveness and main aspects concerned in the two disciplines to improve the performance. Also, previous efforts that have touched the combination of the two disciplines will be discussed. A number of previous research efforts relevant to the method development to potentially tackle the flow modelling challenges arising from combining shaping and flow control in performance optimization will also be reviewed.

2.1 Shaping

As the shape fundamentally determines the performance of an aerodynamic device, shaping has featured in a large amount of studies so far. The basic assumption inherent in all the shape optimization exercises is that the flow solvers adopted, as being imperfect and restricted by turbulence models as they can be, are able to rank the different designs (i.e. tell the differences in performance) consistently. Thus, most studies in shaping were concerned more about the efficiency of the optimization, such as the shape parameterization, optimization methods, formulation of the optimization problems, the combination with other disciplines, etc. A comprehensive review of the shape optimization is not intended here. This section will only review some examples to give an overall picture of how shaping has been used to improve the performance.

2.1.1 Examples of Shaping

In external aerodynamic problems, shape optimization studies range from two-dimensional airfoils to three-dimensional wings and from individual components to an entire aircraft. For example, Vavalle and Qin (2007) optimized the shape of a two dimensional RAE2822 airfoil to minimize the wave drag at a single point operating condition. After the shaping, the shock wave on the airfoil was greatly smeared and the drag was reduced by 32%. The work by Epstein and Peigin (2005) used a genetic algorithm to optimize a three dimensional wing to reduce the drag at multiple operating points. A study by Simpson et al. (2001) optimized the shape of a three-dimensional aerospike nozzle using the Kriging method for thrust maximization, weight minimization and etc. A study by Lyu and Martins (2014) optimized the shape of a whole blended-wing-body aircraft and was able to reduce the drag by around 25%.

For internal aerodynamic problems, shape optimization has been applied to designing the blades of turbomachinery. Similar to the cases in external aerodynamics, a large number of papers have been published and the present research is only able to list a few. For example, in the work of Burguburu et al. (2004), the blade in a three-dimensional transonic compressor blade row is optimized, leading to a 1.07% increase in the isentropic efficiency of the blade row. The increase in the efficiency is largely due to the reduction of passage shockwave intensity due to shape change. In another example, Luo et al. (2011) optimized the three-dimensional geometry of a low pressure turbine blade in a cascade and managed to increase the pressure ratio and reduce the efficiency over a range of mass flow rate. The adjoint method of Jameson (1988) was implemented to efficiently compute the gradient of the objective function.

Aside from pure aerodynamics, shaping has also been coupled with multi-disciplinary analysis to improve the performance besides the aerodynamic ones. In relation to blade aeroelasticity, a study by Wang and He (2011) coupled the fluid and structural analysis in the shape optimization to concurrently optimize both the aerodynamic and aeroelastic performance of a three-dimensional compressor blade. They used the nonlinear harmonic phase solution method (He, 2008) to efficiently analyse the unsteady flows and the adjoint method for efficient computational of the gradient of the objective function in the presence of a large number of design variables.

It can be said in general that shape optimization is being applied routinely in designing aerodynamic surfaces, and there should be no doubt that it is a main approach for performance improvement.

2.1.2 Shape Parameterization

The shape parameterization method is an important aspect in shape optimization, so that all the candidate shapes can be represented by a few design variables. In shape deformation for the lifting surfaces, it is required that the shape should remain continuous and smooth, which is also conducive to a smooth objective function. Moreover, an efficient shape parameterization method will enable the designers to seek greater flexibility to explore a wider range of shapes, or represent the desired shape more precisely, while using as fewer design variables as possible to reduce the computational cost. Whereas each optimization case may have its own best suited shape parameterization method, there are some shape parameterization methods that have been used very widely and quite effective in shaping the lifting surfaces. This section will review some of them.

To achieve flexibility, a widely used shape parameterization method is the Hick-Henne bump function, proposed by Hick and Henne (1978). It creates the new shape by adding a number of Hick-Henne bump functions to the vertical coordinate of the baseline shape:

$$y(x) = y_{baseline}(x) + \sum_{i=1} \alpha_i f_i(x) \tag{2.1}$$

$$f_i(x) = \left[\sin\left(\pi x^{\frac{\log 0.5}{\log b_1}}\right)\right]^{b_2}, 0 \le x \le 1 \tag{2.2}$$

where the bump function coefficients α_i are usually taken as the design variables. The controlling parameters b_1 and b_2 determine the peek location and the width of a bump. Thus it can be seen that it offers the flexibility in choosing where and how local the shape deformation can be. This has made it a popular choice in many shaping applications for airfoils (e.g. Hicks et al., 1974, Lee and Eyi 1993), wings (e.g. Hick and Henne, 1978; Reuther et al., 1999), and for turbomachinery blades (e.g. Wu et al., 2003; Wang et al., 2010; Walther and Nadarajah, 2015).

Another widely used method for complex shape parameterization is based on the B-spline curve family, such as the Non-Uniform Rational B-Spline (NURBS) (see Rogers, 2000). The NURBS offers great flexibility and accuracy in geometry representation, from smooth low-curvature features to noisy, spiky and sharp features. It also offers an intuitive method to change the shape by moving the control points while still maintaining the smoothness of the shape. Besides aerodynamics, it is also the main shape representation method the computer aided industrial applications, such as computer aided design (CAD), manufacturing (CAM). According to (Castonguay and Nadarajah, 2007), compared with Hicks-Henne bump function method, the NURBS can represent the same complexity of geometry with fewer design variables and better accuracy. Another advantage is that as NURBS is widely used in other applications such as CAD, it makes it more natural to integrate the CAD geometric modelling to the design optimization (Castonguay and Nadarajah, 2007). Many researchers have adopted NURBS in shaping, such as those by Painchaud-Ouellet et al. (2006) for the airfoils, Lepine et al. (2001) for the wings and Rooij et al. (2005) for the blades.

Indeed, there are numerous other shape parameterization methods and the designer should choose the suitable one based on the problems at hand. All the shape parameterization methods reviewed above are able to preserve continuity and smoothness, and achieve good flexibility without using too many design variables, and are the common choices for shape optimization.

2.2 Flow Control

As another approach to improve performance, flow control has also been studied extensively for decades. Unlike shaping, most flow control research efforts are focused on understanding the underlying fluid mechanics, as the flow field in the flow control is usually complicated and non-optimum (more off-design conditions than design ones). Many of the researchers use experiments, while some use high-fidelity numerical simulations such as LES.

Flow control can be classified as passive or active. Passive flow control does not require extra energy expenditure. It is usually applied as local minor geometry modifications on the aerodynamic surfaces. Active flow control requires extra energy expenditure as well as actuation. It is often seen as blowing or suction which requires energy to power the jets.

With a large number of works published, this section will only review some example applications relevant to the present research to give a picture of how flow control can improve the performance. A comprehensive review of the flow control is provided by Gad-el-Hak (2001).

2.2.1 Passive Flow Control Examples

The vortex generator is a typical passive flow control application (e.g. Wendt and Dudek, 1998). It usually consists of placing arrays of small fins in the boundary layer. This can lead to the intensification of the mixing in the boundary layer to delay the separation. For example, it can be used to increase the lift on high lift airfoils, such as the work of Lin et al. (1994). In their work, they placed the vortex generators on the flap of a three-element airfoil at a wind tunnel test. Their results showed that the vortex generator could reduce the boundary layer separation on the flap for the landing configuration, so that the lift can be increased due to attached boundary layer and the drag can be reduced over a range of angle of attack. A review of the various applications of the vortex generator can be found, for example, in that by Lin (2002).

2.2.2 Active Flow Control Examples

Active flow control offers a high level of flexibility and control authority. There are many types of active flow control applications, such as airfoil/blade surface injection and suction, trailing edge blowing, circulation control, synthetics jets, plasma actuators, etc.

The injection and suction is a common type of active flow control. It has been used in separation suppression, blade wake loss reduction, airfoil lift enhancement, etc. An example of using injection for separation control and loss reduction is the work of Fernandez et al. (2013). They applied an array of steady micro jets on the blade suction surface that experiences flow separation. They found that the micro jets were able to reduce the integrated wake loss by as high as 85%. The reverse flow region on the suction surface was observed to be eliminated. The boundary layer on the suction surface was reattached due to increased local streamwise vortices by the micro jets, leading to enhance freestream entrainment to the near wall. The boundary layer is energized due to the enhancement of turbulent mixing, which further helps to reattach the boundary layer. However, the authors also found that if the blowing ratio is too large, the jets would lead to larger separation instead.

Suction has also been used for performance improvement. In turbomachinery, suction was used to reduce separation and loss and to increase the blade loading. The experiment by Reijnen (1997) applied suction on the suction surface of the blade of a transonic compressor stage and demonstrated that the suction could reduce the boundary layer thickness on the suction side, leading to higher flow turning and static pressure ratio, and delayed rotating stall. In another study, Gbadebo et al. (2008) computationally and experimentally investigate the aspiration on the suction surface and end wall at the end wall corner of a compressor stator cascade to control the three-dimensional corner separation. The three-dimensional RANS equations were solved and showed that the aspiration on the suction and the end wall removed the corner separation region, leading to increased blade loading and pressure rise, significant reduction of loss and produce more uniform outlet flow. The experiment in this study confirmed the computation results and showed that at the zero incidence angle, the loss was reduced by 22%, with increased flow turning up to 4°, and larger pressure difference between the pressure and suction surface.

An interesting area which has been hardly explored in the context of flow control is related to gas turbine blade cooling. Coolant is routinely ejected from blade surface primarily for generating a film layer to protect blade metal surface from the very hot gas. This source of high pressure coolant (typically taken from high pressure (HP) compressor) can in principle also be used as a means for flow control. In addition, a certain amount of coolant used for internal cooling needs to be exhausted anyway. The coolant of this kind is circulated internally in the blade to pick up heat (thus raise its own temperature in the process) are typically exhausted from the trailing edge, thus effectively acting similarly as a normal flow control. A direct consequence is the altering of the wake deficit, which can be used for reducing the noise, aerodynamic loss, besides the usual use for blade surface cooling. On the high pressure turbine blade, this trailing edge blowing of the cool gas is usually applied with pressure side cutback due to the thinness of the blade. From the cooling perspective, the adiabatic film effectiveness is usually used to measure the performance. This performance can be affected by many factors, such as effect of injection angles, slot lip thickness to height ratio, slot width to height ratio, density ratios and blowing ratio, as were investigated by Taslim et al. (1992). In particular, the taper angle of the lands (the supporting structures between the slots) was investigated by Ling et al. (2015). They experimentally studied the effect of the taper angle and showed that it had a strong effect on the jet mean flow and the adiabatic effectiveness in the cutback region as well as the flow in the wake.

The effect of the blowing ratio was investigated by several authors. Holloway et al. (2002a, b) and Schneider et al. (2012) reported a non-monotonic change of the adiabatic effectiveness with the blowing ratio, which was attributed by the authros to the large scale unsteady structures in the cutback injection region. Whereas Taslim et al. (1992) and Murata (2012) reported a monotonic change of the effectiveness with the blowing ratio. The internal geometry of the coolant plenum was also found to be influential on the flow and the surface temperature, as observed by Martini et al. (2005a) and Ling et al. (2013). Besides, Benson et al. (2012) also found that the flow and the adiabatic effectiveness were independent of the Reynolds number, when the flow was in the fully turbulent regime.

Another interesting application of the trailing edge injection is the circulation control. It is a flow control technology that can drastically increase the lift of an airfoil. The primary mechanism is the Coanda effect, that is the high momentum of the jet allows it to remain attached to the curved trailing edge for some distance before it separates (Slomski et al., 2006), thus increasing the circulation around the airfoil and wings and hence the lift. The capability to substantially increase the lift has attracted many researchers. For example, Min et al. (2009) used a RANS simulation method to investigate the circulation control on a symmetric NCCR 1510-7076N airfoil with a blunt trailing edge at low Mach number of 0.116 and zero angle of attack. The blowing slot is opened on the upper side of the blunt trailing edge. The result showed that as the blowing momentum coefficient increases, the lift coefficient can be increased to as high as 4, more than twice the maximum lift coefficient of a conventional airfoil. The result was in accordance with the experiment by Abramson (1977). Besides, the circulation control effect was also found to be sensitive to the shape of the Coanda surface, i.e the blunt trailing edge. For example, Schlecht and Anders (2007) experimentally tested a symmetric circulation control airfoil at Mach number of 0.3 with four elliptical and three biconvex Coanda trailing surfaces, and found that the elliptical Coanda surface with the most rounded shape produces the highest lift. Besides lab research, the circulation control concept has also been put on the actual flight test, such as the one on the A-6 intruder (Englar et al. 1979). As the main objective for circulation control is to achieve short take-off and landing, most investigations were conducted at low Mach numbers.

2.2.3 Numerical Flow Analysis for Flow Control

The flow fields associated with application of flow control tend to be complicated and unsteady. This is in clear contrast to the shape optimization where the flows tend to be relatively simple and steady in which RANS will be sufficient. As pointed out by Rizzetta et al. (2008), typical cases to apply the flow control may contain large regions of highly unsteady separated flows that are not in turbulent equilibrium. A wide range of scales of fluid structures may be present. Furthermore, flow control can be applied as unsteady forcing (e.g. oscillatory jet). In these conditions the time-accurate resolution is needed, which is beyond the capability of traditional RANS models. A review by Spalart (2000) also pointed out that to analyse the massively separated flow, it is debatable that RANS and URANS are suitable. Alternatively, Georgiadis et al. (2010) and Rizzetta et al. (2008) stated that it is suitable or required to use LES to predict the flow control flows. In the literature, though to the author's knowledge, there are yet no exclusive reviews about the numerical methods for the flow control simulation, there are still quite a number of numerical investigations of the specific flow control applications and related cases which have shown the deficiency of low-fidelity RANS or even URANS and the high accuracy of the LES, as briefly discussed below.

In many cases, the LES was generally applied to relatively simple flow control geometries to investigate the detail fluid dynamical mechanisms, such as the wall mounted hump injection (You et al., 2006), the backward facing step cavity injection (Rizzetta and Visbal, 2003), and airfoil injection (You and Moin, 2008). For example, You and Moin (2008) used the LES to simulate the synthetic jet on the NACA0015 airfoil at a high angle of attack. They studied the detailed interaction between the jet and the boundary layer to investigate the mechanism to control the flow separation. The predicted surface and wake profile agrees very well with the experiment for both the controlled and uncontrolled cases.

In more realistic cases such as the circulation control, existing results in RANS showed that the prediction can be highly dependent on the choice of turbulence models. For example, Slomski et al. (2002) and Rumsey and Nishino (2011) found that jet streamline and separation point varied considerably with different turbulence models. Moreover, Slomski et al. (2002), Swanson et al. (2005) and Chang et al. (2005) showed that unphysical numerical results could occur in which the jet wrapped around to the airfoil bottom or even all the way towards the leading edge, when some common RANS turbulence models were used, such as the k-ω SST, the Spalart-Allmaras, or the full Reynolds-stress model. This problem can be alleviated by using turbulence models with the curvature corrections (Swanson and Rumsey, 2009).

In the turbine blade cutback film cooling case, Holloway et al. (2002a, b) showed that the unsteady vortex shedding in the cutback region played an important role in the turbulent mixing between the coolant and the main flow. It has been shown by Medic and

Durbin (2005), Martini et al. (2005b), Egorov et al. (2010), Ravelli and Barigozzi (2013) that the RANS was unable to reproduce the experimentally measured film cooling effectiveness, resulting in the over-prediction of the adiabatic effectiveness, and also proposed that the unsteady effect should be resolved. However, it was shown by Medic and Durbin (2005), Joo and Durbin (2009) that the URANS still over-predicted the film cooling effectiveness significantly. Ling et al. (2014) showed that the over-prediction of the effectiveness by RANS was due to the under-prediction of the turbulence mixing by the RANS turbulence models. They reduced the turbulent Schmidt number of the eddy viscosity turbulence model to increase the turbulent diffusivity, and showed that the streamwise reduction of the effectiveness was captured and the error against the experiment was significantly reduced. Nonetheless, their reduction of the turbulent Schmidt number was still based on the assumption of isentropic turbulent viscosity in this case. On the other hand, the DES by Martini et al. (2005b) on a truncated domain showed a close match of the predicted cooling effectiveness with the experiment. The streamwise decrease of the effectiveness matches very well with the experiment, whilst the effectiveness from the RANS prediction remained almost constant streamwise. A comparison among RANS, URANS and LES and that with the experiment by Ravelli and Barigozzi (2014) also showed a significant difference in the adiabatic effectiveness between the RANS and the experiment. Also, though the URANS prediction was much closer to the experiment, it still over-predicted the effectiveness significantly. Only the LES prediction agreed well with the experiment.

2.2.4 Flow Control Optimization

The effectiveness of flow control in achieving higher performance depends on how it is designed and optimized. There are various previous efforts in optimization of flow control as seen in the literature, though the quantity and the comprehensiveness still appear to be much less than those for the shaping-alone optimizations. While there are investigations based on experiments, e.g. by Schlecht and Anders (2007), most of the flow control optimizations are based on numerical flow simulations. Some examples are presented below.

Han et al. (2010) applied a synthetic jet with a sinusoid blowing near the leading edge on the upper surface of a NACA0015 airfoil and optimized the blowing frequency, blowing angle and amplitude of the sinusoid blowing momentum repeatedly at different AOA ranging from 14° to 22°, at a Mach number of 0.1. As the blowing jet parameters were time-varying, URANS was used for the flow analysis and performance evaluation. Due to the high computational cost of evaluating a single design configuration, the Kriging surrogate model was employed in the optimization method as a surrogate to the true objective function to reduce the computational cost. Their result showed that, at a high AOA of 16°, the initial design of the flow control, when compared with the baseline airfoil without flow control, had no benefit in increasing the lift and reducing the drag. The extent of the large separation region on the upper surface near the trailing edge on the baseline airfoil was barely affected by the initial flow control design. But after the flow control optimization, the lift coefficient was increased by 16.9% and the corresponding drag coefficient was reduced by 13.4% with respect to the initial flow control design. The flow separation at the trailing edge was also nearly eliminated.

Yagiz et al. (2012) sought to minimize the drag of a two dimensional RAE5243 airfoil at 0.68 Mach number and 0.77° AOA, by applying and optimizing the passive and active flow controls on the upper surface. The passive flow control was a local contour bump and the active flow control was a steady suction actuator. The design variables for the passive contour bump were the length, maximum height, chordwise position and the crest position relative to the bump. The design variables for the actuator were the mass flow coefficient, chordwise position and the blowing angle. The gradient-based optimization method was used to find the optimum. The flow was analysed by solving the RANS equations. The flow control optimization was carried out in three approaches. In the first approach, only the passive flow control was applied and optimized with no suction. In the second approach only the suction was applied and optimized with no bump. In the third approach, both the bump and the suction were applied and optimized at the same time. The result showed that the highest performance was obtained by the hybrid third approach in which the drag reduction was 3.73% and lift increased by 5.71%, higher than the first and second approaches where drag reductions were 0.98% and 3.13% and lift increments were 0.54% and 3.17%.

2.3 Combining Shaping and Flow Control

As discussed, both shaping and flow control have been studied intensively for decades. However, to the author's knowledge, there are only a very small number of papers that have considered these two disciplines at the same time to achieve higher performance together. This section will give a review of these papers.

In the circulation control research, a few papers have included shaping with the flow control, in terms of optimizing the shape of the circulation control airfoil. An early example is the work by Tai et al. (1982) who optimized a circulation control airfoil to maximize the lift at low subsonic speed. The airfoil was represented by the linear combination shape parameterization method, with the combination coefficients being the design variables. As the circulation control was sensitive to the shape of the trailing edge, and the baseline shapes were taken to be several circulation control airfoils with different trailing edges. The vortex panel method was used for flow analysis. The optimization method was the steepest decent method. The result showed that the optimized airfoil increased the lift by 12% to 15% compared with the baseline circulation control airfoils. In this work, the flow control injection was kept constant, only the shape was optimized. Although the shaping was conducted in the presence of the flow control, the flow control itself was not optimized. The intent of the authors appeared to be exploring an optimal shape under a given fixed flow control, but not optimizing the flow control for further performance improvement. Also, the vortex panel method was too simplistic thus the flow prediction can hardly be deemed to be realistic. A similar work was also carried out by Tai and Kidwell (1985) at a high subsonic speed.

Probably the most relevant work as far as the present objectives are concerned is the recent study by Pehlivanoglu and Yagiz (2011). They applied the active flow control on a symmetric NACA 64A010 airfoil, and optimized both the airfoil shape and the flow control. The flow control was applied as a steady-state actuator with unified blowing/suction depending on the sign of the mass flow rate. Four cases corresponding to four optimization approaches were investigated: I) optimizing the flow control only, without changing the baseline airfoil shape; II) optimizing the airfoil shape only, without applying the flow control; III) concurrently optimizing the airfoil shape and the flow control; IV) sequentially optimizing the shape and the flow control, by first optimizing the shape using the results of case II, and then applying and optimizing the flow control on the optimized shape. The optimizations were carried out in a typical transonic cruise condition with a freestream Mach number of 0.85, AOA of 1° and Reynolds number of 607,000. In this condition, the predicted lift and drag coefficients of the baseline airfoil without flow control were 0.13 and 0.0329. The airfoil was parameterized with the Bezier curves, with the coordinates of the control points being the shape design variables. The mass flow coefficient, actuator location, and the blowing/suction angle defined as the flow control design variables. The flow field was analysed by the steady-state RANS equations with the Spalart-Allmaras turbulence model. The optimization algorithm was the non-gradient-based genetic algorithm so that the global optimum in the design space could be found. The objective was to maximize the lift to drag ratio. A compound objective function was adopted, containing the main objective which is to maximize the lift to drag ratio, and the penalty functions for the nonlinear constraints. A penalty function was activated if the lift coefficient was lower than the baseline value. Another penalty function was also included in shaping and activated if the maximum thickness was different than that of the baseline.

The results showed that the increase in lift to drag ratio is the lowest in Case I (flow control optimization only) – 211%, followed by Case III (concurrent shaping and flow control optimization) – 499%, and then Case II (by shaping only without flow control) – 521% and the highest in Case IV (sequentially optimizing the shape and then optimizing the flow control) – 576% . Thus the authors concluded that the sequential approach of shaping followed by the flow control optimization was the best. Also, it is interesting to note from their results that when concurrently optimizing the shape and flow control, the performance was actually worse than the shaping only, whereas optimizing the flow control on the baseline shape improved the performance. This seems to indicate some potential interaction between the shaping and the flow control. However, cautions must be taken in viewing these results. First of all, the thickness penalty function was only included in the shaping related optimizations. This would have very likely led to inconsistent constraints among the four optimization approaches, in that some of the cases could explore more design space than the others, leading to “unfair” comparisons. Therefore, the difference in the relative performance improvement between the cases and thus the authors’ conclusion is questionable. Secondly, there should be another sequential approach included in the comparison, which would be to optimize the flow control first and then optimize the shape. However, it had not been investigated. Thirdly, the geometry chosen for the investigation was too simplified. Given all these factors, it becomes unclear whether the findings from this particular research would have a general applicability.

Another recent study by Nielsen and Jones (2011) investigated the shaping and flow control optimization on a three-element airfoil system, with the leading edge slot and trailing edge flap deployed in the take-off or landing configuration. URANS was used for the flow solution. The flow control was applied as ten blowing jets distributed on the three elements of the airfoil. The objective was to maximize the lift coefficient. Two optimization approaches were used in the two case studies. In the first case, the shape and relative locations of the airfoil elements were fixed as the baseline values, and the flow control configurations were optimized. In the second case, both the flow control and the shape and relative locations of the airfoils are optimized concurrently. Because the number of design variables was relatively large, the gradient-based based optimization method was used in order to use the adjoint method to efficiently compute the gradient of the objective function. The results showed that, in the first case, when only the flow control was optimized, the lift was increased by 22%. In the second case, when the shape and flow control were optimized concurrently, the lift was increased by 27%. Thus, the results appeared to show that combining the shaping and flow control would produce better performance than optimizing the flow control alone. However, caution should be taken in accepting this conclusion. First of all, as the gradient based method was used for the optimization, it is likely that the search starting from the baseline design could be trapped in a local optimum, so that the optimized design in each case may not be the true global optimum in the design space of the corresponding optimization approach. Moreover, as the main intent of the paper was not to compare different optimization approaches, the other possible approaches were not investigated. Thus, neither would this work sufficiently demonstrate the benefit of combining shaping and flow control optimization relative to the single-disciplinary optimization, nor would it provide a systematic guide on the sequencing of the disciplines in a combined optimization.

To the best knowledge of the present author, the above papers are all that can be found to have considered both shaping and flow control. Compared with the numerous research papers in each of the two disciplines, it is a stark contrast in quantity. These papers, though rather small in number, have suggested that combining the shaping and flow control would bring about a different design and a higher performance improvement than only considering the shape or flow control. Also they have indicated considerable differences which can result from sequencing the disciplines in the final design. However, as the above discussions shows, all of these works stem from the purposes to solve a particular flow control problem. None of them were spawned from or followed up to formally addressing the concept of combining shape optimization and flow control optimization in a general sense. Therefore, the findings in these works are not considered to be comprehensive, or even consistent. They are thus regarded as inconclusive for general applicability.

2.4 Numerical Optimization Methods

In many situations, an optimization is the process of finding the best candidate in the design space, sometimes subject to certain constraints. It requires evaluating the objective function of the candidates as well as the trend of how the objective function changes in the design space, so as to know which candidates are better ones or how to modify the candidates to produce even better ones. Based on different criteria, the optimization methods can be classified in different manners, such as direct design versus inverse design, deterministic versus heuristic, etc. Based on the manner of searching the design space, the optimization methods can also be classified into gradient-based and non-gradient-based methods.

2.4.1 Gradient-Based Optimization Methods

In the gradient-based method, the design space is explored by computing the gradient of the objective function of the current design, so as to inform the search direction for an improved design. The optimization cycle converges when the gradient is almost zero or the boundaries of the constraints are met.

One of the most efficient gradient computation methods is the adjoint method proposed by Jameson (1988). Regardless of the number of design variables, the adjoint method only requires one flow solution and one solution of the adjoint equation to compute the gradient. Thus, it is very efficient for cases with large number of design variables and becomes quite popular nowadays in the gradient-based optimizations of the practical cases (Walther and Nadarajah, 2015; Zingg and Elias, 2006; Lepine et al., 2001; etc). For example, Wang and He (2010) coupled the adjoint method with the mixing plane model, which enabled the adjoint design optimization in multi-stage turbomachinery (Wang et al., 2010). In their cases, up to 1023 design variables were used for a 3.5 stages transonic compressor, while the computational cost was still affordable. A drawback of the adjoint method is that it is not easy to implement into the flow solver (Anderson and Bonhaus, 1999; Neilson and Anderson, 1999). Also, it may not always be readily available in the mainstream commercial solvers.

The gradient-based method is apt to be trapped in a local optimum if the objective function is multi-modal, and the final design will likely be dependent on the chosen initial configurations (Narducci et al, 1995; Elliott and Peraire, 1997; Buckley et al, 2010;). If the unique global optimum is desired, some measures can be used to reduce the possibility of local optima trapping, such as by using multiple staring points (Ugray et al., 2007; Peri and Tinti, 2012; Chernukhin and Zingg, 2013), or by the hybrid with global search methods (Vicini and Quagliarella, 1999; Chernukhin and Zingg, 2013). Also, successful use of the gradient-base method requires the objective function to be continuous and smooth.

2.4.2 Non-Gradient-Based Optimization Methods

Non-gradient-based methods do not require the gradient of the objective function locally at the design sites. Rather these methods are more often focused on exploring the global trend of the objective function and are used more often in the global optimization in search of the global optimum. Common methods include the genetic algorithm and the surrogate models.

The genetic algorithm (GA) is a popular method for global optimizations in shaping (Rallabhandi and Mavris, 2007; Okui et al., 2013) and flow control optimizations (e.g. Huang et al., 2007). It explores the design space by selecting the best designs from the current generation of design population, and applies the mutation, crossover operations to generate a better generation of population. As the designs in the population usually span across the whole design space, the GA is robust in finding the global optimum. It is also suitable for multi-objective optimizations in which the optimal designs form a Pareto front (Rallabhandi and Mavris, 2007). However, as in each generation the design population is usually large, the GA can be very time consuming if the designs are evaluated directly using physics based models, e.g. solving flow governing equations.

The surrogate model (Queipo et al., 2005; Gano et al., 2006; Forrester and Keane, 2009; Viana et al., 2014) can be used to reduce the computational cost of exploring the global design space by replacing the expensive physics-based models with cheap surrogate models. A surrogate model is constructed based on the true objective function evaluated by the physics-based model at a few sample points across the design space. Thus it is able to approximate or interpolate the global trend of the true objective function, and identify the approximate location of the global optima. Once it is constructed, the GA (Jeong et al., 2005; Song and Keane, 2007; Schabowski et al., 2014) or the multi-start gradient-based search (Ugray et al., 2007; Peri and Tinti, 2012) can be used to find the global optimum of the surrogate objective function that approximates the true global optimum. Another benefit of the surrogate model is that it can be used to visually represent the trend of the objective function to facilitate the investigation of the optimization problems (Zhang and He, 2014).

The response surface method (Myers et al., 2004; Myers et al., 2016) is a common surrogate model. In most situations it uses a quadratic surface to approximate the real objective function. The model is relatively simple and is cheap to construct and evaluate, and has been used in many practical optimizations (Ahn et al., 2001; Schabowski et al., 2014). However, since the quadratic polynomial is uni-modal, it cannot approximate the multi-modal (i.e. multiple local optima) objective functions (Giunta and Watson, 1998).

The Kriging method is another popular surrogate model (Sacks et al., 1989; Simpson, 2001; Martin and Simpson, 2005). The Kriging model passes through all the sample responses and interpolates the true objective function in the design space. It is able interpolate highly nonlinear and multi-modal objective functions. Hence it is suitable to identify both the local and global optimum. As the objective function in shaping or flow control optimization problems is likely to be multi-modal (Buckley et al., 2010; Chernukhin and Zingg, 2013), the Kriging seems suitable for this situation. Due to its effectiveness and flexibility, the Kriging has been widely used in shaping (Simpson et al., 2001; Keane, 2003; Jeong et al., 2005; Paiva et al., 2010) and flow control optimizations (Meunier, 2009; Han et al., 2010; Duvigneau and Chandrashekar, 2012).

The design of experiment (DOE) method (Giunta et al., 2003) is required to choose the sites of the sample designs for the construction of the surrogate model. Since the sample designs are evaluated using the expensive physics based model, it is hence desirable to use efficient sampling strategy such as the DOE to select as fewer samples as possible to obtain as much information about the global trend of the real objective function. Various DOE methods have been developed for numerical optimizations, such as the orthogonal arrays (Owen, 1992), Latin Hypercube method (McKay et al., 2000), Uniform Design (Fang et al., 2000), etc. Despite the use of DOE, the number of required samples will still be large with a large number of design variables, which is known as the "curse of dimensionality" (Koch et al., 1999).

2.5 Efficient Methods for Unsteady Flow Analysis

When the expensive unsteady simulation is required to provide accurate solution of complex flows fields in the design optimization, it is highly desirable to develop efficient methods to reduce the computational cost but maintain the prediction accuracy. With regard to the objective of this thesis, it is desirable to use the LES wisely in the optimization, possibly coupling with other cheaper flow analysis methods, to maximize its benefit and minimize the required number of runs. To serve this objective, this section will review the previous efforts in developing the efficient methods for the unsteady flow analysis. More often than not, these methods made use of the specific properties or requirements of the problems at hand to achieve their purposes.

2.5.1 Harmonic Methods

In the turbomachinery flows, the frequency domain methods have been used to effectively reduce the computational cost of the URANS analysis. One example is the Nonlinear Harmonic Phase Solution method proposed by He (2008). The method makes use of the periodic nature of the blade row interaction or vibration and the consequential temporal periodicity of the flow, and expresses the flow solution in a Fourier series in terms of the frequency of the period. As a result, the unsteady flow equations can be transformed into a set of steady-state-like equations at the different phases of the period of unsteadiness. If one harmonic is retained, which is usually sufficient for turbomachinery blade aeroelasticity modelled by URANS, three phases in a period can be chosen, resulting in three steady-state equations regarding the coefficients of the harmonic and the time-averaged solution. Therefore, the method has transformed the expensive time-domain integration to the solution of three steady-state equations, significantly improving the efficiency in the URANS analysis. Besides, other frequency domain methods, such as the Harmonic Balance method (Hall et al., 2002), the Nonlinear Frequency Domain Method (McMullen, 2003), can also be used to improve the efficiency of the unsteady flow analysis based on the similar theory. The harmonic method has also been applied for concurrently combined aerodynamic and aeroelastic optimizations (He and Wang, 2011).

2.5.2 Deterministic/Unsteady Stress Methods

In many design optimizations, the designers are mainly interested in the time mean effect. Thus in the URANS simulations, the ‘deterministic stress’ or ‘unsteady stress’ methods have been used to efficiently compute the time-mean unsteady flow. Instead of performing the expensive time integration and averaging the solution of each time-step, these methods place the deterministic stresses in the source terms of the RANS equations, and use the RANS to compute the time-mean flow.

Due to the nonlinearity, time-averaging the unsteady flow equations will produce extra ‘stress’ terms arisen from any unsteadiness. The concept and system equations of the so called ‘deterministic stress’ were firstly derived by Adamczyk (1985) for bladerow interaction related unsteadiness. The term ‘deterministic’ was adopted as the frequency (as well as circumferential length scale) of the unsteadiness are linked to rotor rotating speed and blade counts, all being known prior to the solution. Adamczyk’s original effort has been attempted to model these stress terms. Later on, the nonlinear harmonic method with multiple harmonics was proposed and developed by He and Ning (1998) to efficiently compute (rather than model) these blade-row interaction related deterministic stress terms.

The deterministic stresses can also be computed by the Lumped Deterministic Source Term (LDST) method, proposed by Sondak et al. (1996) and later used by Busby et al. (1999) for deterministic blade row interaction. In this method, the time-averaged URANS flow solution can be put into the RANS solver as the solution. The residual of the RANS solver is then the lumped deterministic source term for each equation, which is the sum of all the deterministic stresses for that equation. However, the expensive URANS simulation was still needed to obtain the time-averaged solution for the computation of the lumped deterministic source term. Alternatively, the deterministic stresses can be modelled using the transport equations, analogous to modelling the Reynolds stresses with the turbulence transport equation, by noticing the similar appearance of the two types of stresses in the time-averaged unsteady equations. Stollenwerk and Nuernberger (2008) and Stollenwerk and Kugeler (2013) used the Boussinesq’s approximation to model the deterministic stresses, and solved the transport equation for the deterministic pseudo-viscosity together with the RANS equations. Wall et al. (2000), Charbonnier and Leboeuf (2004) used deterministic stress transport equations to model the deterministic stresses, analogous to modelling the Reynolds stresses with the Reynolds stress transport equations. The transport equations are solved together with the RANS equations. The advantage of the transport equation methods is that all the time-mean solution can be computed directly by RANS without requiring any unsteady computation beforehand. However, as the transport model has no input from any actual time-averaged solution, how best the RANS-predicted time-mean solution approximates the true time-mean, will depend on the validity and accuracy of the transport equation models.

In a fundamental sense, the deterministic stresses generated by time-averaging the unsteady flow equations are essentially the same as the Reynolds stresses that are due to the turbulence. Very much in a same way as that the accuracy of RANS solution depends on the modelling of Reynolds stress terms, the accuracy of an time-averaged URANS solution depends on the modelling of the deterministic stresses. The main difference between the two types is that the bladerow-induced unsteadiness is externally driven with known periodicity, whilst turbulence is purely self-driven. This difference will have a considerable bearing in the present work, thus needs to be examined further.

A notable original effort to link the two types of unsteadiness (forced vs self-excited) is the work by Ning and He (2001). They firstly performed a nonlinear time-domain URANS simulation on a cylinder and a model turbine blade passage. In these cases, purely self-excited unsteadiness due to vortex shedding was captured in the time-domain solution. The unsteadiness, though coherently periodic, has no known frequency, which has implication both on the time-averaging and the stability of the solutions. As such, the corresponding stresses are called ‘unsteady stresses’ rather than ’deterministic stresses’. Both laminar flow and turbulence flows were considered, and for the latter the Reynolds stresses were still modelled by the turbulence model. From converged unsteady solutions, the ‘unsteady stresses’ were obtained through post-processing. The results showed that on the same geometry, the RANS simulation with the unsteady stress terms converged quite well and matched the time-average URANS solution quite well. On the other hand, the RANS simulation without the deterministic stress remained oscillatory and could not converge to a steady-state solution. This study showed that not only the unsteady stresses could help to balance the RANS solution approaching that of the time-averaged flow field, but also they played a stabilizing role in getting the steady-like solution.

Similar to the lumped deterministic stress method for the deterministic stresses (Sondak et al., 1996; Busby et al., 1999), lumped unsteadies stress source terms can be obtained by using the time-averaged flow variables to calculate flux residuals for self-excited unsteadiness, as shown by He (2008) for the limit cycle type of flow instabilities. In this case, frequencies of the unsteadiness were unknown. Thus sufficiently long time-traces would be needed for time-averaging, and this was done in a simple on-the-fly averaging during the time-domain solution (He, 2008). Galbraith and Orkwis (2008) used the lumped method to compute the lumped unsteady stresses for the cylinder flow and unsteady synthetic jet flow. Lukovic et al. (2006) used the lumped deterministic stress method to compute the self-driven unsteady cavity flow. In addition, they also used the artificial neural network to propagate the lumped unsteady stress terms from the URANS-evaluated baseline geometries to the similar new geometries and flow conditions, to solve the time-averaged URANS solutions there in the RANS mode. The results showed that the RANS solution with the propagated lumped unsteady stresses was much closer to the time-averaged solution URANS than the RANS solution with turbulence model on the new configuration, though the match between the steady-state solution with unsteady stresses and the time-averaged URANS was not very close.

2.6 Summary

The shaping and flow control have been studied extensively for decades. Both have shown to be quite effective in improving the performance of the aerodynamic devices. The study of shaping tends to be more focused on the computational efficiency of the optimization. On the flow simulation side, the well-developed but low-fidelity steady-state

RANS simulations are usually shown to be sufficient and are often chosen to provide fast performance evaluations.

On the other hand, existing flow control studies tend to be focused more on the complicated fluid dynamical mechanisms, due to the complexity of the flow control related flow field. Also, a considerable amount of researches has shown that the high-fidelity unsteady numerical methods like LES may be required to simulate the flow accurately. There are also numerical optimization studies for flow control but the quantity is much smaller than those in shape optimization.

However, the efforts to the two disciplines in the optimization are still very rare. From the few existing papers, it has been indicated that there can be the potential to further improve the performance by combining the optimizations of the shape and flow control. However, these works were either based on very low-fidelity flow predictions and simplified 2D geometries, or did not include comprehensive and consistent comparisons. Thus the results were inconclusive.

The non-gradient based optimization methods based on the surrogate models, such as the Kriging method, have been shown to be able to capture the global optimum of the design space efficiently. This ability will be required to compare the potential of different optimization approaches (e.g. combined versus un-combined) in a consistent manner.

It is recognised that a combination of shaping with flow control optimization, if attempted, would bring conflicting requirements between the computational efficiency and the high fidelity in performance evaluation in the optimization. Several methods were developed in the past to analyse the unsteady flow field more efficiently: the harmonic method, the deterministic and unsteady stress methods. These methods exploited the specific properties or requirements of the corresponding problems to transfer the expensive unsteady-simulation to the faster steady-state simulations. However, the harmonic methods and the deterministic/unsteady stress methods were developed only for the URANS.

Chapter 3

Methods for Optimization and Baseline Flow Simulation

This chapter introduces the methods used in this work for the optimizations and the baseline flow simulations. The optimization methods sections discuss the optimization approaches, procedures and algorithms. The baseline flow simulation methods include the steady-state simulation based on the Reynolds Average Navier-Stokes (RANS) equations and the time-accurate large eddy simulation (LES). The word “baseline” is used to differentiate the well-established flow simulation methods descripted in this chapter from the new efficient LES method to be presented in later chapters.

3.1 Optimization Methods

3.1.1 Optimization Approaches

An optimization problem is generally formulated as maximizing or minimizing the objective functions based on the main performance of interest. The extremum of the objective function is reached by finding the optimal values for the design variables that govern the design, subject to the constraints on the design variables and the other performances. The optimization can be single-disciplinary or multiple-disciplinary, in which the design variables belonging to one or multiple disciplines are optimized. In many practical cases, the objective function and the constraints need to be evaluated from the time-consuming physics-based simulations, such as computational fluid dynamics (CFD). To facilitate the optimization, the physics-based evaluation of the objective and constraints can be replaced by the fast-to-evaluate surrogate models.

To investigate the effect of combining shaping and flow control optimization, the present work seeks to carry out the optimizations using different optimization approaches and investigate the best performance improvement that can be obtained respectively by each approach. In total, five optimization approaches can be identified. They are:

1) shaping only, without flow control;

2) flow control optimization only, without shaping;

3) shaping and then flow control optimization;

4) flow control optimization and then shaping;

5) concurrent shaping and flow control optimization.

Approach 1 and 2 are the conventional approaches used in the shaping and flow control optimization in many previous studies, hence serving as references. The last three approaches are the combined ones, which were rarely investigated in the past. Accordingly, the formulation of the optimization problem for each approach is described below.

Approach 1: shaping only. The objective and constraints are functions of the shape design variables only. The shape design variables are optimized. The flow control is not present.

Approach 2: flow control optimization only. The objective and constraints are functions of both the shape and the flow control design variables. The shape design variables are fixed as baseline. Only the flow control design variables are optimized.

Approach 3: shaping and then flow control optimization, i.e. the first sequentially combined approach. The first step is identical to Approach 1. In the second step, the objective and constraints are functions of both the shape and flow control design variables. The shape design variables are fixed at the optimized values. Only the flow control design variables are optimized.

Approach 4: flow control optimization and then shaping, i.e. the second sequentially combined approach. The objective and constraints are functions of both the shape and flow control design variables. The first step is identical to Approach 2. In the second step, the flow control design variables are fixed at the optimized values. Only the shape design variables are optimized.

Approach 5: optimizing the shape and flow control together, i.e. the concurrent combined approach. The objective and constraints are functions of the shape design variables and the flow control design variables, both of which are optimized at the same time.

3.1.2 Kriging-Based Optimization Procedure

A. Overall Description

To compare the maximum performance that can be obtained respectively by each optimization approach, the key requirement is to find the global optimum in the design space of each optimization approach. This requires the global exploration of the design space. A surrogate model can be used to explore the design space efficiently. To construct a well representative surrogate model, evaluations of the objective and constraints at a few representative sample sites in the design space are required. The samples sites are usually chosen by the method of design of experiment (DOE). After the surrogate model is built, it can be updated during the optimization to further improve its ability in finding the global optimum as well as precisely pin-pointing its location.

An optimization procedure is developed, as shown in Figure 3-1. The procedure is able to consistently carry out the optimization in all the optimization approaches. The present framework is built around the Kriging surrogate model. The procedure is made modular. The DOE method, the CFD evaluation method, the search method on the surrogate function, and the method to update the surrogate model, are made compatible to different choices of in-house or commercial codes. In the DOE module, the Uniform Design (Fang, 2000) method is chosen. In the CFD module, the ANSYS ICEM CFD and

Fluent packages are chosen for the meshing and flow solution respectively. The Sequential Quadratic Programing (Han, 1977) with Multi-start (Ugray et al., 2007) option is used to search for the global optimum on the surrogate objective function. The Statistical Lower Bound (SLB) method (see Jones, 2001; Cox and John, 1997) is used to update the Kriging surrogate model.

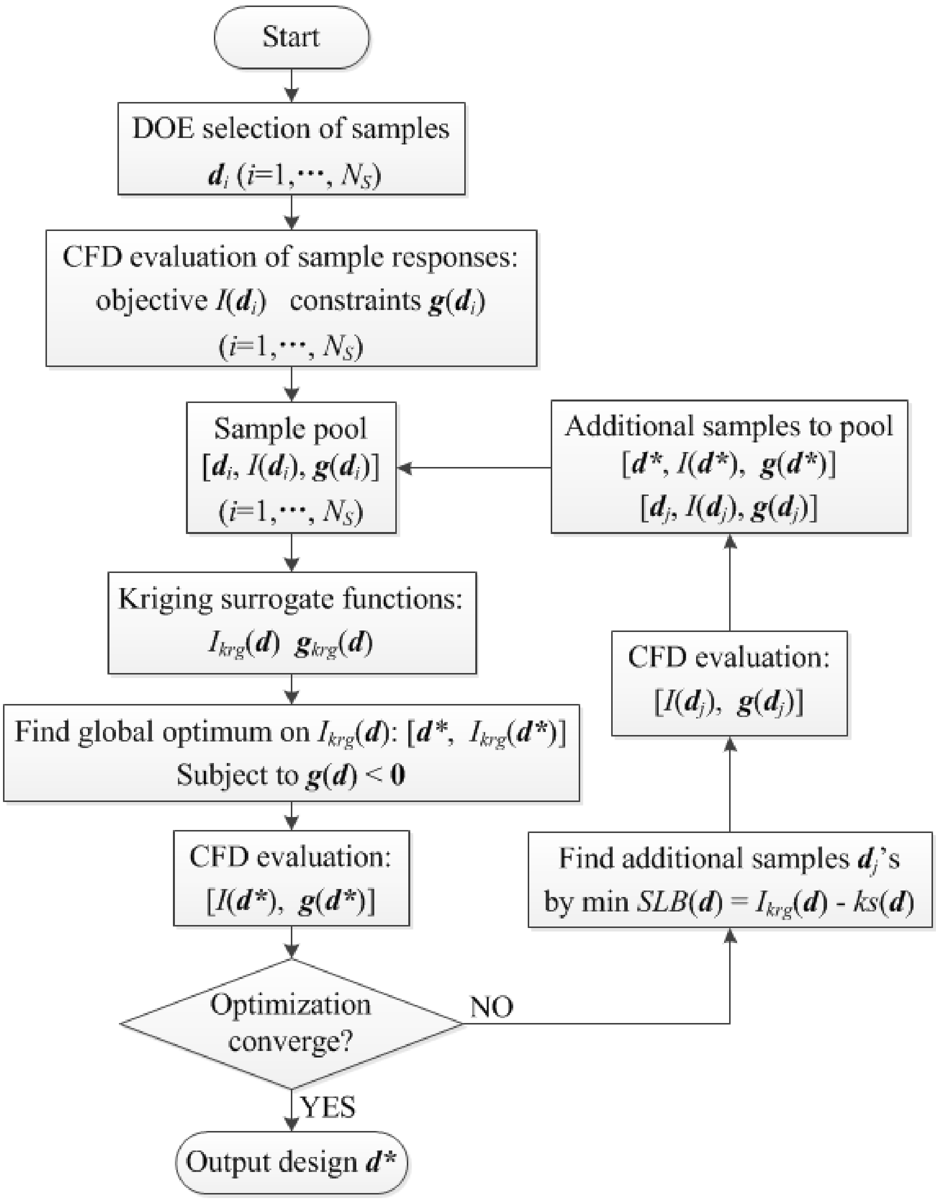


Figure 3-1 The Kriging-based optimization procedure

The overall procedure is illustrated in Figure 3-1 above. Firstly, the DOE is used to choose the sample sites in the design space. Next, the CFD is applied to evaluate the responses of the objective function and the constraint functions of the sample designs. Next, the Kriging surfaces are built to interpolate the sample responses across the design space, acting as the surrogates for the real objective function and the real constraint functions. Thirdly, the multi-start SQP method is applied to find the global optimum on the surrogate objective function, subject to the surrogate constraints and the constraints of the design variables. Next, the real objective function of the global optimum found based-on the surrogate objective function is evaluated and compare to the surrogate objective function to check the convergence. If the optimization has not converged, the SLB method is used to further explore the design space by locating additional sample sites and evaluate them using CFD. Then, the current global optimum and the additional samples are added to the sample pool. The Kriging model is then rebuilt upon the updated sample pool for the next optimization cycle. To reduce human error, the procedure can be executed fully automatically, including the meshing and the flow analysis. The whole procedure is controlled by a MATLAB program running alongside to manage the module execution and data I/O.

There are several criteria for the convergence. First, the predicted global optimum must have changed by less than 1% for three consecutive cycles. Second, the relative error of the surrogate objective response of the global optimum should be less than 0.5% compared with the real response. Third, the real constraints should not be violated. The change of the optimum is defined as:

$$\frac{\|\boldsymbol{d}^{*,m} - \boldsymbol{d}^{*,m-1}\|}{\Delta_d}$$

where, $\boldsymbol{d}^{*,m}$ is the vector of design variables of the surrogate global optimum produced by the m-th optimization cycle, and $\|\cdot\|$ is the $\mathcal{L}^2$-norm. Δ_d is length of each dimension of the design space, same for all the dimensions as the design space is normalized in the optimization. The relative error of the surrogate objective function is defined as:

$$\frac{\left|I_{krg} - I_{real}\right|}{\left|I_{real}\right|}$$

Where I_{krg} is the surrogate objective function and I_{real} is the real objective function, both evaluated at the current global optimum.

B. Kriging Surrogate Model

The Kriging interpolation is chosen as the surrogate model for its ability to interpolate nonlinear, unimodal or multimodal functions, a desired feature in finding the global optimum, as the objective function in this study can be multi-modal. In addition, the Kriging model is not too complicated to build. The principle of the Kriging model is described here. More details can be found in Sacks et al. (1989). In this study, the model is constructed using the MATLAB Kriging Toolbox developed by Lophaven et al. (2002).

In the Kriging model, the surrogate function $y_{krg}(\boldsymbol{d})$ for the real function $y(\boldsymbol{d})$ (e.g. the objective, the constraints, etc.) at a site $\boldsymbol{d}$ in the N_V dimensional design space can be viewed as the sum of the value of a deterministic trend function plus and a stochastic distribution, expressed as:

$$y_{krg}(\boldsymbol{d}) = \boldsymbol{f}^T(\boldsymbol{d})\boldsymbol{\beta} + Z[\boldsymbol{d}] \quad \textbf{(3.1)}$$

where $\boldsymbol{d}$ is a vector containing N_V design variables, $\boldsymbol{f}^T(\boldsymbol{d})\boldsymbol{\beta}$ is the trend function and $Z[\boldsymbol{d}]$ is the stochastic function. In this work the trend function is chosen to be a quadratic polynomial, in which $\boldsymbol{f}(\boldsymbol{d})$ is a vector containing a set of regression basis functions and $\boldsymbol{\beta}$ is a vector containing the coefficients for the regression basis functions.

The stochastic function $Z[\boldsymbol{d}]$ literally means the probabilistic distribution Z of the site $\boldsymbol{d}$, rather than being a function of $\boldsymbol{d}$. In this study Z is taken to be the same for any $\boldsymbol{d}$, and is assumed to be a Gaussian probability distribution. The mean and variance for the distribution are chosen to be 0 and σ^2. The covariance of the distributions between any two sample sites $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$ is expressed as:

$$\operatorname{cov}\{Z[\boldsymbol{d}_i], Z[\boldsymbol{d}_j]\} = \sigma^2 R(\boldsymbol{d}_i, \boldsymbol{d}_j)\ , \tag{3.2}$$

where $R(\boldsymbol{d}_i, \boldsymbol{d}_j)$ is the correlation between the sample sites $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$. The formulae of $R(\boldsymbol{d}_i, \boldsymbol{d}_j)$ is determined by the type of the correlation chosen. In this work, the Gaussian correlation is adopted (see Lophaven et al., 2002):

$$R(\boldsymbol{d}_i, \boldsymbol{d}_j) = \exp\left[-\sum_{k=1}^{N_V} \theta_k \left(d_i^k - d_j^k\right)^2\right], \tag{3.3}$$

in which d_i^k is the k-th element of $\boldsymbol{d}_i$, and θ_k is the correlation parameter for the k-th element. Likewise, the correlation between an arbitrary site $\boldsymbol{d}$ and a sample site $\boldsymbol{d}_i$ is denoted as $R(\boldsymbol{d}_i, \boldsymbol{d}_j)$. The correlations regarding the N_S number of sample sites can be grouped as a matrix $\boldsymbol{R}$ and a vector $\boldsymbol{r}$ as:

$$\boldsymbol{R} = \begin{bmatrix} R(\boldsymbol{d}_1, \boldsymbol{d}_1) & \cdots & R(\boldsymbol{d}_1, \boldsymbol{d}_{N_S}) \\ \vdots & \ddots & \vdots \\ R(\boldsymbol{d}_{N_S}, \boldsymbol{d}_1) & \cdots & R(\boldsymbol{d}_{N_S}, \boldsymbol{d}_{N_S}) \end{bmatrix} \tag{3.4}$$

and

$$\boldsymbol{r} = \left[R(\boldsymbol{d}, \boldsymbol{d}_j) \ \cdots \ R(\boldsymbol{d}, \boldsymbol{d}_{N_S})\right] \tag{3.5}$$

Following the derivation in the work of Sacks et al. (1989) or Lophaven et al. (2002), the Kriging-predicted response of the objective function at site $\boldsymbol{d}$ can be expressed as:

$$y_{krg}(\boldsymbol{d}) = \boldsymbol{f}^T(\boldsymbol{d})\boldsymbol{\beta} + \boldsymbol{r}^T\boldsymbol{R}^{-1}(\boldsymbol{Y}_S - \boldsymbol{F}\boldsymbol{\beta}) \tag{3.6}$$

in which $\boldsymbol{F} = \left[\boldsymbol{f}^T(\boldsymbol{d}_1) \ \cdots \ \boldsymbol{f}^T(\boldsymbol{d}_{N_S})\right]^T$, $\boldsymbol{Y}_S$ is the vector containing N_S sample responses. The regression function coefficients $\boldsymbol{\beta}$, the variance σ^2, and the parameters θ^k's in $R(\boldsymbol{d}_i, \boldsymbol{d}_j)$ can be calculated by the maximum likelihood estimation, resulting in:

$$\boldsymbol{\beta} = (\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F}^T)^{-1}\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{Y}_S \tag{3.7}$$

$$\sigma^2 = \frac{1}{N_S}(\boldsymbol{Y}_S - \boldsymbol{F\beta})^T \boldsymbol{R}^{-1}(\boldsymbol{Y}_S - \boldsymbol{F\beta}) \quad \textbf{(3.8)}$$

As mentioned previously, if the current optimization iteration has not converged, the Kriging model will need to be updated for the next optimization cycle using the sample pool reinforced with additional samples. In this study, the SLB method, as illustrated by Jones (2001), is used to select additional sample sites to enhance the exploration of the design space. Assumed that the objective function $I(\boldsymbol{d})$ is to be minimized, the additional sites are calculated by minimizing:

$$slb(\boldsymbol{d}) = I_{krg}(\boldsymbol{d}) - k \cdot s(\boldsymbol{d}), (k > 0) \quad \textbf{(3.9)}$$

where $s(\boldsymbol{d})$ is an estimate of the possible root mean squared (RMS) error of the Kriging prediction at site $\boldsymbol{d}$ (formulation see Sacks et al, 1989). As $s(\boldsymbol{d})$ is higher at the less sampled region, a larger k will lead to an additional sample located in the less sampled region, whereas a smaller k will lead to an additional sample located near the current minimum. Multiple values for k can be chosen to satisfy both global exploration and local refined sampling. According to Jones (2001), possible values for k can be 5, 10 or etc., depending on the problem. After the additional samples and the current global optimum are evaluated by the CFD, they are added to the sample pool for the update of the Kriging model.

C. Design of Experiment (DOE)

The DOE method is used to select the sites for the initial samples. If the number of design variables is large, the high-dimensional design space will require a large number of samples to comprehensively explore the trend of the objective and constraint functions. As the samples are evaluated by the expensive CFD, the DOE method is used to provide more efficient design space sampling.

In this research, the Uniform Design method by Fang et al. (2000) is used. It has previously been used along with the Kriging model (e.g. Han et al., 2010). The method selects the sample sites by maximizing the uniformity of the samples' distribution (detail in Fang et al., 2000) in the design space. Based on the dimensional of the design space, the calculated sample distributions are formulated as the uniform design table (Fang et al., 2000, or comprehensively in website http://uic.edu.hk/isci/UniformDesign/table/CD2.htm).

D. Simplification of Optimization Procedure

If the dimension of the design space is small and the turn-around for the CFD evaluation is short, the above optimization can be simplified. For example, the Kriging can still be constructed but without update, and the design space can be sampled with full-factorial uniform sampling in each dimension without using DOE. More radically, the space can be filled simply by dense enough of samples and simply select the best one as the global optimum, provided that the entire procedure is not computationally costly.

3.2 Numerical Flow Simulation Methods

3.2.1 Governing Equations

The flow field can be described mathematically by a set of partial differential equations: the continuity, momentum and energy equations, grouped together called the Navier-Stokes (NS) equations. For brevity, the Einstein notation is adopted, the Cartesian coordinate in the x, y, z directions are indexed by $i = 1, 2, 3$. Repeated occurrence of an index, such as $u_k n_k$ implies the summation over all the indices. For general compressible flow, the NS equations are expressed as:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x_i} = \frac{\partial \boldsymbol{F}_{v,i}}{\partial x_i} \qquad \textbf{(3.10)}$$

where vector $\boldsymbol{U}$ contains the solution variables in the conservation form. $\boldsymbol{F}$ and $\boldsymbol{F}_v$ are the inviscid and viscous flux vectors. They are expressed as:

$$\boldsymbol{U} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{bmatrix}, \boldsymbol{F} = \begin{bmatrix} \rho u_i \\ \rho u_1 u_i + p\delta_{1i} \\ \rho u_2 u_i + p\delta_{2i} \\ \rho u_3 u_i + p\delta_{3i} \\ u_i(\rho E + p) \end{bmatrix}, \boldsymbol{F}_{\boldsymbol{v},\boldsymbol{i}} = \begin{bmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ u_j \tau_{ji} + \kappa \frac{\partial T}{\partial x_i} \end{bmatrix} \tag{3.11}$$

u_i is the velocity in the i-th direction. $E = e + \frac{1}{2} u_k u_k$ is the total energy where e is the internal energy. τ_{ji} is the viscous stress tensor computed as:

$$\tau_{ji} = \mu \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) + \lambda \left(\frac{\partial u_k}{\partial x_k} \right) \delta_{ji} \tag{3.12}$$

where μ is the molecular viscosity, $\delta_{ji}$is the Dirac delta. Based on the Stokes' assumption, $\lambda = -\frac{2\mu}{3}$. κ is the thermal conductivity of the air. In this research, the body force is not considered hence it is not included in the momentum and energy equations.

3.2.2 Basic Aspects of Flow Solver

A. Finite Volume Discretization

The commercial solver package ANSYS Fluent v13.0 is used for flow analysis. Fluent uses the finite volume (FV) method to discretize the integral form of the flow equations. The solution variables are stored at the mesh cell centroid. The flow equations in the integral form after the FV discretization are expressed as:

continuity:

$$\frac{\partial \rho}{\partial t} \Delta V + \sum_{f}^{N_{face}} \rho_f \vec{u}_f \cdot \overrightarrow{A_f} = S_m \cdot \Delta V \tag{3.13}$$

momentum in the i-th direction:

$$\frac{\partial \rho u_i}{\partial t}\Delta V + \sum_f^{N_{face}} \rho_f u_{i,f}\left(\vec{u}_f \cdot \overrightarrow{A_f}\right) = -\sum_f^{N_{face}} p_f\left(\vec{n}_{i,f} \cdot \overrightarrow{A_f}\right) + \sum_f^{N_{face}} \vec{\tau}_{i,f} \cdot \overrightarrow{A_f} + S_i \cdot \Delta V \qquad \textbf{(3.14)}$$

energy:

$$\frac{\partial \rho E}{\partial t}\Delta V + \sum_{\mathrm{f}}^{\mathrm{N_{face}}} \left(\rho_f E_f + p_f\right) \cdot \left(\vec{u}_f \cdot \overrightarrow{A_f}\right)$$
$$= \sum_f^{N_{face}} \kappa\left(\overrightarrow{\nabla T}\right)_f \cdot \vec{A}_f + \sum_f^{N_{face}} \left(u_{i,f}\, \vec{\tau}_{i,f}\right) \cdot \overrightarrow{A_f} + S_e \cdot \Delta V \qquad \textbf{(3.15)}$$

In above, ΔV is the volume of the mesh cell. $\vec{A}_f$ is the outward normal vector of the cell face with the module equal to the face area. S is the source term for the equations, whose value is stored in the cell centroid.

B. Spatial Discretization

The discrete integral requires to evaluate the values at the faces of the cells. In Fluent, the value at the face centroid is used to represent the value for the entire face. The face values are interpolated from the cell centroid by choosing one of the spatial discretization schemes. Fluent provides a number of spatial discretization schemes. This section describes those that are relevant to this work.

In Fluent, the second-order upwind (SOU) scheme computes the face value based on the Taylor expansion of the value at the cell centroid. With second-order accuracy, the face value is computed as:

$$\phi_f = \phi_{c,up} + \overrightarrow{\nabla\phi}_{c,up} \cdot \vec{r}_{c,up} \qquad \textbf{(3.16)}$$

where ϕ is the solution variable, such as density, velocity components, or temperature (the pressure may be treated differently). $\phi_{c,up}$ and $\overrightarrow{\nabla\phi}_{c,up}$ are the value and the gradient at the upstream cell-centroid. $\vec{r}_{c,up}$ is the displacement vector from the upstream cell centroid to the face centroid. Evaluation of the gradient will be described shortly after.

Another second order accurate scheme is the central differencing (CD) scheme. The scheme computes the face value as:

$$\phi_f = \frac{1}{2}(\phi_0 + \phi_1) + \frac{1}{2}(\overrightarrow{\nabla\phi}_0 \cdot \vec{r_0} + \overrightarrow{\nabla\phi}_1 \cdot \vec{r_1}) \tag{3.17}$$

where subscript 0 and 1 denotes the two cells that boarder at the face. ϕ_0 and ϕ_1 are the cell centroid values, $\overrightarrow{\nabla\phi}_0$ and $\overrightarrow{\nabla\phi}_1$ are the gradient at the cell centroid, and $\vec{r_0}$and $\vec{r_1}$ are the displacement vectors from the cell centroids to the face centroid.

The CD scheme could be a good choice for the LES due to its low numerical diffusion. However it can also produce unphysical oscillations in the solution. To ease the problem, the bounded central differencing (BCD) schemeis recommended by Fluent as the default scheme for LES. The BCD scheme is based on the normalized variable diagram (NVD) approach (Leonard, 1991). It is a composite NVD-scheme which is a composition of the central differencing scheme and the second order upwind scheme. More detail of the scheme is given by Jasak et al. (1999).

The interpolation scheme of the pressure on the cell face may be treated differently from the above discussion. However, the second-order interpolation for the pressure chosen in this study is the same as the SOU interpolation discussed above.

C. Gradient Evaluation

For the spatial discretization, the gradient at the cell centroid may be required. In Fluent, the gradient can be computed by the Least-Squared Cell-Based (LSCB) method by solving the least-squared solution of the linear system of equations:

$$\begin{bmatrix} \left(\overrightarrow{\Delta r}_{c1}\right)^T \\ \vdots \\ \left(\overrightarrow{\Delta r}_{ci}\right)^T \\ \vdots \\ \left(\overrightarrow{\Delta r}_{cn}\right)^T \end{bmatrix} \cdot \left[\overrightarrow{\nabla \phi}_{c0}\right] = \begin{bmatrix} \phi_{c1} - \phi_{c0} \\ \vdots \\ \phi_{ci} - \phi_{c0} \\ \vdots \\ \phi_{cn} - \phi_{c0} \end{bmatrix} \tag{3.18}$$

where $\overrightarrow{\nabla \phi}_{c0} = \begin{bmatrix} \frac{\partial \phi}{\partial x_1} & \frac{\partial \phi}{\partial x_2} & \frac{\partial \phi}{\partial x_3} \end{bmatrix}^T$ is the gradient vector to be solved, $\overrightarrow{\Delta r}_{ci} = [x_{ci} - x_{c0} \quad y_{ci} - y_{c0} \quad z_{ci} - z_{c0}]^T$ is the displacement vector from the current cell centroid to the centroid of the i-th neighbouring cell. $(\phi_{ci} - \phi_{c0})$ is the difference between the cell-centroid values.

D. Temporal Discretization

The unsteady governing equations can be expressed as:

$$\frac{\partial \phi}{\partial t} = F(\phi) \tag{3.19}$$

where the time derivative term is singled out, and $F(\phi)$ is the rest of the terms to be discretized in space. In some steady-state simulation methods, the physical time t is replaced by the pseudo-time τ. Some temporal discretization schemes in Fluent are described below. Assumes the equation is discretized over a time step Δt.

The first order accurate time discretization scheme is expressed as:

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = F(\phi) \tag{3.20}$$

The second order accurate time discretization scheme is expressed as:

$$\frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\Delta t} = F(\phi) \tag{3.21}$$

In the above expressions, Δt is the time step and n denotes the current time level.

Each temporal discretization scheme can be explicit or implicit. In explicit schemes, $F(\phi)$ is evaluated using the known ϕ at the current and previous time levels. For implicit schemes, $F(\phi)$ is evaluated using the unknown ϕ at the next time level. For example, the first-order explicit scheme is expressed as:

$$\frac{\phi^{n+1}-\phi^{n}}{\Delta t}=F(\phi^{n}) \quad \textbf{(3.22)}$$

and the second-order implicit scheme is expressed as:

$$\frac{3\phi^{n+1}-4\phi^{n}+\phi^{n-1}}{2\Delta t}=F(\phi^{n+1}) \quad \textbf{(3.23)}$$

Thus, for the explicit scheme, marching over a time-step can be done efficiently as only one or a few loops over the cells (e.g. in the Rung-Kutta method) are required. For the implicit scheme, $F(\phi)$ also depends on the solution at the new time level, so the solutions of all the cells have to be solved together in a system of nonlinear equations, solved iteratively through a number of “sub-iterations”, in each of which a linearized system of equations is solved though a number of “inner iterations”. Therefore, for each time step, the computational cost is higher for the implicit scheme.

However, the time step for the explicit scheme is constrained by the stability limit of the solver, such as the Courant-Friedrichs-Lewy condition. The time step constraint can be very severe when very small cells are present in the computational domain, such as in the injection holes, on the boundary layers, etc. But for the implicit schemes such constraint does not exist or is much loosened. So it is possible that much fewer time steps can be used for the implicit scheme to cover the same physical time interval. Also, the implicit scheme provides the freedom to manually choose a fixed time step. After some comparisons in this study, it is shown that the implicit scheme is overall more efficient for both the unsteady and steady-state simulations. So the implicit scheme is preferred.

E. Pressure-Based and Density-Based Solvers

Fluent offers two types of solvers, the pressure-based solver and the density-base solver. The pressure based solver was historically developed for low Mach number and incompressible flows while the density-based solver was used for high speed incompressible flows. But in Fluent they are both capable to solve both types of flows.

In the pressure based solver, the continuity equation is replaced by the pressure correction equation, derived from the pressure-velocity coupling method. Several pressure-velocity coupling methods are available, such as the SIMPLEC method, the fractional step method, etc. For steady-state simulation, the SIMPLEC algorithm is found to provide fast and robust convergence, hence it is adopted in most standard steady-state simulations. For the implicit time discretization, usually tens of sub-iterations are required per time-step. This is undesirable for LES. Thus, Fluent offers the "non-iterative time advancement" (NITA) option (ANSYS Fluent Theory Guide), in which only one sub-iteration is required per time-step. Though the NITA will generate the additional "splitting error", it does not impair the overall time accuracy as the splitting error is on the same order of magnitude as the spatial discretization error, especially when a small time step is used. So the NITA option is used for the LES. Since the option is only available in the pressure-based solver, the LES is performed using the pressure-based solver, with the non-iterative fractional step method (Armfield and Street, 1999) pressure-velocity coupling.

The other type is the density-based solver. In Fluent, the explicit time or explicit pseudo-time discretization is available only in the density-based solver. As the efficient LES method developed later will require the explicit pseudo-time discretization. The density-based solver is chosen for that part.

3.2.3 RANS Flow Modelling Methods

In most engineering problems, the mean behaviour of the flow is interested. If the flow is turbulent but the unsteadiness is small, the flow is usually modelled by the steady-state Reynolds-Averaged Navier-Stokes (RANS) equations, which are much cheaper to solve than to temporally integrate the unsteady equations. The RANS equations can be produced by the time-averaging of the unsteady NS equations. Following the idea of Reynolds decomposition, an unsteady variable can be decomposed into the time-averaged part and the fluctuating part, i.e. $\phi = \bar{\phi} + \phi'$, where $\bar{\phi}$ is the time-average; or by Favre-averaging in the compressible flow, be decomposed as $\phi = \tilde{\phi} + \phi''$, where $\tilde{\phi} = \frac{\overline{\rho\phi}}{\bar{\rho}}$. Using the decomposition, the compressible RANS equations can be expressed as:

$$\frac{\partial \overline{\boldsymbol{F}}_i}{\partial x_i} - \frac{\partial \overline{\boldsymbol{F}}_{v,i}}{\partial x_i} = 0 \tag{3.24}$$

where:

$$\overline{\boldsymbol{F}}_i = \begin{bmatrix} \bar{\rho}\tilde{u}_i \\ \bar{\rho}\tilde{u}_1\tilde{u}_i + \bar{p}\delta_{1i} \\ \bar{\rho}\tilde{u}_2\tilde{u}_i + \bar{p}\delta_{2i} \\ \bar{\rho}\tilde{u}_3\tilde{u}_i + \bar{p}\delta_{3i} \\ \bar{\rho}\tilde{u}_i\widetilde{H} \end{bmatrix},$$

$$\overline{\boldsymbol{F}}_{v,i} = \begin{bmatrix} 0 \\ \bar{\tau}_{1i} - \overline{\rho u_1'' u_i''} \\ \bar{\tau}_{2i} - \overline{\rho u_2'' u_i''} \\ \bar{\tau}_{3i} - \overline{\rho u_3'' u_i''} \\ \tilde{u}_j\left(\bar{\tau}_{ji} - \overline{\rho u_j'' u_i''}\right) + \left(\frac{\bar{\mu} c_p}{Pr}\frac{\partial \tilde{T}}{\partial x_i} + \overline{\rho u_i'' h''}\right) + \left(-\frac{1}{2}\,\overline{\rho u_j'' u_j'' u_i''} + \overline{u_j'' \tau_{ji}}\right) \end{bmatrix} \tag{3.25}$$

in which, $Pr = \frac{\bar{\mu} c_p}{\kappa}$ is the Prandtl number. The material properties $\bar{\mu}$, c_p and κ can be considered constant or dependent on $\tilde{T}$. $\bar{\tau}_{ji}$ is given by $\bar{\tau}_{ji} = 2\bar{\mu}\tilde{s}_{ji} - \frac{2}{3}\bar{\mu}\tilde{s}_{kk}\delta_{ji}$, where $\tilde{s}_{ji} = \frac{1}{2}\left(\frac{\partial \tilde{u}_j}{\partial x_i} + \frac{\partial \tilde{u}_i}{\partial x_j}\right)$. For ideal gas, $\bar{p} = \bar{\rho}R\tilde{T}$. The Favre-averaged total enthalpy is $\widetilde{H} = c_p\tilde{T} + \frac{1}{2}\tilde{u}_k\tilde{u}_k + k$ and the Favre-averaged total energy is $\tilde{E} = \widetilde{H} - \frac{\bar{p}}{\bar{\rho}}$. $k = \frac{1}{2}\widetilde{u_k'' u_k''}$ is the turbulent kinetic energy (TKE). The quantities $\overline{\rho\nu'' \frac{\partial h''}{\partial x_i}}$ ($\nu = \frac{\mu}{\rho}$ is the kinematic viscosity) and $\overline{\rho\nu'' \left(s''_{ji} - \frac{1}{3} s''_{kk}\delta_{ji}\right)}$ can be neglected as they are often relatively small.

Due to the nonlinearity of the equations in the flux terms, the Reynolds stresses $\overline{\rho u''_j u''_i}$ in the momentum equations and the other turbulence correlation terms in the energy equation needs to be modelled to close the equations. In RANS, the closure is accomplished by the turbulence models.

In this study, the Spalart Allmaras and the k-ω SST turbulence models are chosen for RANS simulations. Details of these well-known models can be found from many sources, such as the original works of Spalart and Allmaras (1992) and Menter (1994).

3.2.4 LES Methods

In many highly turbulent flows situations such as those with flow separation, the RANS turbulence model may fail to model the large scale turbulence correctly. Therefore, the large-eddy simulation (LES) may be necessary to resolve the temporal and spatial dynamics of the turbulence. In LES, around 85% to 90% of the turbulence is resolved, with the smallest and universal sub-grid-scale (SGS) turbulence modelled, either by the SGS model, or simply by the numerical dissipation of the spatial discretization scheme. The theory of the LES is discussed in detail by Sagaut (2006) and Garnier et al. (2009). This section will briefly describe the basic theory and the LES methods applied in the solver of this study.

To decompose a variable into the resolved scale and the SGS scale, the resolved scale can be explicitly filtered out by a convolution operation:

$$\bar{\phi}(\boldsymbol{x}) = \int_D \phi(\boldsymbol{y}) G(\boldsymbol{x},\ \boldsymbol{y}; \Delta) d\boldsymbol{y} \qquad \textbf{(3.26)}$$

On the other hand, the filtering can also be accomplished implicitly by the computational mesh, as is done in Fluent. In this situation, the flow dynamics captured by the mesh resolution is the resolved scale. The scales smaller than the cell size which cannot be captured are considered the SGS scales. After the filtering operation, the resulting decomposition is expressed similar to the Reynolds decomposition: $\phi = \bar{\phi} + \phi'$ for incompressible flow and $\phi = \tilde{\phi} + \phi''$ for compressible flow with Favre filtering:

$$\tilde{\phi} = \frac{\overline{\rho\phi}}{\bar{\rho}} \tag{3.27}$$

Applying the filter operation to the governing equations and using the variable decomposition, the filtered Navier-Stokes equations for the compressible flow can be expressed as:

continuity:

$$\frac{\partial(\bar{\rho}\tilde{u}_i)}{\partial t} + \frac{\partial\left(\bar{\rho}\tilde{u}_j\right)}{\partial x_j} = 0 \tag{3.28}$$

momentum in i-th direction:

$$\frac{\partial(\bar{\rho}\tilde{u}_i)}{\partial t} + \frac{\partial\left(\bar{\rho}\tilde{u}_j\tilde{u}_i\right)}{\partial x_j} + \frac{\partial\bar{p}}{\partial x_i} - \frac{\partial\check{\tau}_{ji}}{\partial x_j} - \frac{\partial}{\partial x_j}\left(\bar{\tau}_{ji} - \check{\tau}_{ji}\right) = -\frac{\partial\sigma_{ji}}{\partial x_j} \tag{3.29}$$

energy:

$$\frac{\partial\left(\bar{\rho}\check{E}\right)}{\partial t} + \frac{\partial\left(\bar{\rho}\check{E} + \bar{p}\right)\tilde{u}_j}{\partial x_j} + \frac{\partial\check{q}_j}{\partial x_j} - \frac{\partial\left(\tilde{u}_i\check{\tau}_{ji}\right)}{\partial x_j} = -B_1 - B_2 - B_3 + B_4 + B_5 + B_6 - B_7 \tag{3.30}$$

in which,

$$\check{E} = c_p\tilde{T} - \frac{\bar{p}}{\bar{\rho}} + \frac{1}{2}\tilde{u}_k\tilde{u}_k = \frac{\bar{p}}{\bar{\rho}(\gamma - 1)} + \frac{1}{2}\tilde{u}_k\tilde{u}_k$$

$$\check{q} = -\kappa\frac{\partial\tilde{T}}{\partial x_j}$$

$$\check{\tau}_{ji} = 2\mu\tilde{s}_{ji} - \frac{2}{3}\mu\tilde{s}_{kk}\delta_{ji}$$

In the above equations, the terms on the right hand side are the SGS stress terms. The SGS terms in the momentum equation is:

$$\sigma_{ji} = \overline{\rho u_j u_i} - \bar{\rho}\tilde{u}_j\tilde{u}_i$$

In the energy equation, the SGS terms B_i are expressed as:

$$B_1 = \frac{1}{\gamma - 1}\frac{\partial}{\partial x_j}\left(\overline{pu_j} - \bar{p}\tilde{u}_j\right)$$

$$B_2 = \overline{p\frac{\partial u_k}{\partial x_k}} - \bar{p}\frac{\partial \tilde{u}_k}{\partial x_k}$$

$$B_3 = \frac{\partial}{\partial x_j}\left(\sigma_{kj}\tilde{u}_k\right)$$

$$B_4 = \sigma_{kj}\frac{\partial \tilde{u}_k}{\partial x_j}$$

$$B_5 = \overline{\sigma_{kj}\frac{\partial u_k}{\partial x_j}} - \overline{\sigma_{kj}}\frac{\partial \tilde{u}_k}{\partial x_j}$$

$$B_6 = \frac{\partial}{\partial x_j}\left(\bar{\tau}_{ji}\tilde{u}_i - \breve{\sigma}_{ji}\tilde{u}_i\right)$$

$$B_7 = \frac{\partial}{\partial x_j}\left(\bar{q}_j - \breve{q}_j\right)$$

Vreman et al. (1995) compared the relative magnitude of the terms and stated that the terms $\frac{\partial}{\partial x_j}\left(\bar{\tau}_{ji} - \breve{\tau}_{ji}\right)$, B_6 and B_7 can be neglected.

In this research, the SGS terms are modelled by the classical Smargorinsky-Lilly (SL) SGS model, developed by Smargorinsky (1963). It is selected due to its simplicity and low computation cost. Though there are some other models in Fluent, the SL model has shown to provide good results. The default values of the model constants are used, with $C_s = 0.1$ and the energy Prandtl number of 0.85.

In the LES of this work, the transient pressure based solver is used. The second-order implicit scheme is chosen for temporal discretization. The NITA option is chosen for faster time integration. The fractional step method is used for pressure-velocity coupling. In the spatial discretization, the SOU interpolation is chosen to compute the pressure at the cell face. The bounded central difference (BCD) scheme (Jasak et al., 1999) is chosen for the density interpolation and the discretization of the momentum and energy equations.

3.2.5 Boundary Conditions

This section will describe the key aspects of the various kinds of boundary conditions in Fluent that are used in this study

A. Velocity Inlet

The velocity inlet is suitable to model the injection in the flow control. At the velocity inlet, the velocity is specified, either by specifying the magnitude and direction, or the magnitude of each component, or the magnitude normal to the boundary surface, or profiles of velocity components obtained from other data. The static temperature should be specified at the boundary if the energy equation is solved. If the flow at the inlet is supersonic, the static pressure needs to be specified as well. For RANS, the inlet turbulence can be provided by turbulence intensity, turbulence viscosity ratio, turbulence length scale, depending on the turbulence model chosen. For LES, ‘no perturbation’ can be chosen to specify no turbulence.

B. Pressure Inlet

This type of inlet boundary condition specifies the total pressure, inflow direction, the total temperature (if energy equation is solved). If the incoming flow is supersonic, the static pressure also needs to be specified. For RANS, the inlet turbulence can be provided by turbulence intensity, turbulence viscosity ratio, turbulence length scale, depending on the turbulence model chosen. For LES, ‘no perturbation’ can be chosen to specify no turbulence.

C. Mass Flow Inlet

The mass flow inlet boundary condition is also suitable to model the injection in the flow control. This boundary condition specifies the total mass flow rate (or local mass flux), inflow direction, the total temperature (if energy equation is solved), the static pressure (for supersonic inflow). For RANS, the inlet turbulence can be provided by turbulence intensity, turbulence viscosity ratio, turbulence length scale, depending on the turbulence model chosen. For LES, ‘no perturbation’ can be chosen to specify no turbulence.

D. Pressure Outlet

At the pressure outlet boundary condition, the constant or average value of the static pressure is specified. If the average static pressure is set, the static pressure is allow to vary locally on the boundary but the average over the boundary remains fix. This is conducive to reduce the pressure wave reflection back into the computational domain. In addition, and importantly, the total mass flow rate out of the domain through the boundary can be specified. This makes it suitable to model the suction in the flow control. In this case, the static pressure will no longer be the prescribed value, but will be adjusted at every iteration to meet the prescribed mass flow rate. Based on the simple Bernoulli’s equation, the adjustment at every iteration is calculated in Fluent as:

$$\Delta p = \frac{0.5\rho_{ave}\left(\dot{m}^2 - \dot{m}_{req}^2\right)}{(\rho_{ave}A)^2} \tag{3.31}$$

where $\dot{m}$ is the total mass flow rate at the current iteration, $\dot{m}_{req}$ is the required total mass flow rate, ρ_{ave} is the average density at the boundary at the current iteration, A is the area of the boundary.

E. Pressure Far-Field

This boundary condition can also be used to simulate the free stream condition far away from the external flow devices. The static pressure, Mach number, direction of the flow and static temperature are specified. For RANS, the inlet turbulence can be provided by turbulence intensity, turbulence viscosity ratio, turbulence length scale, depending on the turbulence model chosen. For LES, ‘no perturbation’ can be chosen to specify no turbulence.

F. Wall Boundary

At the wall boundary, either the no-slip wall boundary condition or the slippery condition can be specified. For no-slip condition, the velocity at the boundary cell face is zero relative to the wall boundary. Shear stress for no-slip wall is calculated using the flow solution at the adjacent cells, via the velocity gradient for wall resolved boundary layer or via the wall functions if the boundary layer profile is modelled in some turbulence models. For slipper-wall, zero shear stress is specified at the wall. In terms of the thermal property, heat flux through the wall can be specified, and the temperature is calculated accordingly. With zero heat flux specified, the wall becomes adiabatic. Alternatively, constant temperature can be specified at the wall, the heat flux through the wall is then calculated. Alternatively, the heat transfer coefficient and the freestream temperature can be specified, and the temperature and heat flux are calculated accordingly.

G. Periodic Boundary

The periodic boundary condition is formed with a pair of surfaces. The periodicity can be set as rotational or translational. In translational periodicity used in this work, the spatial periodicity is recognized by the relative displacement of the two periodic surfaces in the pair. In the flow solution, the mesh cells adjacent to one periodic surface of a periodic pair are treated as the direct neighbours of the cells adjacent to the other surface in the periodic pair. No parameter input is required at the boundary.

H. Symmetry Boundary

At the symmetry boundary, the flow field and geometry is treated as mirror about the boundary. Zero flux of all the quantities is assumed, leading to zero normal velocity and zero normal gradient of the flow quantities on the boundary surface. The symmetric boundary can also be used to simulate the slippery wall, since the shear stress at the boundary is zero.

3.2.6 Mesh Generation

The mesh generator in this research is the commercial software ANSYS ICEM CFD. Both structured and unstructured meshes can be generated. The detail of the mesh generation process is dependent on each case and will be discussed along with the cases in later chapters.

ICEM CFD also provides scripting for the automatic mesh generation. This feature is very useful for automizing the optimization cycle. The automatic meshing reduces human labour, and can generate more consistent meshes for different designs in the optimization.

3.3 Summary

This chapter describes the baseline numerical methods for the optimization and flow simulation. To investigate the combined optimization and the sequencing of the disciplines, five optimization approaches for the shaping and flow control optimization are identified. They are: 1) shaping only; 2) flow control optimization only; 3) shaping, then flow control optimization; 4) flow control optimization, then shaping; 5) concurrent optimization of the shape and the flow control. An automatic optimization system is developed and implemented in Matlab to perform the optimization consistently for each approach. The system is modular whose parts are compatible with different methods. To efficiently find the maximum performance improvement by each optimization approach, the Kriging-surrogated model is used to explore the design space for the global optimum.

The flow governing equations is discretized by the cell-based finite volume method in the ANSYS Fluent flow solver package. The principles and methods in the RANS modelling and the LES modelling are described. Fluent provides both pressure-based and density-based solvers. The LES is carried out in the pressure-based solver to use the NITA option for less expensive time-integration. The computational mesh is generated using the ANSY ICEM CFD and can be made automatic by scripting.

Chapter 4

2D Cases in Shaping and Flow Control Optimization

This chapter presents a computational investigation in relation to the first objective of this research. The cases studies using a RANS model on 2D configurations are regarded as a valid starting point to address the fundamental questions and issues regarding the relation between shaping and flow control. The outcome should also serve as a basis for motivating further more realistic 3D case studies involving wider flow phenomena and higher-fidelity but more expensive flow modelling. Two questions are investigated:

First, how much further performance improvement can be obtained when combining the shaping and flow control optimization, as compared to the conventional shaping alone and flow control optimization alone?

Second, when combining the disciplines, what would be the difference in the optimized performance, if the shaping and flow control optimization are conducted at different sequences?

Two configurations are considered here to cover a sufficiently wide range of interest and applications. An internal flow case and an external flow case are chosen for the investigation. In the internal flow case, the shape and flow control of a compressor blade is optimized to reduce the aerodynamic loss and to increase the operating range without blade stall. In the external flow case, the circulation control airfoil is investigated for the lift maximization.

4.1 Compressor Blade Optimization

4.1.1 Case Description

In general, the objectives in the compressor blade design will include the minimization of the flow loss at the design inlet flow angle and the maximization the operating range. In this case, the potential of combining shaping and flow control optimization in improving the performances can be examined by showing how these two objectives can be better satisfied.

The baseline blade to be optimized is a controlled diffusion blade profile originally designed by Sanger (1983) and experimentally tested by Elazar (1988). The blade coordinate system used in this research is shown in Figure 4-1. The abscissa ξ is in line with the chord, and the origin is placed at the leading edge. The geometry and the boundary conditions all comes from the experiment, and are listed in Table 4-1 and Table 4-2.

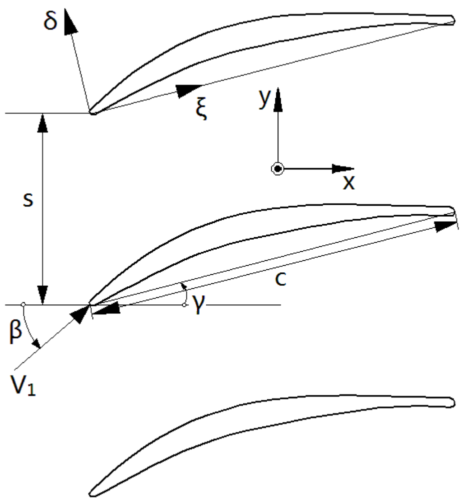


Figure 4-1 Baseline blade geometry and coordinate systems

Table 4-1 Blade geometric parameters

Parameter	Value
Chord length c	126.7 mm
Cascade solidity c/s	1.66
Stager angle γ	14.67˚
Design inlet flow angle β_d	40°

Table 4-2 Computational boundary conditions

Parameter	value
Inlet Mach number Ma_1	0.245
Inlet Reynolds number Re_1	6.75×10^5
Inlet turbulence intensity	1.5%
Inlet turbulence length scale	2.54 cm
Outlet static pressure p_2	1 atm

For the shape optimization, the shape of the blade is changed by adding a perturbation function (Figure 4-2) onto the baseline distribution of camber or thickness $\delta_0(\xi)$, as shown in Figure 4-2 and Figure 4-3 and expressed as:

$$\delta(\xi) = \delta_0(\xi) + f(\xi) \tag{4.1}$$

The perturbation function $f(\xi)$ is formed by the non-uniform rational B-spline (NURBS, see Rogers, 2000) with static and active control points as:

$$[\xi(t),\ f(\xi(t))] = \sum_{i=1}^{n_p} b_i(t)\boldsymbol{P}_i \tag{4.2}$$

where t is the parameter of the curve. $b_i(t)$ $(i = 1,2,\ldots,n_p)$ are the rational basis functions and $\boldsymbol{P}_i = [\xi_i,\ \delta_i]$ are the coordinates of the control point. The coordinates of the static control points are fixed, while the active control points are fixed in ξ but their δ coordinates are determined by the shape design variables in the following manner:

$$\boldsymbol{P}_{cam} = \begin{bmatrix} 0 & 0 \\ 0.15 & 0 \\ 0.33 & d_{cam1} \\ 0.67 & d_{cam2} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} c \\ \max(\delta_{0,cam}) \end{bmatrix} \tag{4.3}$$

$$\boldsymbol{P}_{thick} = \begin{bmatrix} 0 & d_{thick1} \\ 0.15 & d_{thick2} \\ 0.33 & d_{thick3} \\ 0.67 & d_{thick4} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} c \\ \max(\delta_{0,thick}) \end{bmatrix} \tag{4.4}$$

in which $\boldsymbol{P} = [P_1,\ P_2,\ \ldots,\ P_i,\ \ldots]^T$ is a matrix containing the control point coordinates, and d_{cam} and d_{thick} are the shape design variables for the camber and thickness. The total number of shape design variables is six. The constraints for the camber and thickness perturbations are imposed by setting $d_{cam} \in [-0.2,\ 0.5]$ and $d_{thick} \in [-0.5,\ 1.5]$.

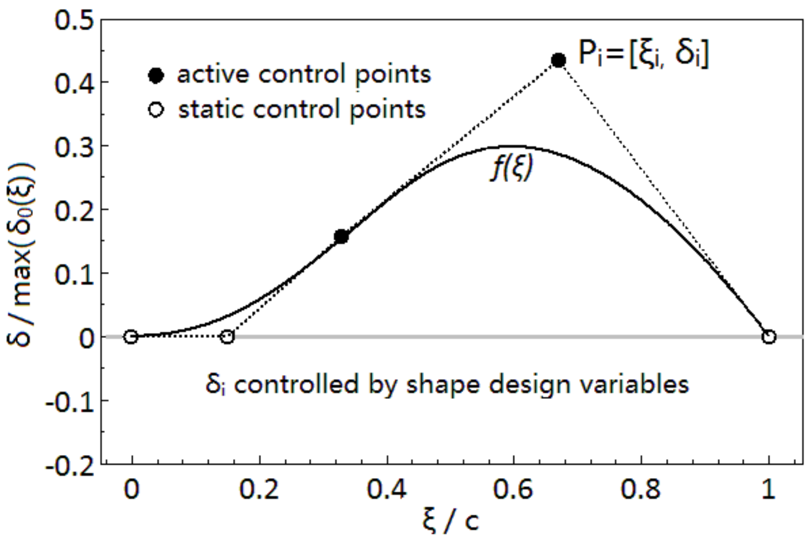


Figure 4-2 Shape perturbation function $f(\xi)$

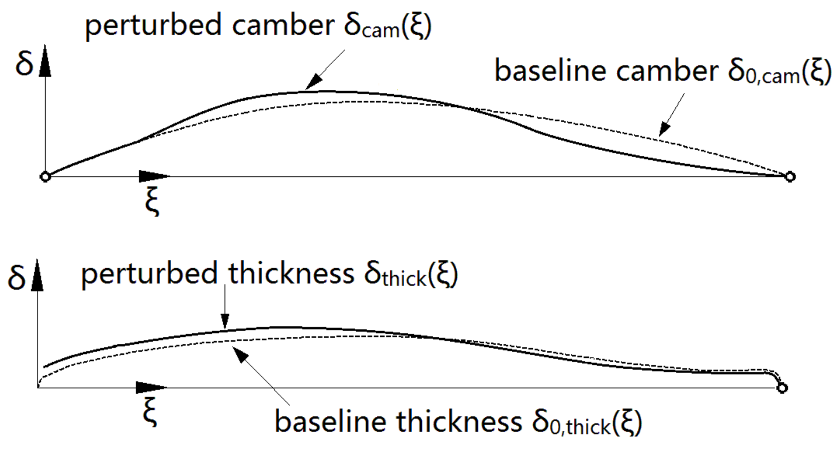


Figure 4-3 Deformation of the camber and the thickness distribution

For the flow control, four design variables are assigned, as shown in Figure 4-4 below. They are the massflow rate $\dot{m}_{jet}$ that is assumed to be the same between the injection and the suction, varying between 0.005 to 0.05 kg/s (i.e. about 0.1% to 1% of the inlet massflow rate at the design inlet flow angle β_d based on the nominal $1\ m$ span); the injection angle is θ_{jet} away from the local blade surface, varying between 20° and 60°; and the positions of the blowing and suction slots ξ_{jet1} and ξ_{jet2} in the chordwise direction. In the flow solution, the blowing is modelled with mass flow inlet boundary condition, and the suction is modelled with pressure-outlet boundary condition with target mass flow rate specified, as explained in Chapter 3.

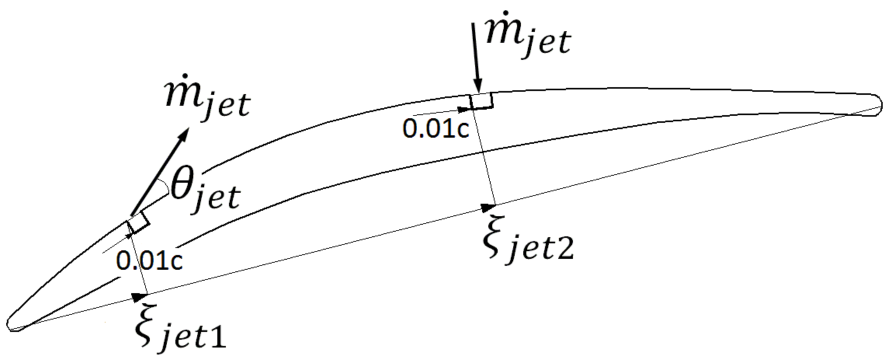


Figure 4-4 Flow control design variables on the blade

For simplicity, some assumptions are made on the flow control without significantly limiting the general applicability of the present findings. First, the blowing slot is always located in front of the suction slot. The widths of the slots are fixed at $0.01c$. Second, the suction is simply modelled as normal to the surface. Third, the flow control configuration can vary as the inlet flow angle β changes. Fourth, the injected air is assumed to solely come from the suction. The suction air is assumed to be recirculated through a piping system (modelled but not solved) inside the blade. As such, additional energy is required to pup the aspirated air to the injection slot and reach at enough total pressure there. The specific work to power the flow control can be roughly estimated as:

$$w_{jet} = \frac{\Delta p_{t,jet} + p_{t,jet1} + p_{t,jet2}}{\rho_{jet}} \tag{4.5}$$

in which ρ_{jet} is the mean flow densities between the blowing and suction slots. $p_{t,jet1}$ and $p_{t,jet2}$ are the total pressure at the blowing and suction slots. The 2D piping system is imagined to be a straight pipe $1\,mm$ in height with nominal span of $1\,m$. The total pressure decrease across the piping system $\Delta p_{t,jet}$ including the entrance effect is interpolated based on the Moody chart (see White, 2003, p. 349). The above flow control power consumption is an approximate estimation and may vary if more detail modelling of the flow control is available. However, it is still taken as a reasonable estimation for the purpose of the present investigation.

The aerodynamic loss across the blade passage is evaluated by the total pressure loss coefficient evaluated at the blade passage outlet, expressed as:

$$\zeta = \frac{p_{t1} - p_{t2}}{p_{t1} - p_1} \tag{4.6}$$

where p_{t1} and p_{t2} are the total pressure at the passage inlet and outlet, p_1 is the static pressure at the blade passage inlet.

When the flow control is applied, the energy to power the flow control is integrated into the loss calculation as the price for the reduction of the aerodynamic loss. The resulting equivalent loss can then be calculated after calculating the overall aerodynamic efficiency of the blade passage that includes the effect of the flow control. As used by Gmelin et al. (2012), the aerodynamic efficiency can be considered as the ratio of the benefit to the cost. In a compressor cascade, the benefit is made up of the specific flow kinetic energy at the passage outlet $\frac{U_2^2}{2}$ (U be the averaged velocity magnitude on the boundary), and the specific technical work done to the fluid by the blade passage $\frac{(p_2-p_1)}{\rho_1}$. The cost is made up of the specific kinetic energy at the passage inlet $\frac{U_1^2}{2}$ and the specific energy cost for the flow control, w_{jet} as defined earlier in Equation (4.5). Finally, all the items are massflow-weighted to produce the aerodynamic efficiency as:

$$\eta = \frac{0.5\dot{m}_1 U_2^2 + (p_2 - p_1)\dfrac{\dot{m}_1}{\rho_1}}{0.5\dot{m}_1 U_1^2 + w_{jet}\cdot\dot{m}_{jet}} = \frac{[(p_{t2} - p_2) + (p_2 - p_1)]\dfrac{\dot{m}_1}{\rho_1}}{(p_{t1} - p_1)\dfrac{\dot{m}_1}{\rho_1} + w_{jet}\cdot\dot{m}_{jet}} \tag{4.7}$$

in which ρ_1 is constant for the incompressible flow. $\dot{m}_1$ denotes the mass flow rate through both the inlet and the outlet, as $\dot{m}_{jet}$ is assumed to be the same between the blowing and suction. Then, the equivalent loss can be calculated as:

$$\zeta_{eq} = 1 - \eta = \frac{p_{t1} - p_{t2} + w_{jet}\cdot\dot{m}_{jet}\dfrac{\rho_1}{\dot{m}_1}}{p_{t1} - p_1 + w_{jet}\cdot\dot{m}_{jet}\dfrac{\rho_1}{\dot{m}_1}} \tag{4.8}$$

When the flow control is deactivated ($\dot{m}_{jet} = 0$), the equivalent loss reduces to the aerodynamic loss defined in Equation (4.6). Thus the equivalent loss is used as the unified loss quantification for both the controlled and un-controlled blade flow.

4.1.2 Flow Solver Validation and Numerical Sensitivity Study

The mesh of the blade is shown in Figure 4-5. The mesh nodes are uniformly distributed across the blade surface and the flow control slots. Albeit at some increase in the computational cost, the same mesh density between the blade surface and the slot has shown to be necessary to ensure consistent results between the controlled and the un-controlled scenarios. Results of the mesh dependence study are shown in Table 4-3. It shows the original mesh is sufficient, with 5 mesh elements for each slot and 700 mesh elements covering each side of the blade. The wall boundary layer is resolved without wall function. The $y+$ on the blade surface is made on the order of 1. The mesh expansion ratio normal to surface is about 1.07. The k-ω SST turbulence model is used for the RANS.

Table 4-3 Mesh dependency study results

Item	Un-controlled at $\beta = 40°$			Controlled at $\beta = 47°$		
Mesh density in one direction	original	$\times$ 1.5	$\times$ 2	original	$\times$ 1.5	$\times$ 2
Mesh element on a blade pressure surface	700	1000	1500	700	1000	1500
Mesh elements on each slot	---	---	---	5	7	10
Computed aerodynamic loss ζ	0.0252	0.0253	0.0252	0.0420	0.0413	0.0411

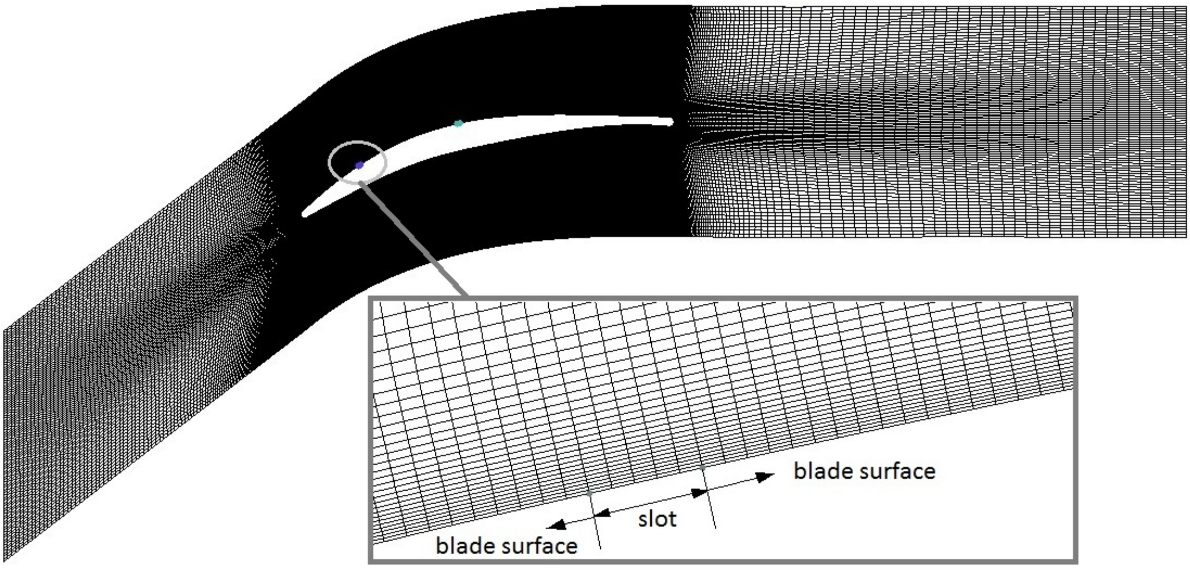


Figure 4-5 Blade mesh in the flow control case

For the baseline blade, the variation of the loss with the inlet flow angle is shown in Figure 4-6 and the surface pressure distribution at the design inlet flow angle is shown in Figure 4-7. It shows that the predictions are in good agreement with the experiment of Elazar (1988).

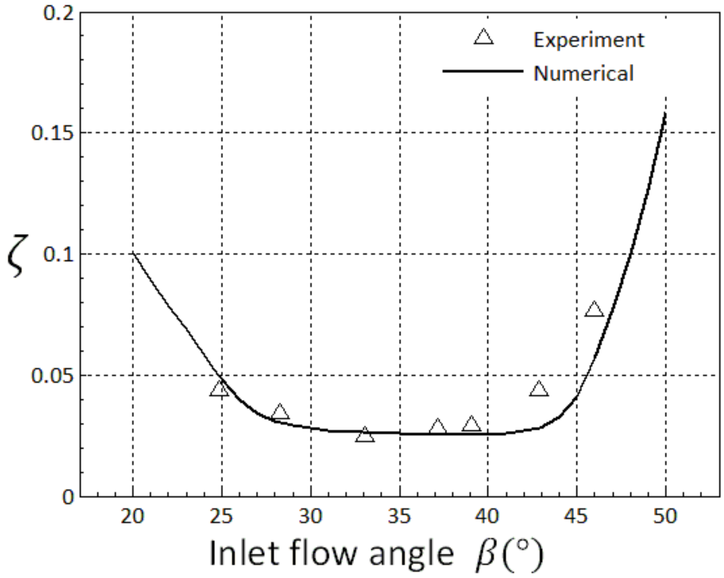


Figure 4-6 Loss variation with the inlet flow angle for the baseline blade

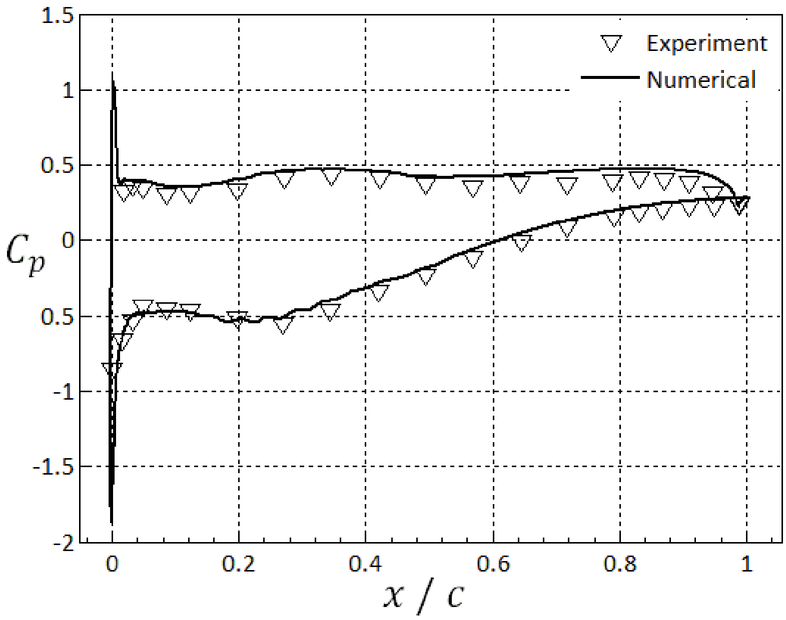


Figure 4-7 Pressure distribution for the baseline blade at the design inlet flow angle ($\boldsymbol{\beta} = 40°$)

4.1.3 Optimization Results and Discussion

For the blade design, the first objective is to minimize the design loss (i.e. the loss at the design inlet flow angle β_d). The second objective is to maximize the operating range. In this paper, the operating range is defined as the maximum inlet flow angle β_{max} within the constraint of $\zeta_{eq} \leq 0.1$.

Based on the two objectives, three optimization options can be considered: option I, minimizing the design loss only, without considering the operating range; option II, maximizing the operating range only, without considering the design loss; option III, the "multipoint" option, that is minimizing the design loss as well as maximizing the operating range. Ideally, the third option is desired. But the first and second options are still carried out to see how much each objective can be improved.

A. Shaping Only

When only shaping is considered, to only minimize the design loss, the optimization can be formulated without constraint functions:

Minimize:

$$\zeta_{eq} = I(\boldsymbol{d}_{shape}, \ \beta = \beta_d) \tag{4.9}$$

in which $\beta_d \equiv 40°$ denotes the design condition.

To maximize the range, one can minimize the loss at a selected high inlet flow angle near the upper boundary of the baseline operating range. This method is efficient as the flow simulation only needs to be performed once at the selected high β per blade design, as has been used in previous works (e.g. Sonoda et al., 2004). Therefore, to maximize the range only, the optimization can be formulated without constraint functions as:

Minimize:

$$\zeta_{eq} = I(\boldsymbol{d}_{shape}, \ \beta = \beta_{max,0}) \tag{4.10}$$

in which $\beta_{max,0} = 48°$ is the maximum inlet flow angle in the operating range of the baseline blade.

In the "multipoint" option, the operating range is maximized first by option two (β_{max} thus increased from $\beta_{max,0}$ to $\beta_{max,maxed}$), then hold the $\beta_{max,maxed}$ and minimize the design loss. Thus the optimization problem can be formulated with a constraint function:

Minimize:

$$\zeta_{eq} = I(\boldsymbol{d}_{shape},\ \beta = \beta_d) \tag{4.11}$$

Subject to:

$$\zeta_{eq}(\boldsymbol{d}_{shape},\ \beta = \beta_{max,maxed}) \leq 0.1 \tag{4.12}$$

If without this constraint, the already maximized range cannot be preserved, and the optimization result will become the result of "min design loss only".

Performances of the blade from the three shaping options are summarized in Table 4-4. The loss variations as well as the corresponding blade shapes are shown in Figure 4-8. For minimizing the design loss only, the loss at β_d is reduced by 4.4% from the baseline blade. However, the operating range also shrinks slightly. Figure 4-9 shows the outlet wake becomes slightly narrower and shallower at β_d, but wider and deeper at $\beta = 47°$.

For maximizing the range only, the β_{max} is increased by 9.3° from the baseline value. However, the design loss also increases noticeably from 0.0252 to 0.0431. As can be seen in Figure 4-10 at the design inlet flow angle, due to a much thicker shape and a higher camber, the flow separated on the suction surface of the blade, producing a large wake, resulting in the increase of the loss at β_d. At a large inlet flow angle β=47˚, the large leading edge radius reduces the flow diffusion and separation around the leading edge, thus increasing the operating range as defined.

For the “multipoint” option, it can be seen that though the maximized range is maintained, shaping is unable to reduce the design loss to be lower than that of the baseline blade.

From the above results, it appears that by shaping only, it is difficult (if not impossible) to increase the range and reduce the design loss at the same time. To achieve both objectives, as indicated by the “desired” loss curve in Figure 4-8, alternative optimization approaches must be considered.

Table 4-4 Blade performance from shaping only

Blade	Performance	
	Loss at β_d	β_{max} (°)
Baseline	0.0252	48
Min design loss	0.0241	47.6
Max range	0.0431	57.3
Multipoint	0.0363	57.5

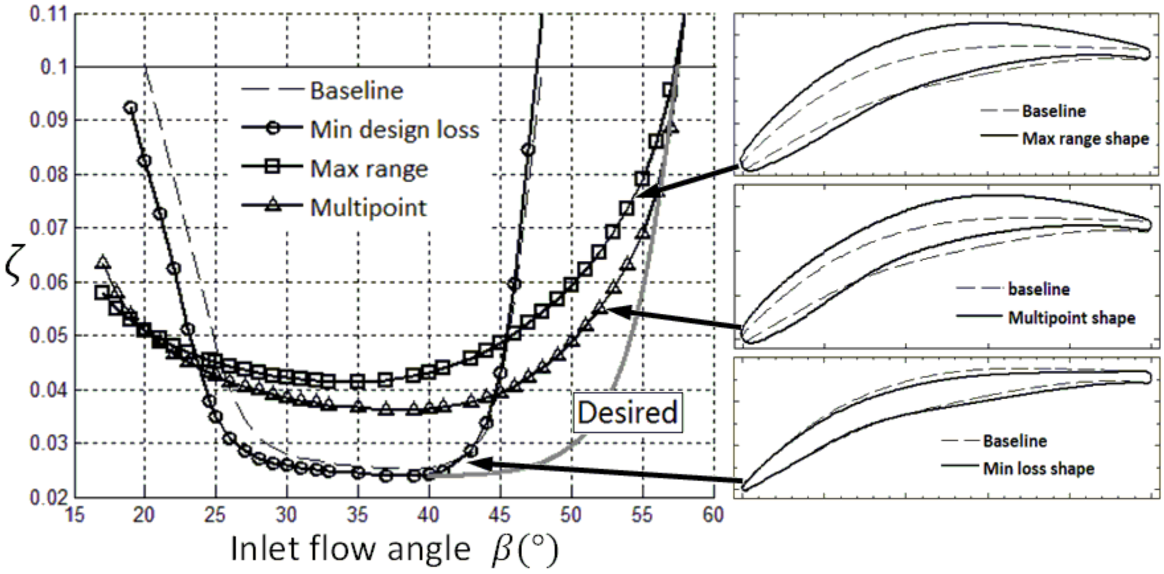
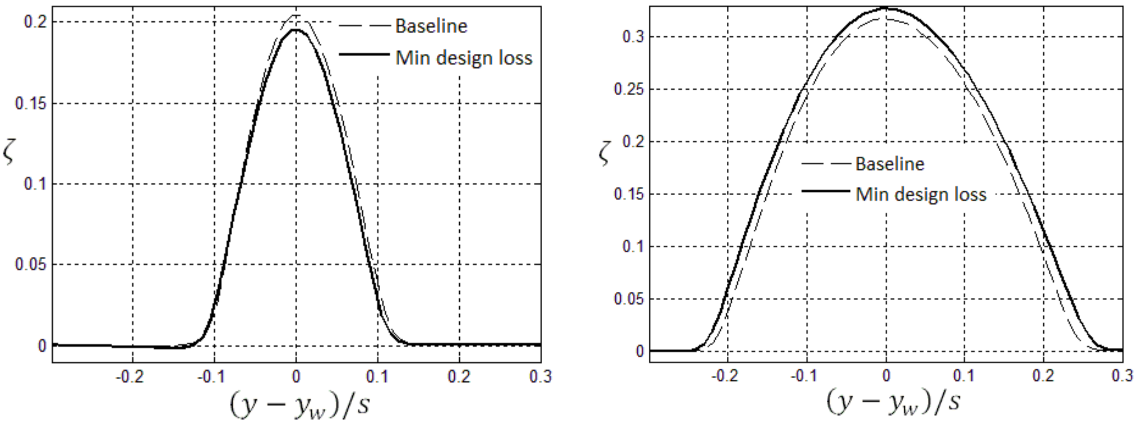
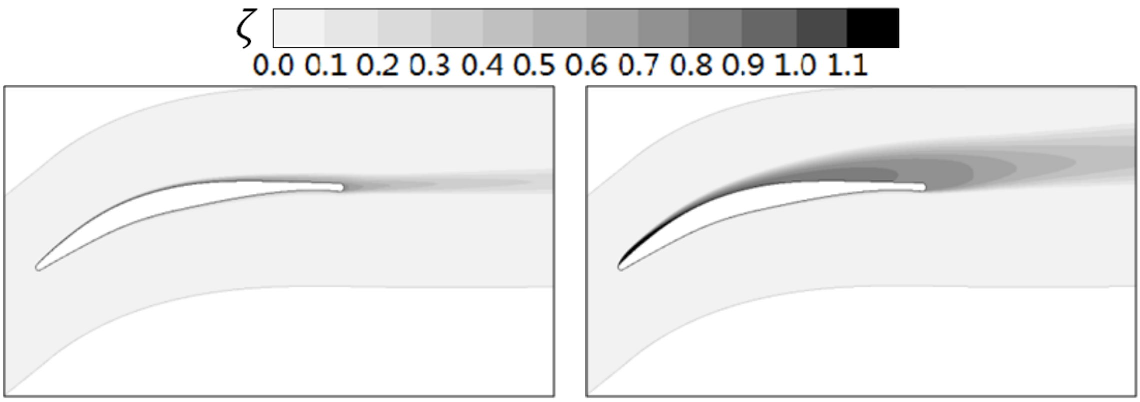


Figure 4-8 Loss curve of the optimized blade by shaping only, as well as the “desired” loss curve

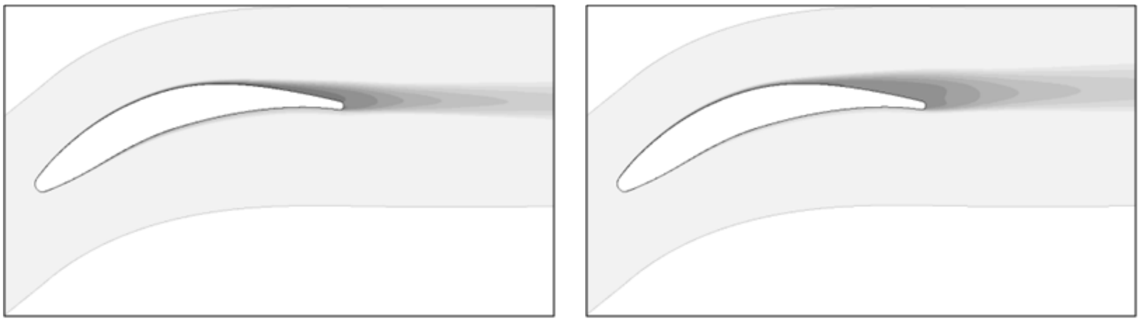


(a) $\boldsymbol{\beta = \beta_d}$ (b) $\boldsymbol{\beta = 47°}$

Figure 4-9 Wake profiles (with wake traverse center y_w) at the outlet plane; (a) $\boldsymbol{\beta = 40°}$, (b) $\boldsymbol{\beta = 47°}$



(a) Baseline, $\boldsymbol{\beta = \beta_d \equiv 40°}$ (b) Baseline, $\boldsymbol{\beta = 47°}$



(c) Maxed range, $\boldsymbol{\beta = \beta_d \equiv 40°}$ Maxed range, $\boldsymbol{\beta = 47°}$

Figure 4-10 Comparison of the total pressure loss (ζ) in the flow field between "max range" blade and the baseline blade at $\boldsymbol{\beta = 40°}$ and $\boldsymbol{\beta = 47°}$

B. Combined Optimization: Shaping → Flow Control

In light of the limitation of shaping only, it is considered to combine shaping and flow control optimization to achieve both objectives. A possible way to do so is first to minimize the design loss by shaping, and then maximize the range by flow control optimization. To this end, minimizing the design loss is the same as the first shaping option, while maximizing the range by flow control optimization is formulated as:

Minimize:

$$\zeta_{eq} = I(\boldsymbol{d}_{FC},\ \beta = \beta_d, \beta_d + 1, \ldots) \quad \textbf{(4.13)}$$

That is, the flow control is optimized to minimize the loss at every β respectively. The inlet flow angle is increased from β_d until the loss is unable to be reduced below $\zeta_{eq} \leq 0.1$. Thus a series of optimal flow control configurations are obtained for different β, representing the adaptation of the flow control to the inlet flow angle.

The resulting loss distribution is shown in Figure 4-11, and compared with other loss curves from the pure shape optimizations. The optimized flow control parameters for each inlet flow angle are listed in Table 4-5. In the table, the equivalent loss ζ_{eq} versus the inlet flow angle β correspond to the loss curve tagged “Min loss shape + FC” in Figure 4-11. It can be seen that this loss distribution is very close to the desired one indicated in Figure 4-8. The design loss is reduced from 0.0253 to 0.0231. Also, the range is significantly increased. In Figure 4-12, the flow fields in two situations where the flow control is turned off and on respectively are compared at $\beta = 47°$. As shown, the flow separation is effectively suppressed by the flow control, leading to a large decrease in the loss at high inlet flow angle conditions thus an increase in the range. The flow control keeps the blade still operating at an optimal state at these off-design conditions, where the blade shape is already far form optimal.

As can be seen from the discussion above, when shaping and flow control optimization are combined, both objectives in improving the blade's performance are achieved, whereas for shaping only, only one of the two objectives can be satisfied.

Table 4-5 Optimized flow control design variables from "shaping → flow control optimization"

$\beta(°)$	$\frac{\xi_{jet1}}{c}$	$\frac{\xi_{jet2}}{c}$	$\frac{\dot{m}_{jet}}{\dot{m}_1}$	$\theta_{jet}(°)$	ζ_{eq}
40	0.002	0.07	0.19%	20	0.0231
42	0.13	0.41	0.58%	20	0.0231
44	0.002	0.07	0.50%	20	0.0231
45	0.002	0.07	0.57%	20	0.0241
46	0.002	0.07	0.62%	20	0.0253
47	0.002	0.07	0.71%	20	0.0269
48	0.002	0.07	0.78%	20	0.0282
49	0.002	0.07	0.86%	20	0.0300
50	0.002	0.07	1.04%	20	0.0325
51	0.002	0.07	1.07%	20	0.0358
52	0.002	0.07	1.09%	20	0.0420
53	0.002	0.07	1.12%	20	0.0575
54	0.002	0.07	1.14%	20	0.0840

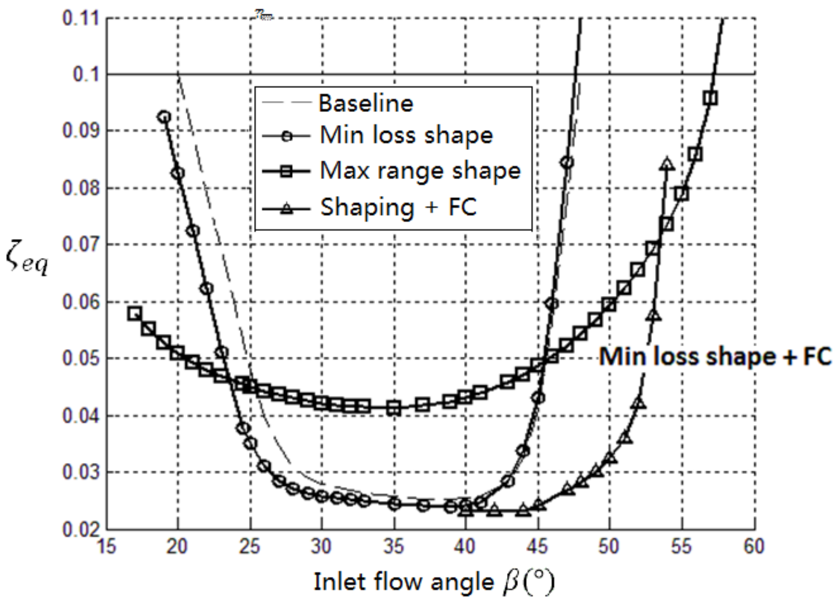


Figure 4-11 Loss curve of the "shaping only" and the "shaping → flow control optimization"

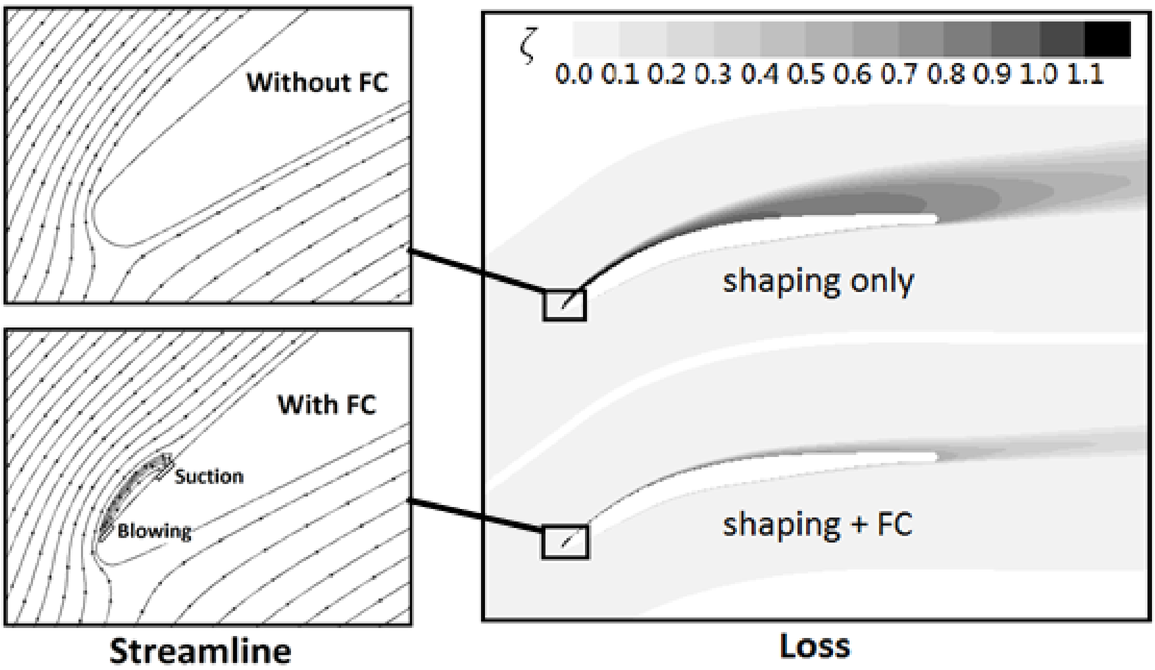


Figure 4-12 Streamlines around the leading edge and the total pressure loss in the flow field at $\beta = 47°$ from different optimization approaches: shaping only and shaping → flow control optimization

C. Computational Cost

The computational cost for the present case is dominated by the flow solutions which are executed in parallel with ten 2.4 GHz CPU cores. Either with or without flow control, each flow analysis takes about 12 minutes to converge. Each shaping option takes 24 hours (or 48 hour for the “multi-point” option) to complete with 60 design-of-experiment samples and around 30 optimization cycles. For the flow control optimization at each inlet flow angle, it takes 17 hours with 48 design-of-experiment samples and around 40 optimization cycles. In total, it takes about 220 hours to optimize the flow control at 13 inlet flow angle conditions to produce the blade performance characteristics given by the “Shaping + FC” loss curve in Figure 4-11. Overall, the computational cost is not huge.

4.2 Circulation Control Airfoil Optimization

4.2.1 Case Description

In this case, the optimization methodology is applied to maximize the lift of a circulation control airfoil. Not only will the potential of combining shaping and flow control optimization be demonstrated, but also the sequencing of the disciplines in the combined optimizations will be investigated.

The principle of the circulation control is that a jet is blown above the blunt trailing edge of an airfoil. Due to the Coanda effect, the jet flows and bends along the blunt trailing edge for a distance and then separates from the surface. As a result, the trailing edge separation point moves below the airfoil, leading to a large increase of the circulation around the airfoil, and hence a large increase in the lift. The effect of the circulation control is sensitive to both the trailing edge shape and the flow control parameters. In this study, the NACA GAW-1 airfoil is chosen as the baseline airfoil. Its aerodynamic data can be found in the experiment of McGhee and Beasley (1973). This airfoil has been used in previous circulation control research, such as Jones et al. (2002). In this case, the freestream condition is set to within the typical flight conditions. The freestream Mach number is set to 0.2. The freestream Reynolds number based on the baseline airfoil chord length is 1,260,000. The flow comes at the angle of attack of 2°. The k-ω SST is used as the turbulence model. The inlet turbulence is set to some reasonable quantities in the absence of available experimental data. The freestream turbulence intensity is set to 1% and the turbulence viscosity ratio is 10. For the injection the turbulence intensity is set as 10% and turbulence viscosity set as 50.

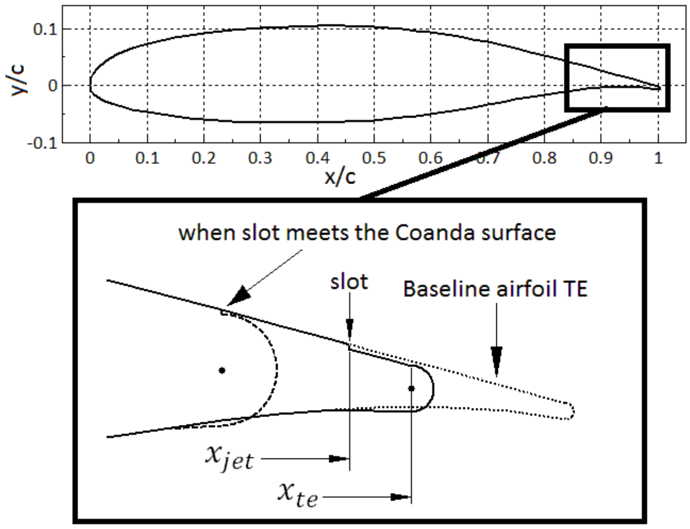


Figure 4-13 Application of the circulation control on the airfoil trailing edge, and the two governing parameters x_{jet} and x_{te}

To apply the circulation control, part of the baseline trailing edge is trimmed blunt to become the Coanda surface, as shown in Figure 4-13 above. The Coanda surface is an 180° semi-circle, whose radius depends on its starting position x_{te}. As the starting point moves backward, the diameter of the Coanda surface approaches the thickness of the baseline trailing edge. The position of the Coanda trailing edge x_{te} varies between $0.85c_0$ and c_0, where c_0 is the chord length of the baseline airfoil. Definition of the shape design variable d_{shape} is embedded in the following equation:

$$x_{te} = d_{shape}(c_0 - 0.85c_0) + 0.85c_0\, , \, d_{shape} \in [0, 1] \quad \textbf{(4.14)}$$

The jet is blown tangentially to the local airfoil surface at the slot exit. The width of the slot is fixed at $0.002c_0$ and the blowing massflow rate is fixed at $\dot{m}_{jet} = 0.9\ kg/(m \cdot s)$, corresponding to the momentum coefficient $C_\mu = \frac{\dot{m}_{jet} U_{jet}}{0.5 \rho U_\infty^2 c_0}$ of about 0.014, which amid of the range tested in the experiment. Due to the geometric constraint that the jet slot is not allowed to be on the Coanda surface, its chordwise position x_{jet} varies between $0.8c_0$ and x_{te}. The FC design variable d_{FC} is then defined by:

$$x_{jet} = d_{FC}(x_{te} - 0.8c_0) + 0.8c_0\ ,\ d_{FC} \in [0, 1] \quad \textbf{(4.15)}$$

Only two design variables are considered, one for the shape and the other for the flow control. By using no more than two design variables, the design space and the objective function can be easily visualized, which makes it convenient to examine the features of different sequencing of the disciplines.

The optimization problem is then formulated as:

Maximize:

$$I = C_L = \frac{L}{0.5\rho_\infty U_\infty^2 c_{trim}} \quad \textbf{(4.16)}$$

Subject to:

$$g = \frac{C_L}{C_D} - 60 > 0 \quad \textbf{(4.17)}$$

in which c_{trim} is the trimmed chord. The lift and drag are calculated by integrating the pressure and frictional forces on the airfoil surface.

Since there are no more than two design variables, the simplified optimization process is adopted.

4.2.2 Flow Solver Validation and Numerical Sensitivity Study

The mesh for the circulation control airfoil is shown in Figure 4-14. It takes on the two-block structure, with the first block wraps around the airfoil from the slot to its upper lip, and the second O-grid type block placed outside the first block. The far-field boundary is placed 10 baseline chord away from the airfoil. The result of the mesh dependence study is shown in Table 4-6. It shows that the difference of the lift coefficient is very small for both the situations with and without the flow control. Thus, the original mesh densities are adopted. For flow control simulations, there are 55, 70 and 100 mesh elements covering the jet slot, the upper and lower surfaces and the Coanda surface. Based on Min et al. (2009), influence of the detailed modelling of the blowing chamber inside the airfoil is small and considered as unnecessary for RANS. The $y+$ on the airfoil surface is on the order of 1, as the wall boundary layer is to be solved directly instead of using wall functions.

Table 4-6 Mesh dependency study results

Item	No flow control		Circulation control	
Mesh density in one direction	original	× 2	original	× 2
Number of elements around the airfoil	214	437	---	---
Number of cells on the O-grid	---	---	335×55	675×110
Computed C_L	0.73	0.71	0.86	0.86

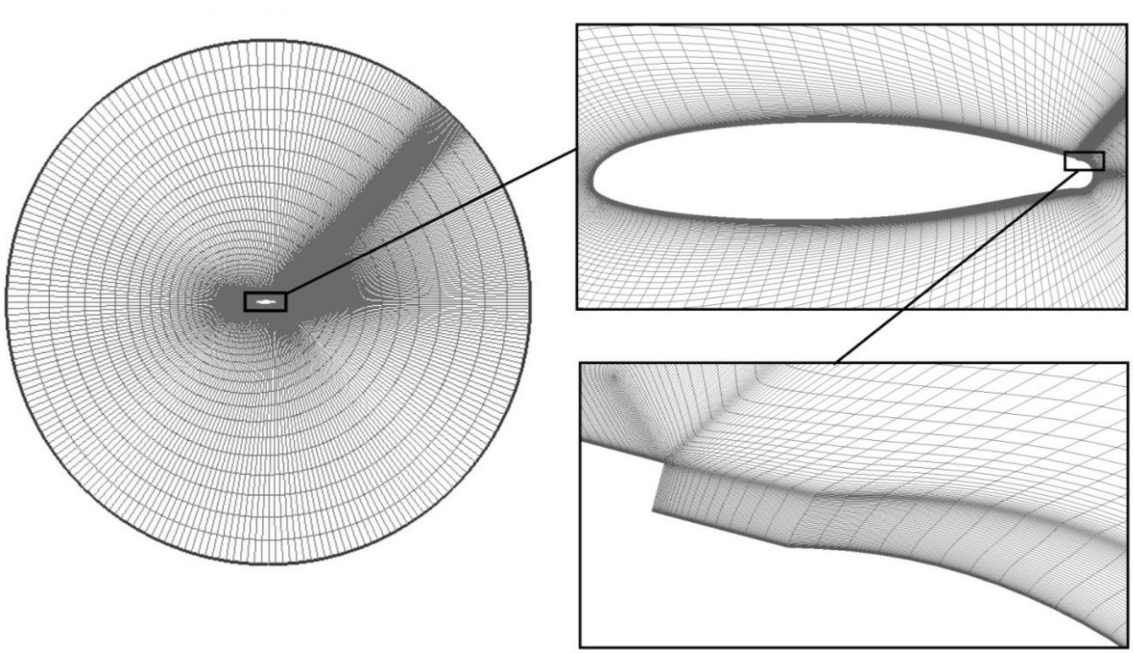


Figure 4-14 Mesh for the circulation control airfoil

The solution of the baseline NACA GAW-1 airfoil is validated against the experimental data (McGbee and Beasley, 1973). Figure 4-15 and Figure 4-16 shows that both the predicted surface pressure coefficient at AOA=2° and the variation of the lift with the AOA agree well with the experiment. The predicted lift coefficient for the baseline airfoil is 0.73.

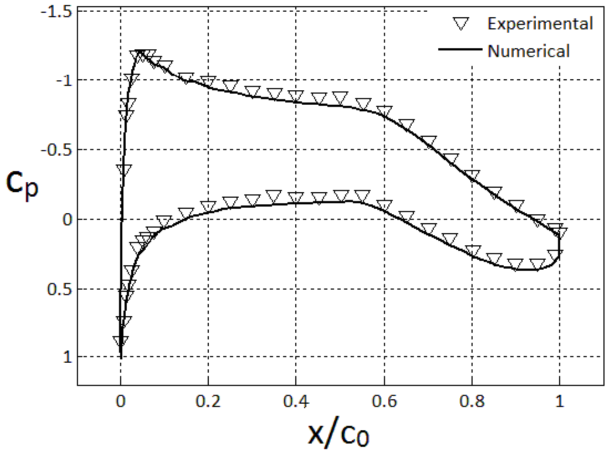


Figure 4-15 Comparison of the predicted and experimental surface pressure coefficient at AOA=2° for the baseline airfoil

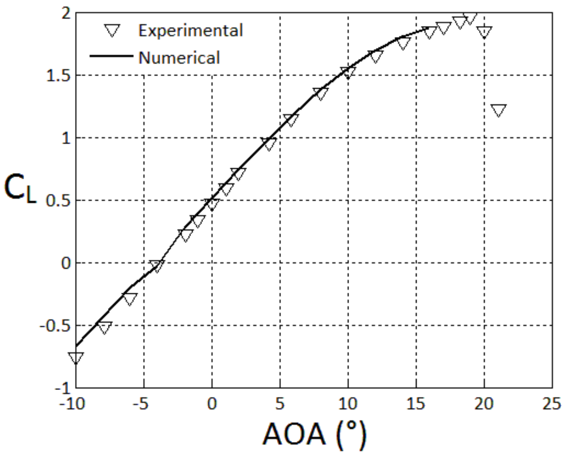


Figure 4-16 Comparison of the computed and experimental lift coefficient against AOA for the baseline airfoil

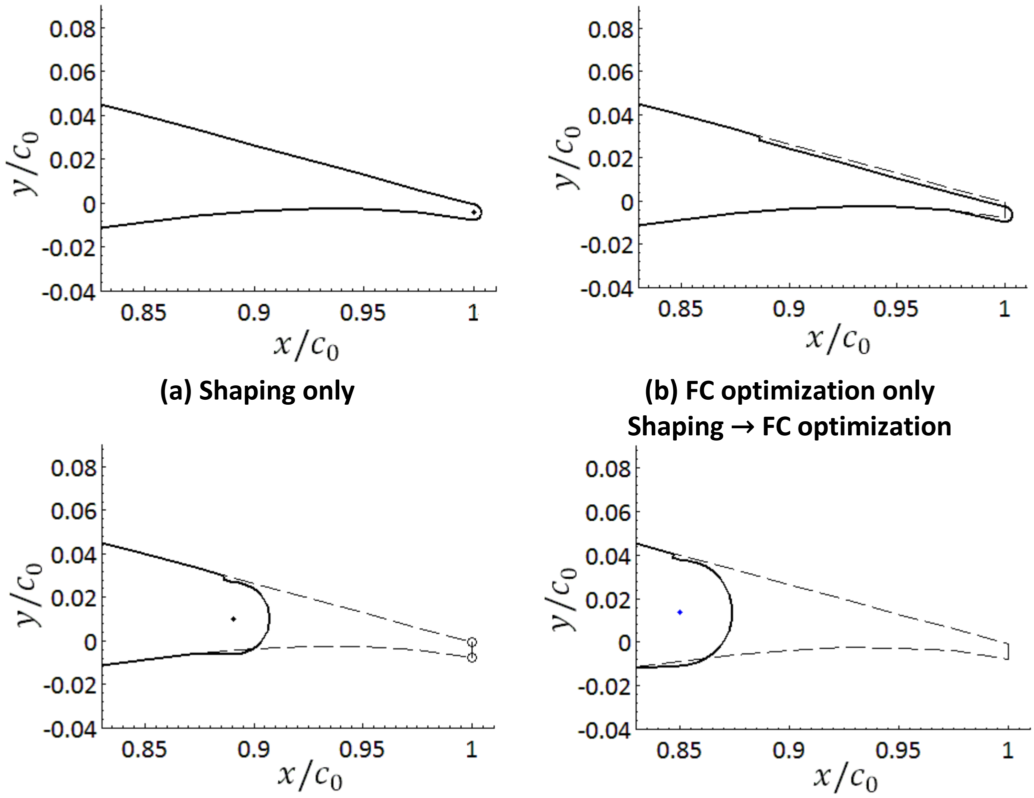
4.2.3 Results and Discussion

To obtain the results, the flow field is computed in parallel on the same computer with 10 CPU cores. Without the flow control, each flow solution takes about 2 minutes to converge. With the circulation control, the convergence time is about 7 minutes. The total optimization time is about one hour for shaping only and about 4.5 hours for flow control only. Thus, it takes about 5.5 hours for a sequential optimization. For the concurrent optimization it takes about 12 hours with a 9×9 full factorial design-of-experiment samples and 25 refined samplings within the proximity of the optimum. Since the optimized configurations are different from the baseline, an additional mesh dependence study has been performed and has confirmed that the optimized results are mesh-independent.

The results of the five optimization approaches are listed in Table 4-7. The optimal trailing edge configurations and the flow fields are shown in Figure 4-17 and Figure 4-18.

Table 4-7 Maximized C_L by different optimization approaches

Design	$I = C_L$	C_L/C_D	Optimized configuration parameters	
			x_{te}/c_0	x_{jet}/c_0
Baseline	0.73	61.3	1	---
Shaping only	0.73	61.3	1	---
FC optimization only	0.88	62.4	1	0.887
Shaping → FC optimization	0.88	62.4	1	0.887
FC optimization → shaping	1.01	60.3	0.890	0.887
Concurrent shaping and FC optimization	1.40	60.3	0.848	0.850



(a) Shaping only

(b) FC optimization only
Shaping → FC optimization

(c) FC optimization → Shaping

(d) Concurrent optimization

Figure 4-17 Optimized design by different optimization approaches, dash line: baseline design; solid line: optimized design

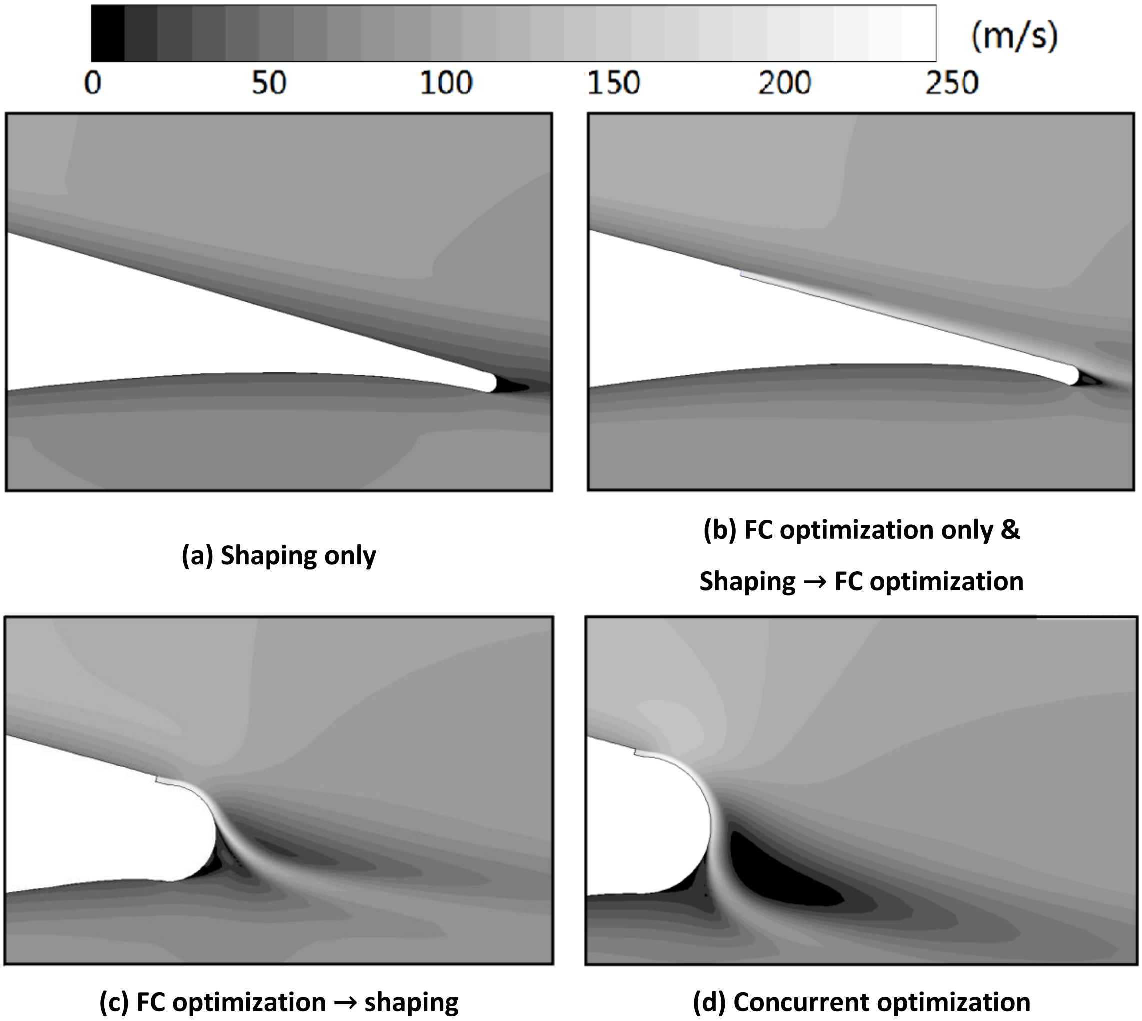


Figure 4-18 Flow field velocity magnitude of the optimized designs by different optimization approaches

For shaping only, the optimum shape takes on the baseline shape, with the lift coefficient of 0.73, which is an obvious results, since the sharp and cambered trailing edge makes the stagnation point at the leading edge and separation point at the trailing edge to stay lower than those of the blunt trailing edge airfoil, resulting in a slightly higher circulation. For the flow control only, the lift coefficient is 0.88, 17% higher than the baseline. However, the optimized result does not effectively capture the Coanda effect. For the sequential optimization “shaping → FC”, the result (Figure 4-18b) is the same as the that of the flow control optimization only, since the baseline shape is the optimal shape when the flow control is not applied. However, in the second sequential optimization “FC

→ shaping" (Figure 4-18c), the Coanda effect is captured. As a result, the lift coefficient is increased by a significant amount to 1.01. However, the highest lift is obtained by the concurrent optimization, with the largest trailing edge radius and the strongest Coanda effect (Figure 4-18d). The lift coefficient is further increased to 1.4, nearly twice the lift of the baseline airfoil. Such a significant lift increase is possible as has also been shown in previous circulation control studies (e.g. Min et al., 2009).

As the results show, the combined optimizations produce a significantly higher lift coefficient than the shape optimization alone or flow control optimization alone. Moreover, in the combined optimizations, the configurations and the lift performance are significantly different between different sequencings of the disciplines.

4.2.4 Sequencing of the Disciplines

To investigate the difference in the optimized results with different sequencings of the disciplines, the shape-only design space (no FC) and the shape-FC-combined design space (with FC) are shown in Figure 4-19 (top-down view) and Figure 4-20 (3D view). For a better illustration, the axes are based on the shape parameter x_{jet} and FC parameter x_{te}. The valid domain under the geometric constraint $x_{jet} \leq x_{te}$ takes on a rectangular trapezoid. In the design spaces, migration routes of the optimum by each optimization approach are shown respectively.

For "shaping only" (Figure 4-19a and Figure 4-20a) and "FC only" (Figure 4-19b and Figure 4-20b), the search for the optimum is confined in either the shape or FC dimension. When shaping and FC optimization is combined, the design space is opened up by the addition of another dimension. The resulting combined design space gives a wider space to search for a higher performance.

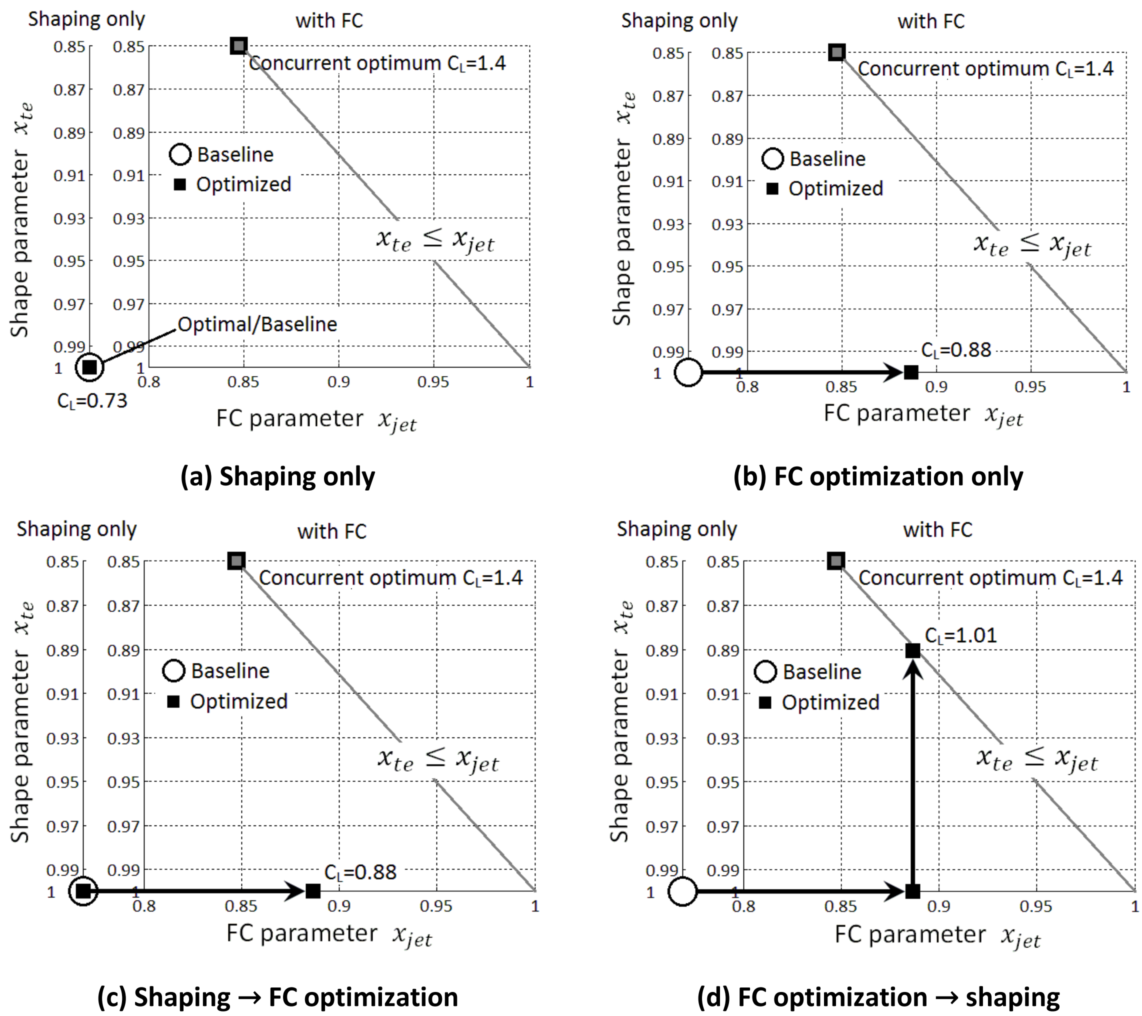


Figure 4-19 Search routes of different optimization approaches in the design space

In the combined optimizations, the sequencing of the disciplines is essentially how the combined design space is searched. For "Shaping → FC", the optimization first searches along the shape dimension, then marches along the FC dimension from the shaping-only optimum into the combined design space. For "FC → Shaping", the optimization first searches along the FC dimension, then marches along the shaping dimension from the "FC optimization only" optimum into the combined design space. In any case, the sequential search is still a one-dimensional search, albeit that the search direction is changed when optimizing the other discipline. While for the concurrent approach, the optimization searches the combined design space plane thoroughly without being confined on the a particular search lines. Therefore, the global optimum in the combined design space is picked up by the concurrent approach, but missed by the sequential approaches, so that the concurrent approach obtains the highest lift.

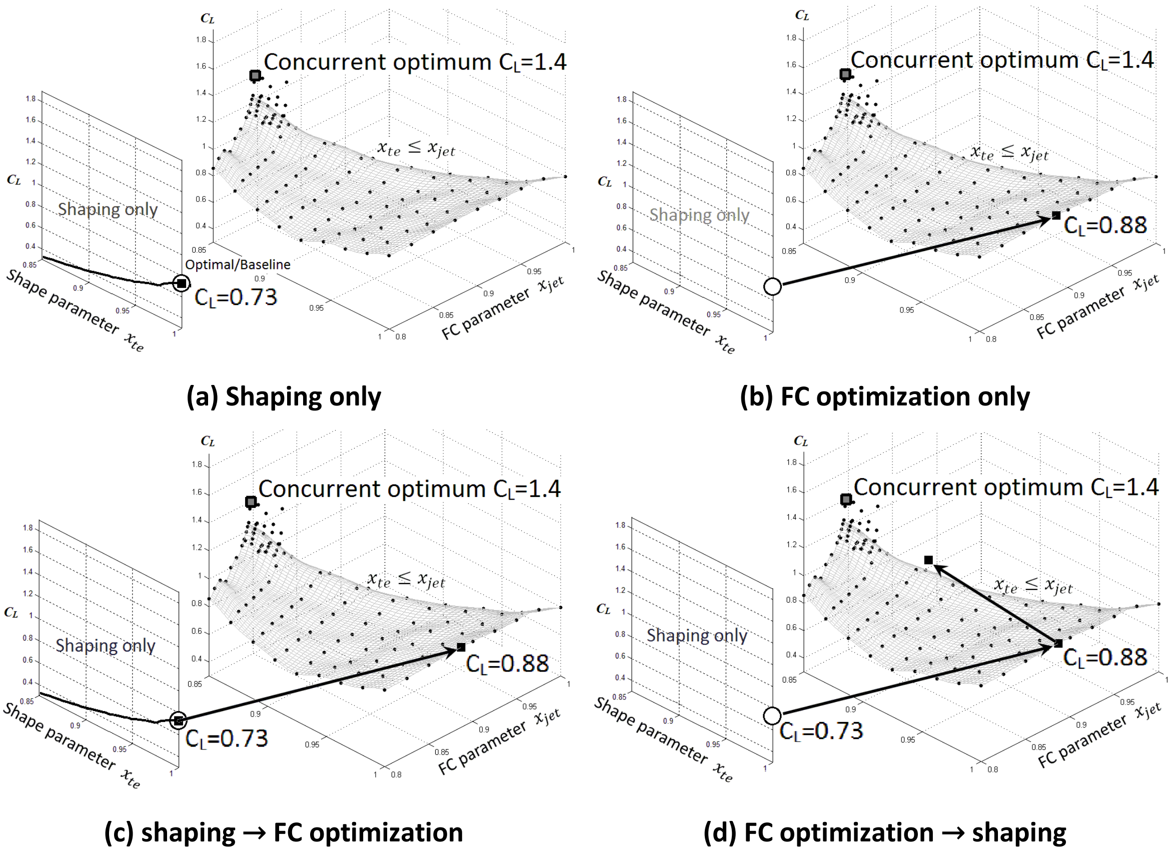


Figure 4-20 3D view of the search routes of different optimization approaches in the design space

Given the issue of sequencing, one may still question that, if the sequential searches are allowed to continue (namely that the shape and FC are allowed to be optimized iteratively), will they eventually lead to the global optimum in the combined design space? And if so, how many more repetitions/iterations are required? To address this question, let the sequential optimization continue from the optimum of the "FC → Shaping", and optimize the FC again ("FC2"), as shown in Figure 4-21. Interestingly, the search cannot find a better solution along the FC dimension, and the optimum is still the optimum of the "FC → Shaping". After this, it is quite obvious that if the shape is to be optimized again, the result will still be the optimum of the “FC → Shaping”. If such a scenario happens, the sequential optimization is “stuck” and will not reach the global optimum in the combined design space, no matter how many further repetitions are performed. For the present airfoil case, the “stuck” scenario will also happen if the sequential optimization continues from the “Shaping → FC”. The occurrence of being stuck is a consequence of the sequencing, and is not related to which algorithm being used in the optimization. To become “unstuck” and obtain the highest performance, the concurrent approach can be used.

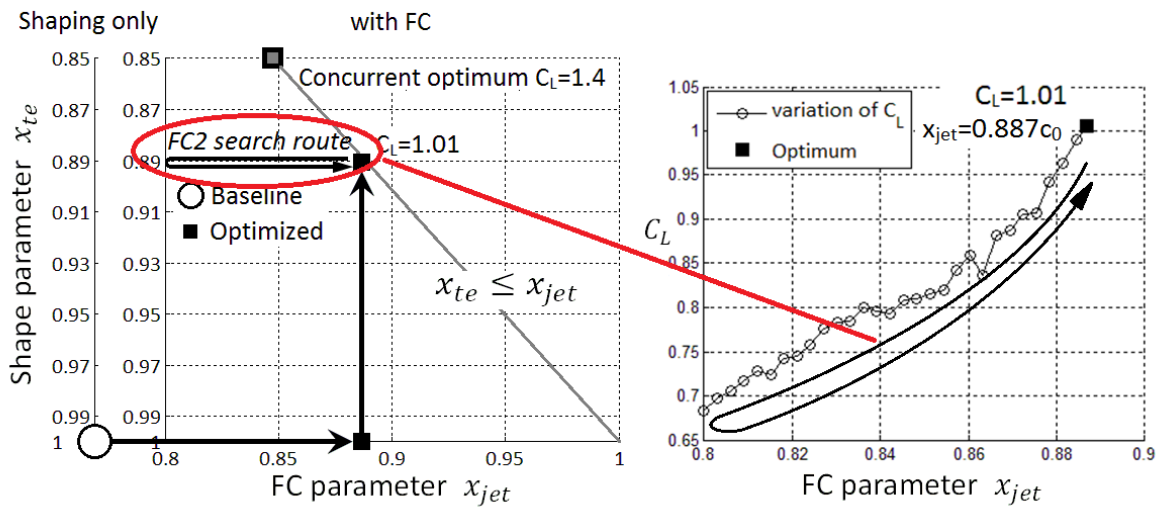


Figure 4-21 Search route for the “FC2” optimization

But if in a luckier case where the sequential approaches will not encounter the “stuck” scenario, then how many optimizations will it require to find the global optimum, and how efficient will it be, as compared to the concurrent approach? To address this question, one can consider a simple numerical experiment with the objective function f of two design variables x_1 and x_2:

$$f(x_1, x_2) = \exp\{-(x_1 - 0.5)^2 - (x_2 - 0.5)^2 + (x_1 - 0.5)(x_2 - 0.5)\},$$
$$x_1, x_2 \in [-1.5, 1.5] \qquad \textbf{(4.18)}$$

Suppose that x_1 is the design variable for Discipline 1 and x_2 is the design variable for Discipline 2. Also we assume that the baseline configuration is represented by $[x_1, x_2]$=[-1,

-1]. As shown in Figure 4-22, the two sequential approaches take 8 and 9 optimizations respectively to find the global optimum. While for the concurrent approach, only one optimization is required. It seems clear that, as expected, the concurrent approach is more efficient in terms of the number of optimizations required.

However, the number of active design variables by the concurrent approach is larger than by the sequential approaches, which can make one concurrent optimization more expensive than a single optimization in the sequential approaches. Therefore, the difference on the efficiency between the concurrent approach and the sum of all the optimizations in a sequential approach cannot be generalized, and is dependent on the particular problem being investigated. It is believed that different optimization options should be made available if the optimization potential can be fully explored for a given resource budget in time and cost. And for consistency, different optimization approaches should be applied under the same optimization framework.

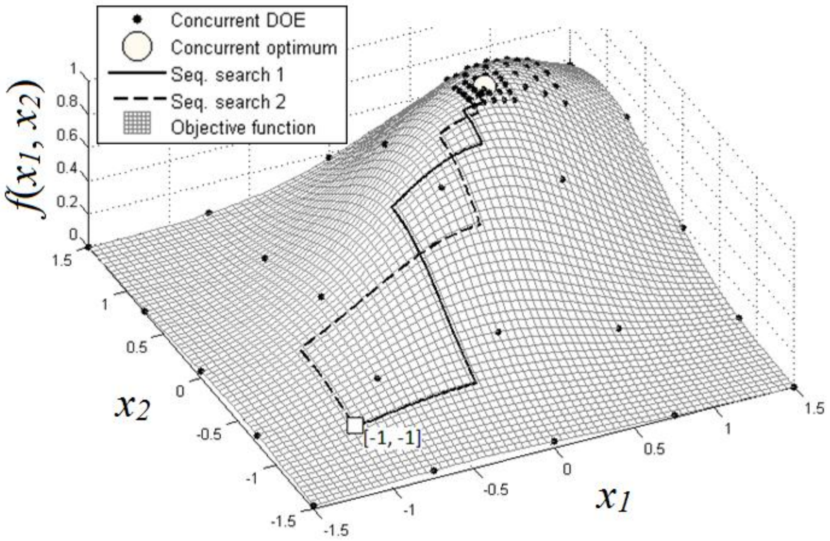


Figure 4-22 The demonstrative objective function in the combined design space, the sequential search routes, and the concurrent optimum

4.3 Concluding Remarks

Conventionally, shaping and flow control tend to be developed and applied as two separate disciplines in improving the performance of the flow devices. In this chapter, the interactions between the two disciplines are examined with two aspects of interest, 1) the extra performance gain by combining the two disciplines, and 2), the impact of the sequencing of the two when combined.

In two case studies chosen for internal and external aerodynamics, the combination between the shape optimization and the flow control optimization has demonstrated a significantly higher potential for performance improvement than the approaches of either shaping only or flow control optimization only. In the case of the compressor blade, the combined optimizations satisfies both objectives of reducing the loss at the design inlet flow angle and increasing the operating range at the same time, whereas the pure shape optimization can only satisfy either one of the two objectives. In the case of a circulation control airfoil, the combined optimizations produce a markedly higher lift than the approach of either shaping only or flow control optimization only.

The sequencing of the disciplines is also examined for the circulation control airfoil case. The highest performance was obtained by the concurrent optimization approach, because the global optimum in the combined design space is found. It has also been observed that a sequential approach may become locally “stuck” for certain objective function and constraints and unable to reach the optimum found by the concurrent approach even with repetitive sequential search. If on the other hand, the restarted sequential optimizations are adopted to avoid being stuck, many more optimization iterations would be required, compared to the concurrent approach.

The 2D investigations presented in this chapter show that the combination of shaping and flow control optimization is effective in further improving the performance, and the sequencing in the disciplines when combined makes remarkable difference in the final optimized performance. The results justify further pursuit of the topic onto the 3D realistic cases on a wider scope of problems such as adding prediction of temperature and heat transfer. As introduced earlier, there is a distinctively different requirement by the conventional shaping optimization, compared to that by the flow control. Due to the complexity of the controlled flow field (e.g. high gradient and mixing around flow injection), the fidelity of RANS used in this chapter is still too low for realistic applications. The potential advantage of combining flow control optimization with the shaping points to the need for high-fidelity LES to accurately evaluate the flow control related performance gain. The conflict between the computational cost of the high-fidelity LES and the fast solution speed as required by the design optimization also highlights the need for developing an efficient LES method for the shape and flow control design optimizations. The development of such a method is to be addressed in the subsequent chapters.

Chapter 5

Baseline LES Validation

The 2D investigation presented in Chapter 4 shows that combining shaping and flow control optimization further improves the performance and motivates further investigations for 3D realistic cases. As it is widely accepted that RANS is insufficient to predict accurately for complex unsteady flows, unsteady simulations which offers higher flow solution fidelity is then required. In many complex unsteady flows (particular those associated with flow control), the fidelity of the URANS may still be inadequate (Spalart, 2000; Joo and Durbin, 2009; Ravelli and Barigozzi, 2014). This means the LES is required. To appreciate the use of LES, this chapter provides a detailed LES study on a few configurations with complex turbulent unsteady flows, and compare the LES solutions with those of the RANS and the experiments to validate the LES capability of the solver.

Two cases are chosen for the present study. The first case is fundamental – the flow past a circular cylinder. The geometry is simple but the flow field is still complex, turbulent and unsteady. It is a popular case for the unsteady flow experiments, LES studies and validations for various CFD methods. It is of wide research interest and has abundant well-established experimental and numerical data for comparison. The second case is more practical - the trailing edge cutback injection on a model blade/airfoil. As the cutback injection configuration is applied in most high pressure turbine (HPT) blades, being an important research point, some well-established experimental data as recently published can be utilised.

5.1 Case I: Flow past a Circular Cylinder

5.1.1 Case Description

The case chosen is running at the Reynolds number of $Re = 3900$ based on the cylinder diameter D. According to the available experiments and previous LES studies, the freestream is low speed and the flow can be assumed incompressible. The temperature effect is neglected as it is passive as well as its variation is insignificant at such a low speed. At low Mach numbers, the characteristic of the cylinder flow is chiefly determined by the Reynolds number. At $Re = 3900$, the flow on the cylinder is laminar before separation, and transition to turbulence takes place at the separation shear layer due to the Kelvin-Helmholtz instability. The breaking of the shear layer results in the vortex shedding in the wake. The shedding has a dominant frequency f, which can also be described by the Strouhal number:

$$St = \frac{f \cdot D}{U_\infty} \qquad \textbf{(5.1)}$$

where D is the diameter of the cylinder and U_∞ is the freestream velocity magnitude. For a long cylinder (i.e. largely 2D), the Strouhal number is largely determined by the Reynolds number, as shown in Figure 5-1 below (White, 2003, p. 296).

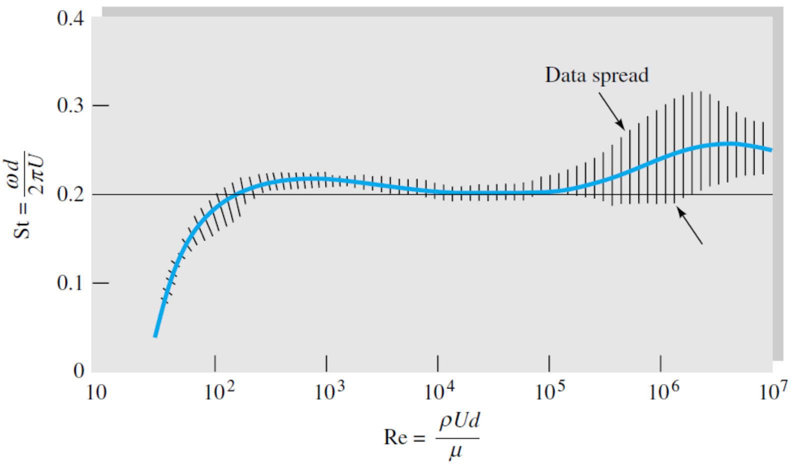


Figure 5-1 Relation between the Strouhal number and the Reynolds number (White, 2003)

The detailed setup of the case largely follows the experiment of Parnaudeau et al. (2008). The diameter of the cylinder is $D = 0.012\,m$, the molecular viscosity is $\mu = 1.7894 \times 10^{-5}\,kg/(m \cdot s)$ as a constant and the density is $\rho = 1.225\,kg/m^3$ as a constant. To match the Reynolds number, the freestream velocity is set to $4.75\,m/s$.

The geometry and the computational domain is quasi-three-dimensional, uniform in the spanwise z-direction, as shown in Figure 5-2 below. The computational domain in the XY-plane is circular in the diameter of $30D$ and is concentric with the cylinder. The axis of the cylinder lies on the z-axis, i.e. $x = 0$, $y = 0$. The length of the span is $Z = \pi D$, which is also chosen by many previous LES studies (e.g. Parnaudeau et al., 2008; Kravchenko and Moin, 2000; Breuer, 1998; etc.).

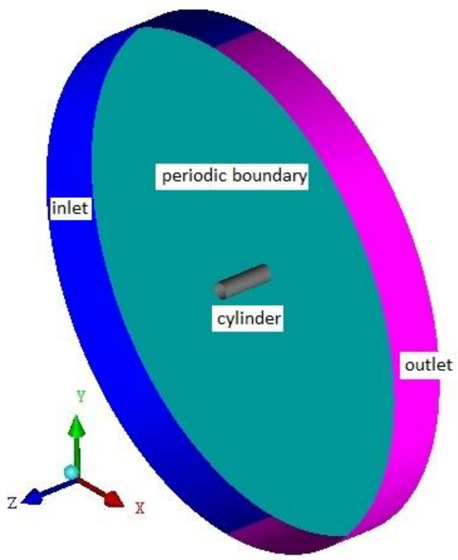


Figure 5-2 Computational domain of the cylinder

The far-field boundary is split into the inlet part and the outlet part at $x = 0$. On the inlet boundary, the "velocity-inlet" is prescribed to set the freestream velocity as $U_\infty = 4.75\,m/s$, parallel to the x-direction. The "no perturbation" condition is prescribed for the freestream turbulence, as the freestream turbulence will become negligible when reaching the cylinder. On the outlet boundary, the "pressure-outlet" is prescribed with the averaged static pressure of 1 atm. The average pressure specification is conducive to reducing the wave reflection back into the domain. On the cylinder surface, the no-slip wall boundary is specified. Periodic boundary condition is specified on the two side boundaries.

The mesh is generated using the ANSYS ICEM CFD program. It is an O-type mesh with all cells being hexahedral, as shown in Figure 5-3 and Figure 5-4 below. The number of cells is $280 \times 200 \times 64$ in the circumferential, radial and spanwise direction respectively. In the spanwise direction, the 64 cells are distributed uniformly as shown on the cylinder in Figure 5-4 together with the mesh on the midspan cutplane. In the circumferential direction, the mesh is made slightly denser in the rear part of the cylinder as the turbulent eddies only exist at the rear part and in the wake. In the radial direction, the mesh is clustered near the surface, making the y+ smaller than one, with the radial expansion ratio of the mesh cell height less than 1.04. Overall, the mesh density is considered sufficient compared to other cylinder LES studies (e.g. Kravchenko and Moin, 2000; Parnaudeau et al., 2008) at this Reynolds number using the same order of spatial discretization.

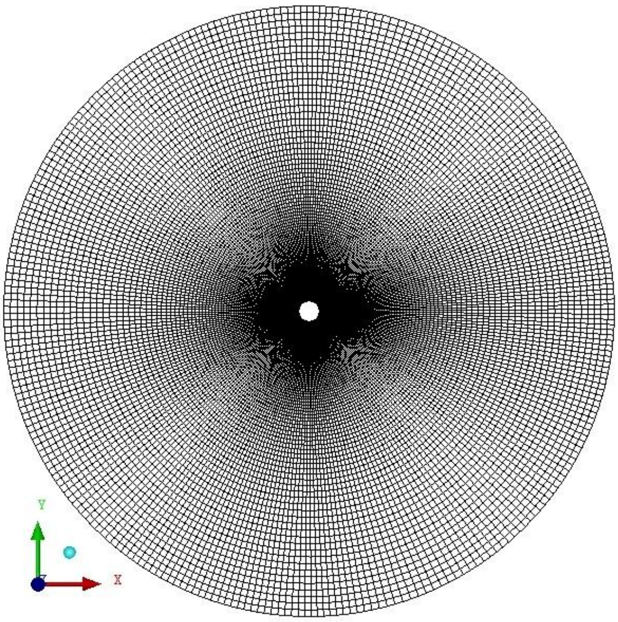


Figure 5-3 Computational mesh on the midpan cutplane view in z-direction

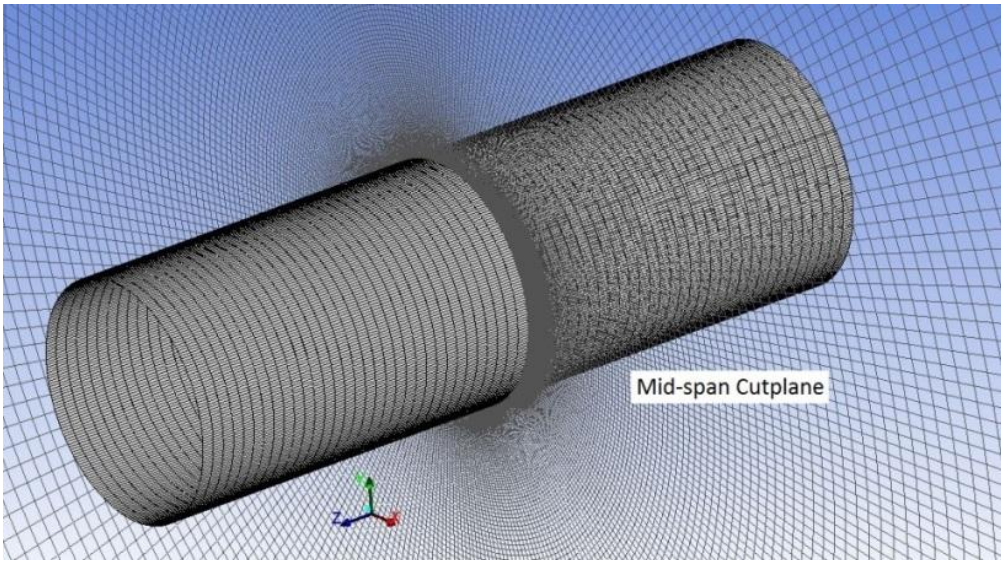


Figure 5-4 Mesh on the cylinder and the midspan cutplane

The pressure-based solver is used for the LES. The sub-grid scale (SGS) model is the Smargorinsky-Lilly model (Smargorinsky, 1963). The spatial discretization for the momentum equation is selected as the bounded central differencing (BCD) scheme (Jasak, 1999), and the second-order interpolation is used to compute the pressure on the cell-face (as discussed in Chapter 3). The gradients of the variables are computed by the Least-Square Cell-Based (LSCB) method.

The temporal discretization is second-order implicit with a constant time step specified. The "non-iterative time-advancement" (NITA) with fractional step method option is selected as the pressure-velocity coupling to reduce the computing cost per time step. From the relationship between the Strouhal number and the Reynolds number in Figure 5-1, the Strouhal number is around 0.2 at the Reynolds number of 3900. Thus the dominant vortex shedding frequency is estimated to be 79.167Hz, and the estimated shedding period is $T = \frac{1}{f} = 1.2632 \times 10^{-2}\ s$. This estimation is used to determine the time step size, which is then set to $\Delta t = 1.2632 \times 10^{-5}\ s$ so that there will be around 1000 time steps per shedding period, which is a typical time-resolution used by other cylinder LES studies (e.g.

Parnaudeau et al., 2008). A time-dependent study with half the size of time-step has also been performed and the results though not reported here showed the present time-resolution is sufficient.

5.1.2 LES Validation Results

The LES is initialized with the incompressible potential flow solution. To drive away initial solution and statistically establish the turbulent flow, the transient stage time integration is run with 6×10^4 time steps corresponding to about 10 domain through-flow times or 60 shedding periods. As the fluctuation patterns of the lift and drag are established, as shown in Figure 5-5 below, the time-step count is reset zero and the statistics collection stage begins to run for 12×10^4 time steps, covering around 120 vortex shedding periods, in which the solution is time-averaged on-the-fly as:

$$\bar{Q}^n = \frac{1}{n}\left[(n-1) \cdot \bar{Q}^{n-1} + Q^n\right] \tag{5.2}$$

where $\bar{Q}^n$ is the time-mean solution averaged from the 1st to the n-th time level of the statistic collection stage. Q^n is the solution of the n-th time level. The formula is implemented in Fluent via the user-defined function (UDF) executed at the end of each time step. The UDF codes were intended to but weren't included due to the page limit.

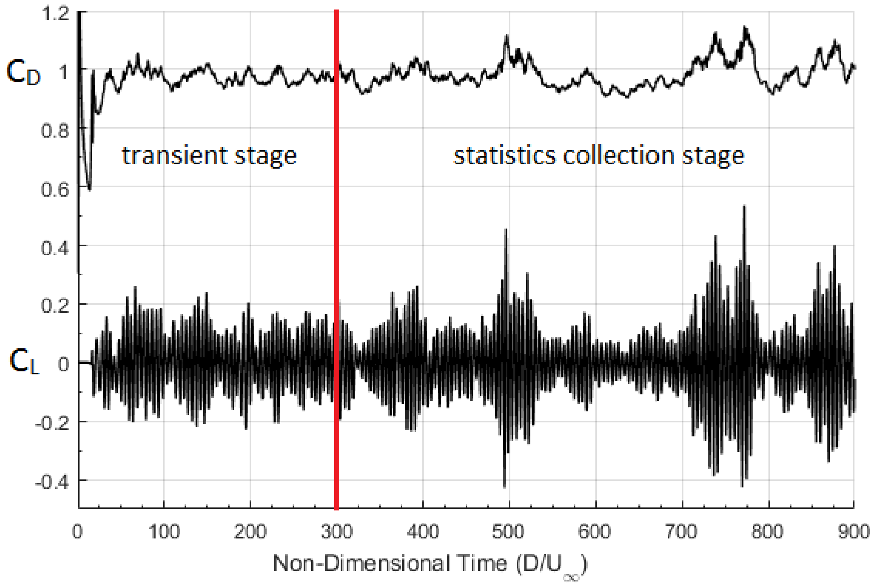


Figure 5-5 Time history of the lift and the drag coefficients, U_∞: freestream velocity

To monitor and post-process the computation, a probe is placed at the wake centre line ($y = 0$, $z = 0$) at $x = 3D$ to monitor the velocity fluctuation. The lift and drag coefficients are also monitored:

$$C_L = \frac{L}{0.5\rho U_\infty^2 ZD} \text{ and } C_D = \frac{D_{drag}}{0.5\rho U_\infty^2 ZD} \quad \textbf{(5.3)}$$

where L and D_{drag} are the total lift and drag calculated by Fluent which accounts for both the pressure and frictional forces on the surface.

The variables to be time-averaged are the velocity components u_i, static pressure p and the velocity correlations $u_i u_j$ $(i, j = 1, 2, 3)$. The time-averaged solution can be further averaged spanwisely to produce the t-z-averaged solution. For the discussion here, the time-averaging of a variable ϕ is denoted as $\bar{\phi}$ and the t-z-averaging as $\langle\phi\rangle$. Hence the time-averaged and t-z-averaged Reynolds stresses are $\overline{u_i' u_j'} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j$, and $\langle u_i' u_j' \rangle = \langle u_i u_j \rangle - \langle u_i \rangle \langle u_j \rangle$.

First, the integral quantities of the present LES are summarized in Table 5-1 and compared with the experiments and other LES studies. Quantities to be compared are the time-mean drag coefficient C_D, t-z-mean static pressure coefficient at the cylinder back point $C_{p,back}$, vortex shedding Strouhal number St, the recirculation length L_r, and the root-mean-square (RMS) of the fluctuation of the lift coefficient $C_{L,RMS}$ (with almost zero mean). L_r is defined as the length of the region with negative t-z-mean x-velocity from the back point of the cylinder in the wake centreline. In the experiments, the C_D and $C_{p,back}$ (and also the surface C_p) obtained by Norberg are reported by Kravchenko and Moin (2000). The experimental $C_{L,RMS}$ from the review of Norberg (2003) is the $C_{L,RMS}$ per unit span, which is derived from the functional fitting from available experiments at other Reynolds number. As far as the author can find, there appears to be no publication of experimental measurement of $C_{L,RMS}$ at the present Reynolds number of 3900.

By the comparison, the mean drag, mean back pressure agrees well with Norberg's measurement (reported by Kravchenko in Moin, 2000), as well as most of other LES. The predicted Strouhal number and the recirculation length agree well with the measurement of Parnaudeau et al. (2008). The RMS lift coefficient is larger than the LES prediction of Beaudan and Moin (1994) but close to the LES of Ouvrard et al. (2010, case F1, with same SGS model). According to the review of Norberg (2003), there is a large scattering of the reported $C_{L,RMS}$ near $Re = 3900$ from both the experiments and numerical simulations, and its more difficult to predict well.

Table 5-1 Integral quantities comparison with other experiments and LES

Cases	C_D	$C_{p,back}$	St	L_r/D	$C_{L,rms}$	Span
Present LES	0.9803	-0.837	0.210	1.60	0.128	πD
Experiments						
Parnaudeau et al., 2008	---	---	0.208$\pm$0.002	1.60	---	---
Cited by Kravchenko and Moin, 2000	0.99$\pm$0.05	-0.88$\pm$0.05	0.215$\pm$0.005	1.4$\pm$0.1	---	---
Norberg, 2003	---	---	---	---	0.083	unit span
Other LES						
Ouvrard et al., 2010, case F1	0.99	-0.85	0.218	1.54	0.125	πD
Parnaudeau et al., 2008, fine grid	--	--	0.208$\pm$0.001	1.56	--	πD
Beaudan and Moin, 1994, Fixed C_s	0.92	-0.94	0.209	1.36	0.07	πD
Breuer, 1998, case D2	1.097	-1.069	---	1.043	---	πD
Franke and Frank, 2002	0.978	-0.85	0.209	1.64	---	πD

In Figure 5-6 below, the frequency spectrum of the y-velocity at probe $x = 3D$ shows a peak Strouhal number of 0.210 of the vortex shedding.

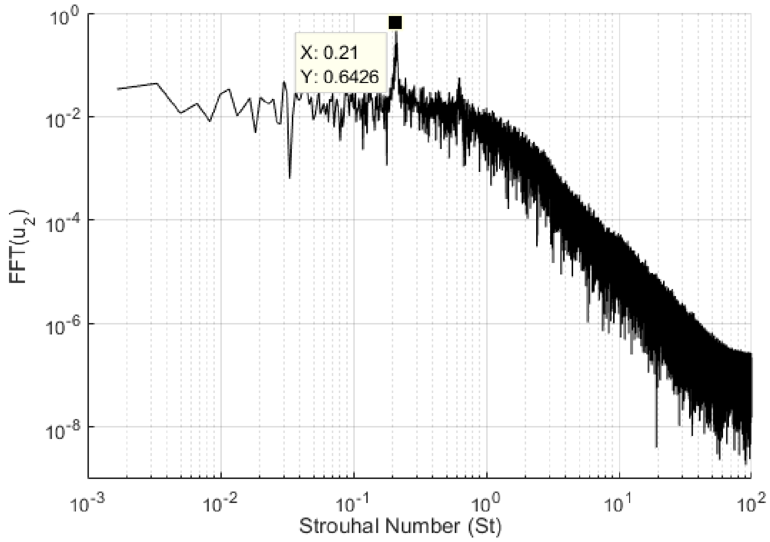


Figure 5-6 Spectrum of Strouhal number of the y-velocity component (u_2) on the wake centreline at $x = 3D$

The instantaneous Q-criterion in Figure 5-7 below shows that the prediction captures the complexity and unsteadiness of the flow that contains a wide range of turbulence scales.

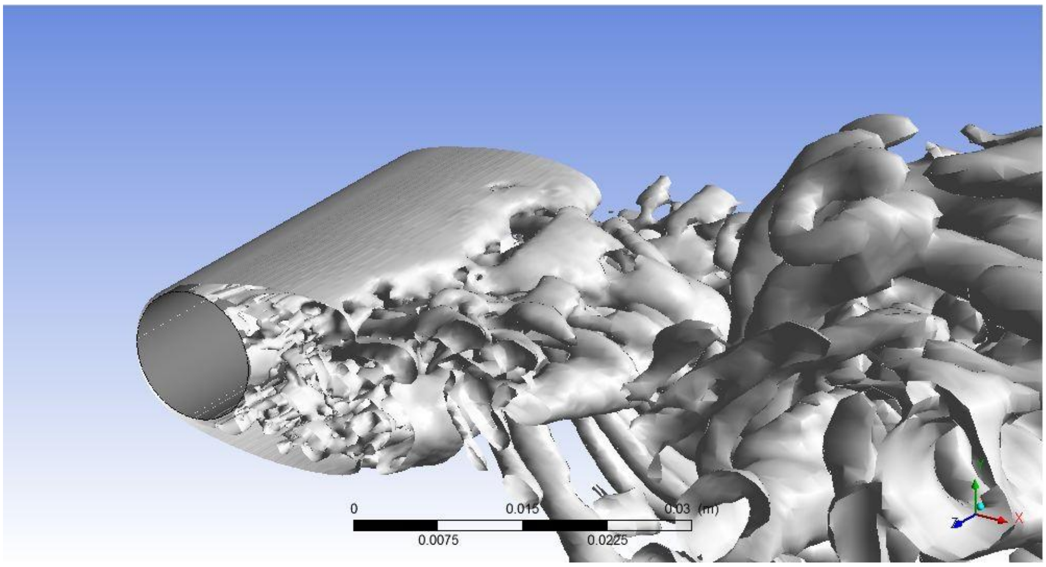


Figure 5-7 Instantaneous iso-surface of Q criterion in cylinder wake

The time-averaged flow field of the LES is shown in Figure 5-8 for the x-velocity on the midspan cutplane. The distributions of other t-z mean quantities, e.g. the surface pressure, wake-centerline x-velocity, wake-traverse velocities and the resolved Reynolds stresses are compared with the experiments and other LES from Figure 5-9 to Figure 5-12. Among these quantities, the agreement between the current LES with the experiments are relatively better for the surface pressure (Figure 5-9) and the wake traverse velocities (Figure 5-11). For the wake centreline x-velocity (Figure 5-10), the current LES predicts a smoother profile than the experiment, with an under-prediction of the recirculation length than the experiment. But the agreement is good with the LES of Parnaudeau et al. 2008 in the near wake, including the length of the recirculation zone. But the agreement is less good in far-wake downstream, possibly due to the coarser mesh resulting from radial expansion from the cylinder. In Figure 5-12, the resolved Reynolds stresses from the current and other LES appears smaller than the total Reynolds stresses measured in the experiments, possibly due to the part of the Reynolds stresses accounted for by the SGS model is not included in the LES data for the comparison. The present LES data also appears less close to the experiment than the other LES data.

In general, the present LES prediction for the first-order quantities e.g. mean drag, and mean pressure and velocity distribution is reasonable accurate. Fair agreement between the predicted second-order mean quantities e.g. $C_{L,RMS}$ and the Reynolds stresses with the experiments and other LES data is achieved, as these quantities are more challenging to predict, which can be seen by the scattering of the data in the published experimental or LES works. The present prediction is nevertheless not far away from some other LES predictions for these quantities. Certainly better agreement would be desired, but it is not the purpose of this work to make the LES itself as accurate as possible. Plus the first-order mean quantities are of more interest for the subsequent studies than the second-order mean quantities, the above LES prediction is still deemed sufficient for the purpose of this work.

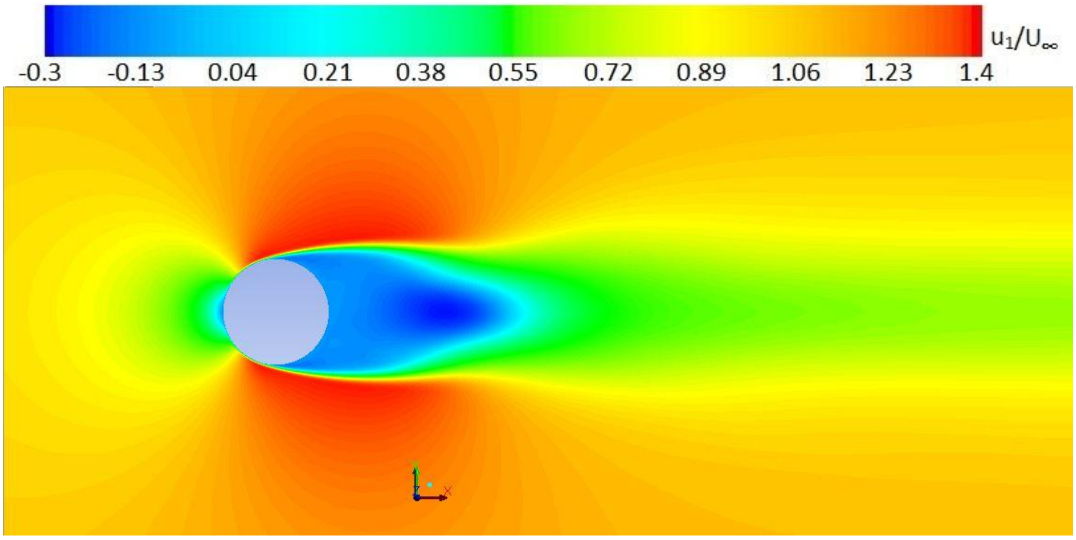


Figure 5-8 Normalized time-averaged x-velocity ($\overline{u}_1$) on the midspan cutplane

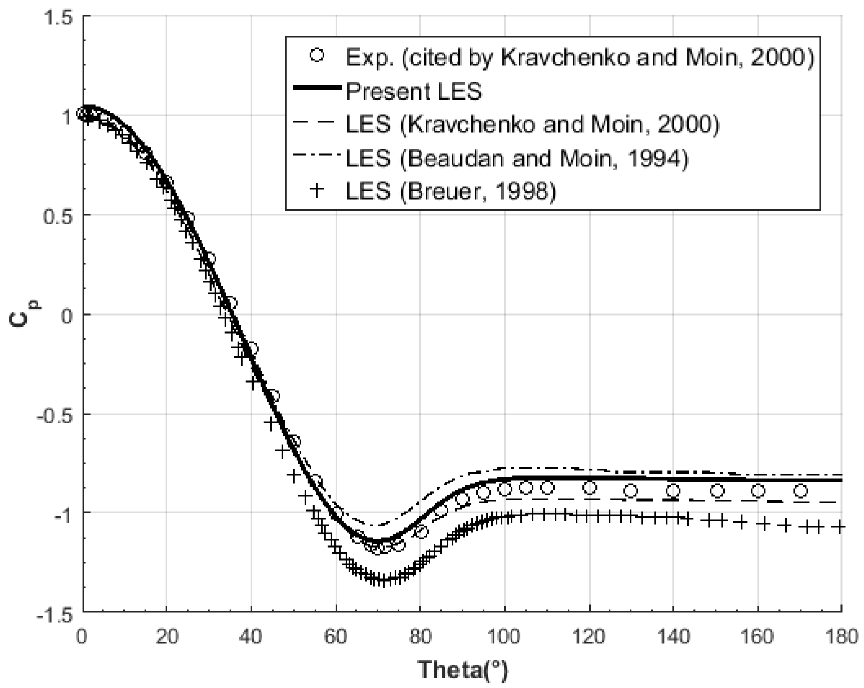


Figure 5-9 Surface pressure coefficients comparison (θ starts from the leading edge, clockwise for the upper surface and counter-clockwise for the lower surface).

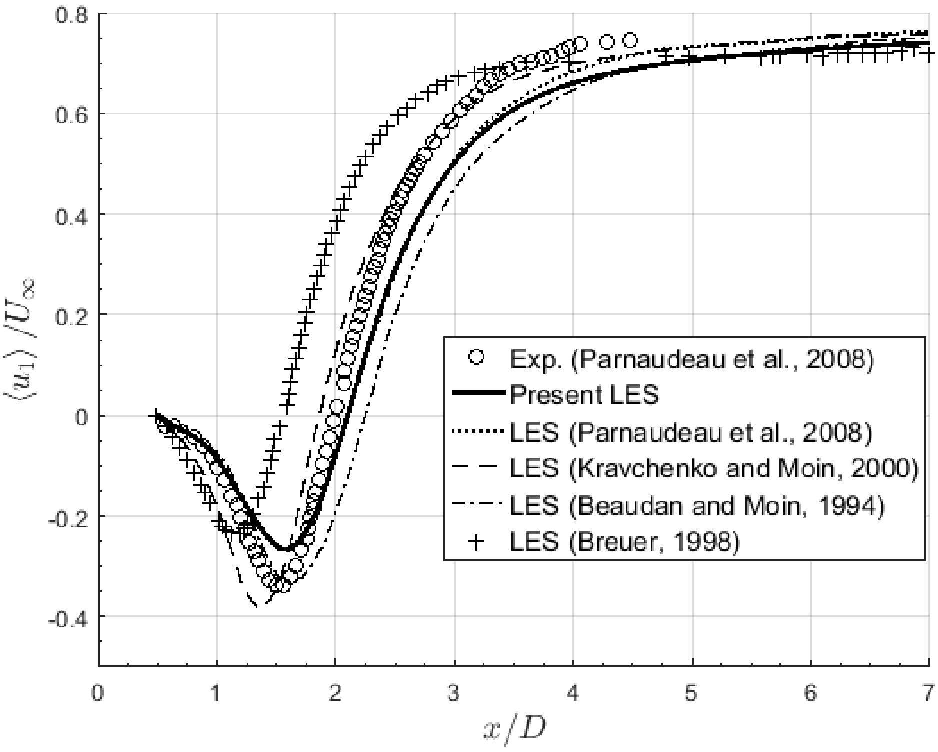
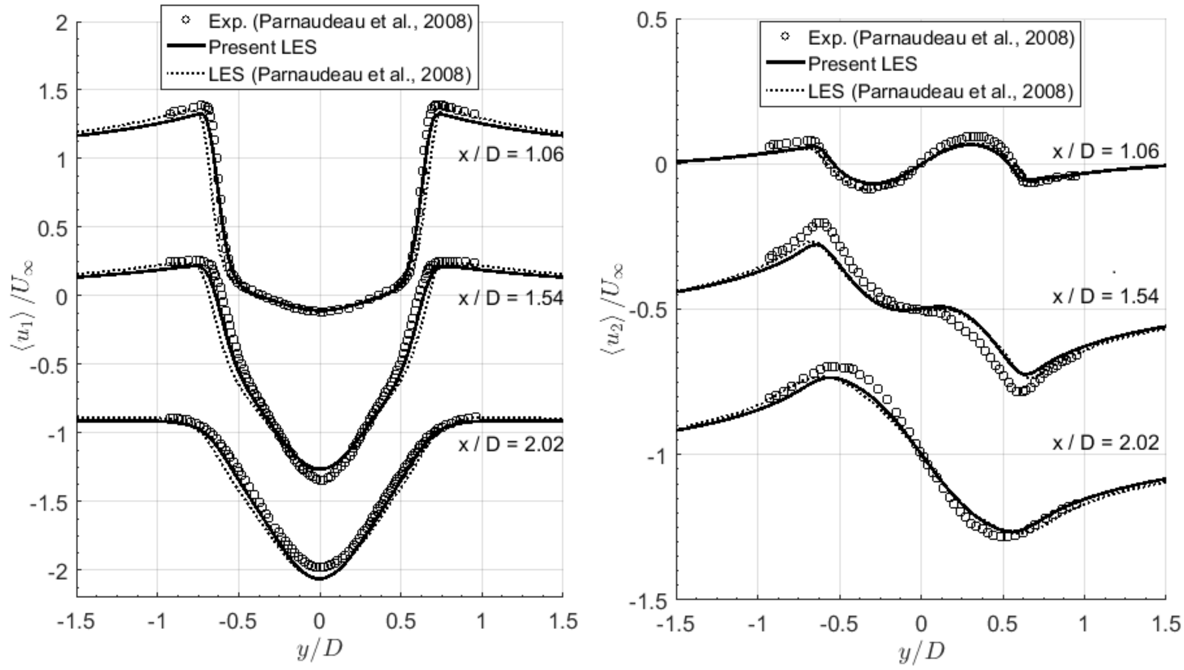
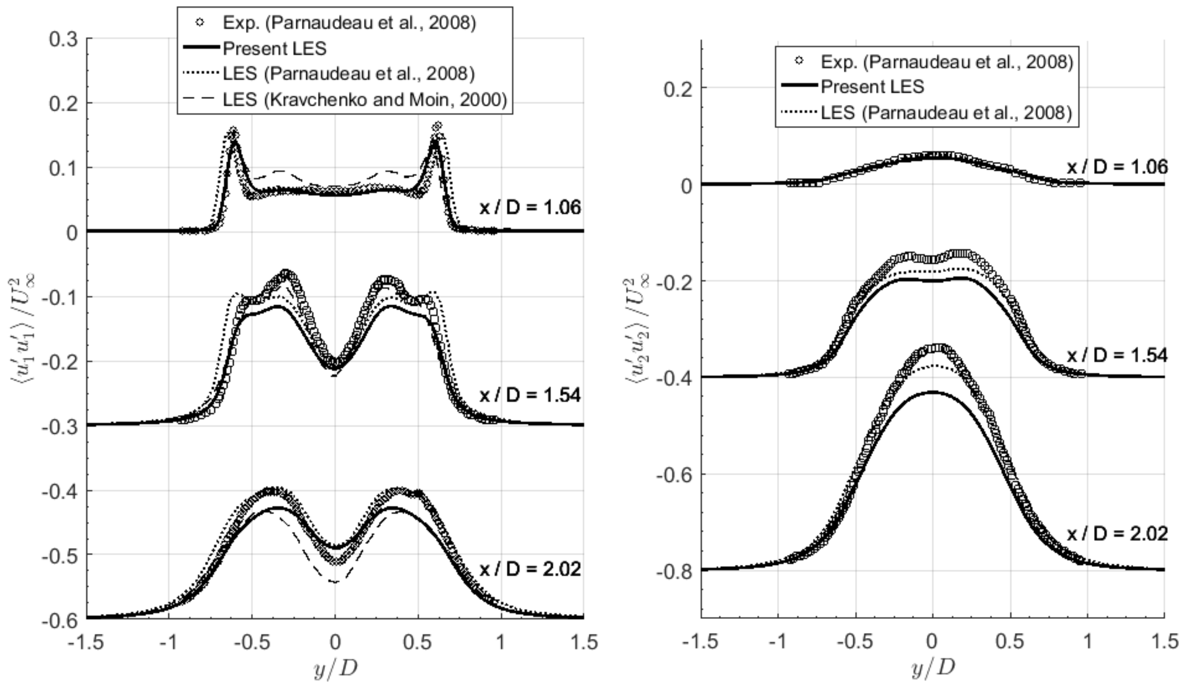


Figure 5-10 Comparison of averaged x-velocity along the centreline of the wake



(a) (b)

Figure 5-11 Comparison of the mean x-velocity (a) and y-velocity (b) in wake traverse direction at different downstream positions



(a) (b)

Figure 5-12 Comparison of Reynolds stresses at in the wake traverse direction at different downstream positions

5.2 Case II: Blade Trailing Edge Cutback

5.2.1 Case Description

Turbomachinery high pressure turbine (HPT) blades usually run at main flow temperature higher than the blade metal melting point and needs to be cooled. Film cooling injection is a typical cooling method by injecting cooled air films onto the blade surface to cover it from the hot mainstream. The blade trailing edge is also subject to intense heat load. Due to the thinness, cutback is usually applied on the trailing edge to eject the coolant which has already been used for internal cooling and can also provide some extra external film cooling.

Many studies (e.g. Holloway et al. 2002b; Martini et al., 2005b; Ravelli and Barigozzi , 2014; etc.) have shown that the injection flow is highly turbulent and contains large scale unsteadiness. The vortex shedding off the slot lip plays an important part in the turbulent mixing between the coolant and the mainstream that strongly influences the surface temperature of the trailing edge. Moreover, the injection is already highly turbulent (as the flow would pass internal pin-fins and tabulators to promote internal cooling) before coming out of the slots. Due to the intense turbulent unsteadiness, it is thus challenging to numerically predict the aero-thermal characteristics accurately for this type of configuration. Much research (e.g. Durbin, 2005; Martini et al., 2005b; Egorov et al., 2010; Ravelli and Barigozzi, 2013) has shown that RANS prediction are insufficient, and LES may be required. In the following content, this type of configuration is chosen for the baseline validation on the aero-thermal problem of the realistic cases. The LES results are also compared to the RANS prediction.

A. The Reference Experiment

The present computational study is based on a recent experiment by Ling et al. (2015). A brief description of the experiment is provided here. The experimental main flow was water, and the cooper sulphate water solution was used to represent the injection stream. Hence the experimental flow was incompressible representing the low Mach number of the air. It was pointed out in the experiment that, though in the real engine environment, the Mach number of the mainstream and the injection is often close to 1, the convective Mach number, which is the relative velocities between the main flow and the injection at the slot exit, is usually less than 0.5. As also pointed out by the other relevant experiments (e.g. Barigozzi et al., 2012), it is the convective Mach number that determines the turbulent structures of the mixing between the main flow and the injection. Therefore, the incompressible flow model was regarded as representative to the real case.

In the incompressible flow, based on the heat and mass transfer analogy, the temperature measurement was replaced by measuring the injection concentration which can then be translated back as the temperature prediction. More details are given by Benson (2011, Section 1.3). According to the experiment, the concentration measurement is a good simulation of the adiabatic condition on the blade surface.

The experiment was focused on the adiabatic film cooling effectiveness on the cutback trailing edge surface, defined as:

$$\eta_{ad} = \frac{T_h - T_w}{T_h - T_c} \tag{5.4}$$

where T_w is the static temperature on the surface, T_h is the temperature of the hot main flow and T_c is the temperature of the supplied coolant. To be consistent with the coolant concentration measurement in the experiment, both the temperatures T_h and T_c should be static temperatures.

As the interest was focused on the trailing edge, the turbine blade in the experiment was modelled by a symmetric NACA0012 airfoil, placed in a test section, as shown in Figure 5-13, that had a 2.45:1 contraction to provide a favourable pressure gradient at the trailing edge, as is the case on a real turbine blade. The airfoil and its trailing edge cutback configuration in the experiment are shown in Figure 5-14(a). The airfoil had a nominal chord length of 300mm, with the trailing edge blunted so that the actual chord length was 283mm. The upper surface at the trailing edge is cutbacked to make room for the injection slots. The shapes of the slots are identical, rectangular and 5mm in height. Between the slots are the lands to provide the structural support. The shapes of the lands are also identical. The top surface of the land follows the airfoil curvature. The lands in the experiment were varied with different taper angles to study the effect on the adiabatic effectiveness. The taper angle, as shown in Figure 5-14(b), is measured with respect to the x-streamwise direction. The land with the taper angle of 0° is called the straight land. Inside the airfoil, an internal chamber supported by several arrays of cylindrical pin-fins was made to direct the injection air to the slots.

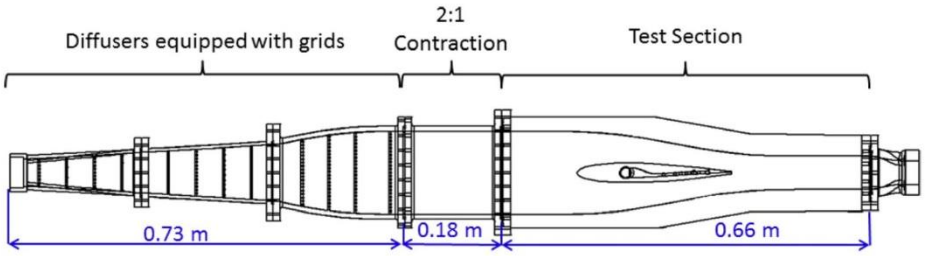


Figure 5-13 Wind tunnel of the experiment, figure from Ling et al. (2015)

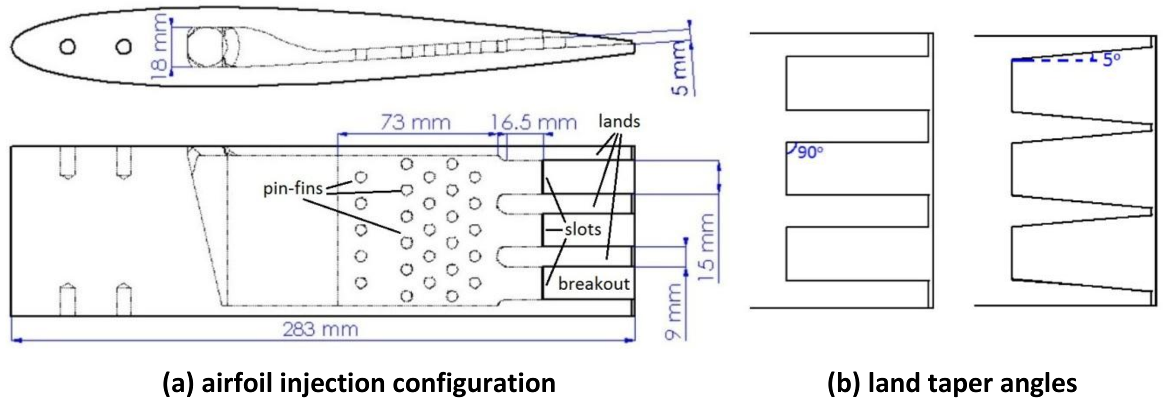


Figure 5-14 Experimental cutback injection configuration, figure based on Ling et al. (2015)

The Reynolds number in the experiment was 110,000 based on the airfoil chord and the test section inlet velocity. According to Ling et al. (2015), though the Reynolds number is often close to 1,000,000 in the real engine environment, the Reynolds number of 110,000 in the test still represents the fully turbulent flow. Another relevant experiment (Benson et al., 2012) has shown that in the range of Reynolds numbers enclosing the present Reynolds number, the mean flow structure is not strongly dependent on the Reynolds number.

The blowing ratio in the experiment is 1.3. It is defined as:

$$BR = \frac{\rho_c U_c}{\rho_m U_m} = 1.3 \tag{5.5}$$

where U_c is the velocity of the coolant injection at the slot, and U_m is the area-weighted-averaged flow velocity on the streamwise cutplane adjacent to the upper airfoil surface at the slot exit. For the incompressible flow, the coolant density ρ_c and main flow densities ρ_m are the same. So the blowing ratio becomes the velocity ratio. In summary, the key parameters of the configuration are listed in Table 5-2.

Table 5-2 Dimensions of the trailing edge cutback injection configuration

Item	Dimension
Airfoil type	NACA 0012
Airfoil chord length (nominal / actual)	300 mm / 283 mm
Airfoil span per slot and land	24 mm
Land width at slot	9 mm
Slot width	15 mm
Slot height (h)	5 mm
Slot lip thickness	3.7 mm
Trailing edge lip thickness	4.4 mm
Streamwise length of the breakout surface	41 mm
Pin-fin diameter	5 mm

B. Computational Setup

The experimental configuration with the land taper angle of 5° is chosen for the LES validation. The LES domain, geometry and boundary conditions are shown in Figure 5-15 below, showing the key parts of the configuration. The air flow is set to be incompressible. The other flow properties are also set constant, with the density of $\rho = 1.225\ kg/m^3$, the molecular viscosity of $\mu = 1.7894 \times 10^{-5}\ kg/(m \cdot s)$, and the specific heat capacity of $c_p = 1006.43\ J/(kg \cdot K)$. The leading edge of the nominal airfoil lies at $x = 0$. To reduce the computational cost, the LES domain only includes the part of test section truncated at $x = 230\ mm$ towards the outlet, containing the trailing edge, the cutback injection region and the coolant chamber. Since the slots are identical and the lands are identical, the span of the LES domain only contains a whole slot and two half-lands on the sides, with the periodic boundaries applied at the mid-span of the lands.

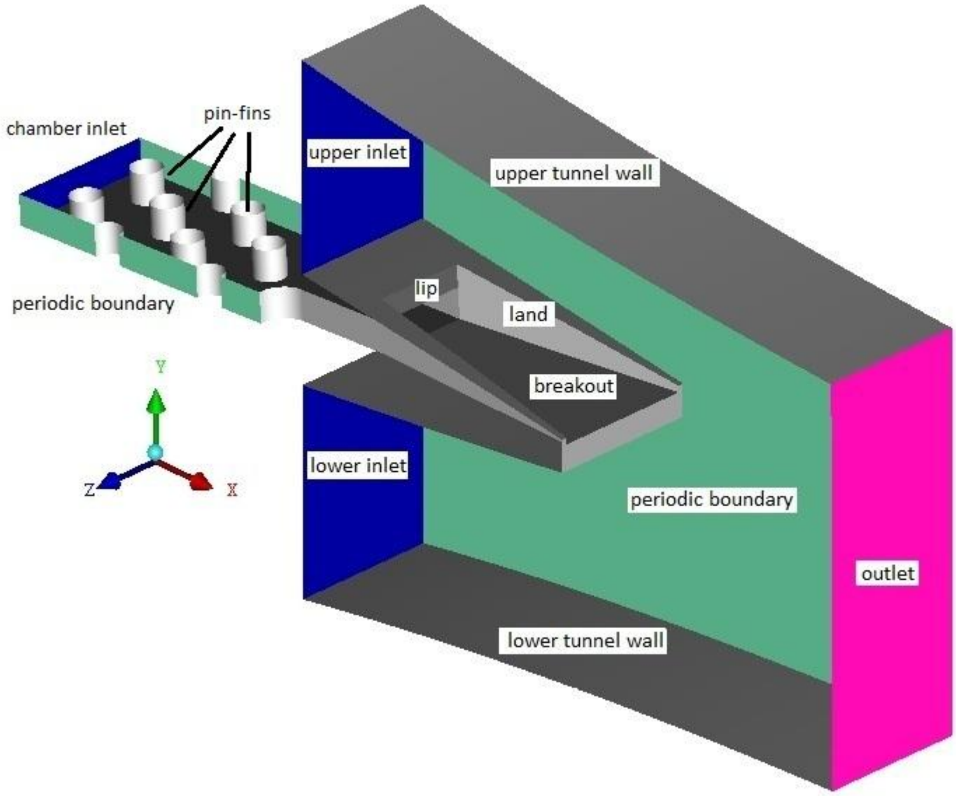


Figure 5-15 LES domain, geometry and boundary conditions for the cutback injection design with taper angle of 5°

At the upper and lower inlets of the LES domain, the velocity profiles need to be specified. These profiles are obtained from a RANS simulation of the full airfoil in the test section, as shown in Figure 5-16 below. In the full airfoil RANS domain, two cut-planes are created at $x = 230\ mm$ to obtain the main flow profiles. As the profiles are to be used by the LES of all the cutback configurations subsequently, to be unbiased, the airfoil is set to the original NACA0012 with the nominal chord length without any cutback modification. The span of the domain is set half a period of the LES domain. Symmetry boundaries conditions are applied on the two sides. To match the Reynolds number of 110,000 of the experiment, the "velocity-inlet" is prescribed at the inlet of the whole test section to set the x-velocity of $5.36\ m/s$. The k-ω SST is used as the turbulence model. The test section inlet turbulent intensity is set to 1% and the inlet turbulent viscosity ratio is set to 1. The inlet temperature is set to $288\,K$. At the outlet, the “pressure-outlet” is prescribed at the test section outlet to set the back pressure to be $1\,atm$. The adiabatic wall boundary condition is prescribed at the tunnel walls and the airfoil surfaces. The mesh for the entire test section consists of 3,264,000 hexagonal cells and the $y+$ on the walls is on the order of 1.

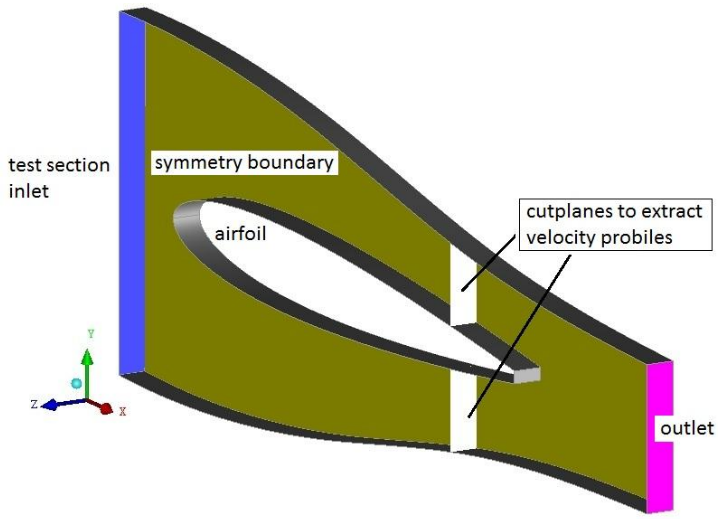


Figure 5-16 RANS domain of the full airfoil and the cut-planes for obtaining the velocity profiles for the LES main flow inlets.

Returning to the LES, the upper and lower inlets of the truncated test section are set as the “velocity inlet”, where the spanwise velocity is 0, and the x and y velocities are set from the velocity profiles obtained from the full airfoil RANS. The profiles are used for all the cutback configurations, as it is assumed that the influence of the cutback and trailing edge flow upstream towards the inlet is negligible. The velocity profiles are obtained from the mesh of the cutplanes of the RANS domain shown in Figure 5-16 above, and interpolated to the LES mesh of the inlets. The static temperatures at the inlets are set to 288K. The inlet turbulence modelling is a separate research issue. As no convenient treatment available, it is neglected in this study with “no-perturbation” option specified, but can be included in future research. For the present case, the amount of inlet turbulence will be negligible as compared to the large amount of the turbulence generated in the cutback, the chamber and the trailing edge. The turbulence in these regions can be self-initiated by the flow separation on the bluff-bodies. At the test section outlet, the back pressure is set to 1 atm with “average pressure specification” to reduce the wave reflection. For all the wall boundaries, the no-slip adiabatic condition is applied.

It is considered in this study that the coolant chamber needs to be included in the LES, especially the part that contains the pin-fin arrays, as the separated flow behind the pin-fins is a major source of the turbulence. As such, the “no-perturbation” option is specified at the chamber inlet. The chamber inlet is set as the “velocity-inlet” to prescribe the velocity of $10.97\ m/s$ normal to the boundary to match the blowing ratio of 1.3 in the experiment. Three arrays of pin-fins $5\ mm$ in diameter are included, arranged in the same way as the experiment shown previously in Figure 5-14(a) and Figure 5-15. The periodicity of the pin-fin arrangement is the same as that of the slot and land. Thus the chamber span is the same as the other parts of the LES domain, with periodic boundaries prescribed at the sides.

The temperature at the chamber inlet is set to 268K, 20K lower than the main flow. For the single-phase flow with low convective Mach number, the main effect of the temperature ratio is reflected in the density ratio between the mainstream and the coolant. As shown by the experiment of Taslim et al. (1992), the adiabatic effectiveness on the surface is insensitive to the density ratio. As the flow is incompressible, the temperature ratio should be set near 1 to be consistent with the density ratio of 1. But, on the other hand, the temperature difference should not be too small to be insignificant. Thus, the current temperature difference is chosen.

The LES mesh is generated using the ANSYS ICEM CFD package. The mesh for the main flow, cutback region and the coolant chamber are shown from Figure 5-17 to Figure 5-19. The mesh is a hybrid of hexagons cells and triangular prism cells. The triangular prism cells are used to fill the cutback region as the land height is getting thinner towards the trailing edge. In the chamber, the cells are all hexahedral. The mesh topology is a mixture of O-mesh around the pin-fins and the H-mesh filling the remaining space. In the main flow, the mesh is H-type. In total, 3,583,800 mesh cells are contained. The mesh is clustered near the surface so that the $y+$ is around 1. The mesh resolution in $x+$ and $z+$ is around 40 in the complex unsteady flow regions. Overall, the mesh resolution is at the same level as other LES studies on similar cutback injection configurations (e.g. Martini et al., 2005b; Joo and Durbin, 2009). Due to the amount of computational time, only temporal resolution is check with half-size time-step. Though not reported here, the predicted flow fields including the spanwise average adiabatic effectiveness on the breakout surface still remains the same.

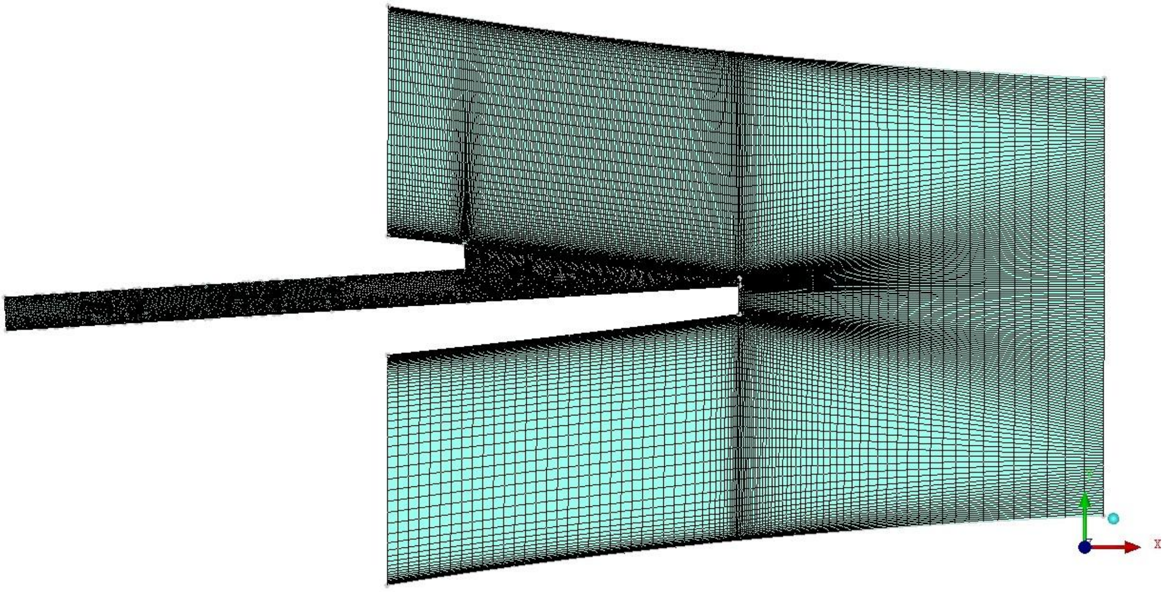


Figure 5-17 LES mesh for the main flow region

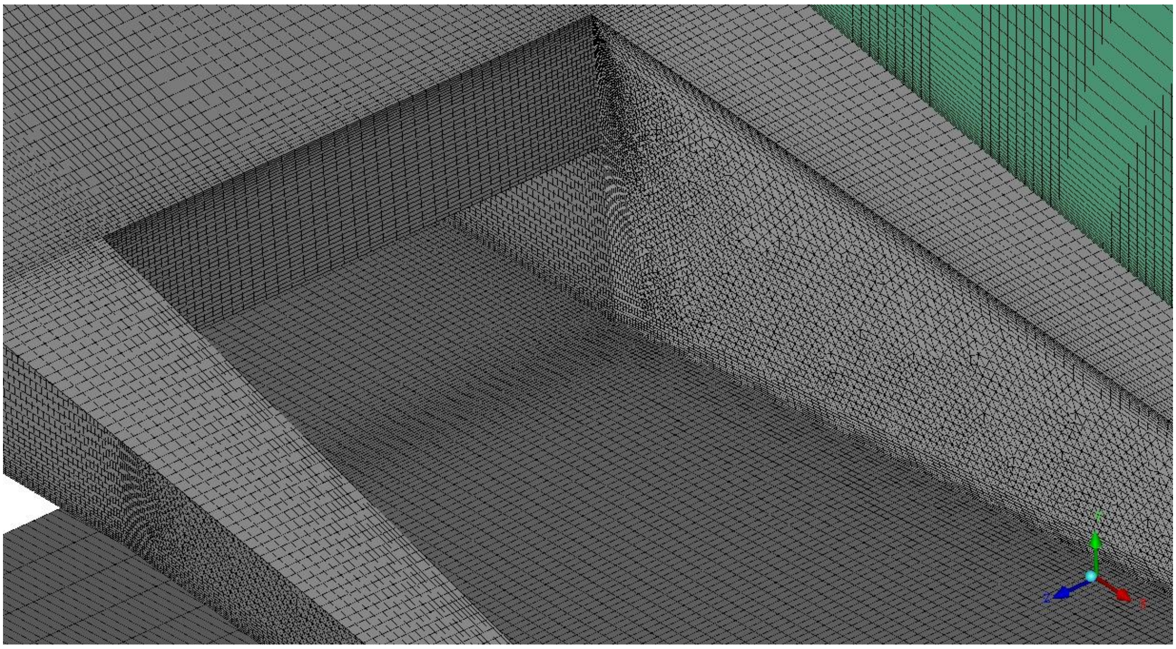


Figure 5-18 LES mesh for the cutback region

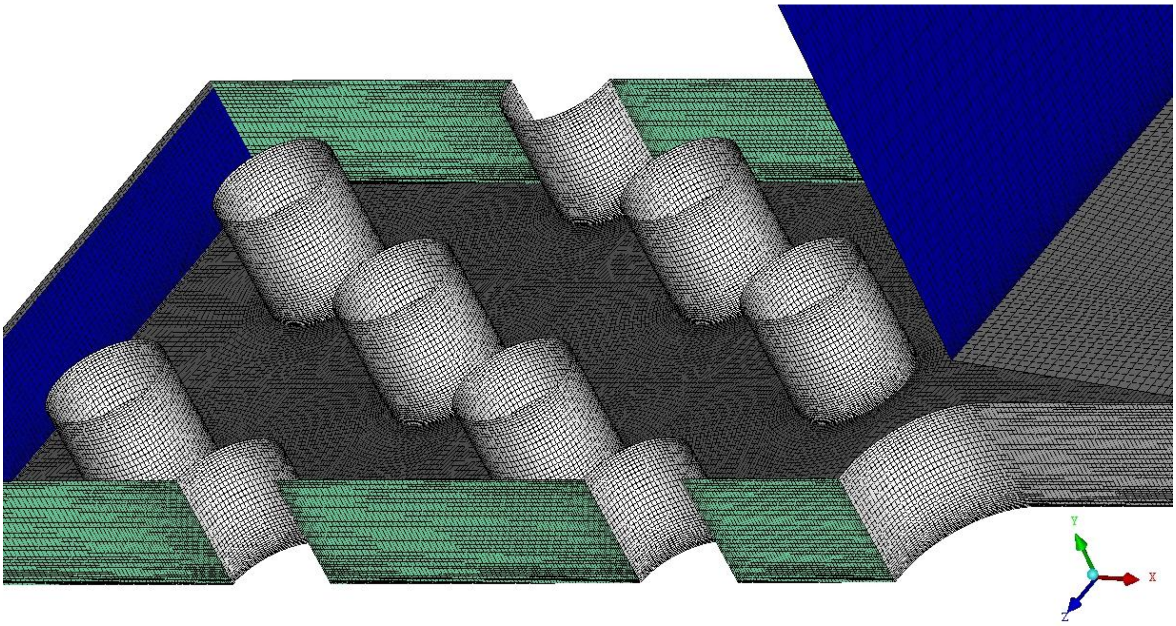


Figure 5-19 LES mesh for the coolant chamber

The spatial and temporal discretization methods are the same as the previous cylinder case. Based on the work of Sieverding and Heinemann (1990), the Strouhal number of the vortex shedding downstream of a flat plate with square trailing edge and turbulent boundary layers is around 0.21, which can be computed as:

$$St = \frac{f \cdot th}{U} = 0.21 \tag{5.6}$$

where th is the thickness of the slot lip. The velocity U takes the larger one of the two streams passing the lip, in this case the injection stream, obtained from the pre-LES RANS used for providing the initial solution for the LES. Based on the thickness of 3.7 mm and the coolant velocity at the slot of $15.7\ m/s$, the time step size is set as $\Delta t = 1.112 \times 10^{-5}$ s, resulting in about 100 time steps per shedding period. Compared to other LES studies on the similar configurations (e.g. Martini et al., 2005b; Joo and Durbin, 2009), the present time resolution is considered sufficient.

5.2.2 LES Validation Results

The initial solution for the LES is obtained by a pre-LES RANS solution, using the same LES domain and boundary conditions settings plus those specific to the turbulence model. The k-ω SST turbulence model is used. At the upper and lower inlets, the turbulence intensity is set to 1% and the turbulence viscosity ratio is set to 1. At the plenum inlet, the turbulence intensity is 5% and the turbulence viscosity ratio is 5.

The solution methods for the pre-LES RANS has been described in Chapter 3. The pressure-based solver with SIMPLEC pressure velocity coupling is used. The second-order upwind is used to discretize the momentum, energy and the two turbulence transport equations. The second-order interpolation is used for pressure interpolation, and LSCB is used for gradient computation.

The pre-LES RANS is run with 3000 iterations. By monitoring the solution residuals in Figure 5-20, it is observed that small fluctuation is presented. The residuals are given in Fluent Theory Guide, Section 28.15.1. As shown in Figure 5-21, there is fluctuation behind the bluff-body structures such as the slot lip and trailing edge, where physically the flow is highly unsteady. But little fluctuation of the RANS solution is observed around the pin-fins in the chamber, though not shown here.

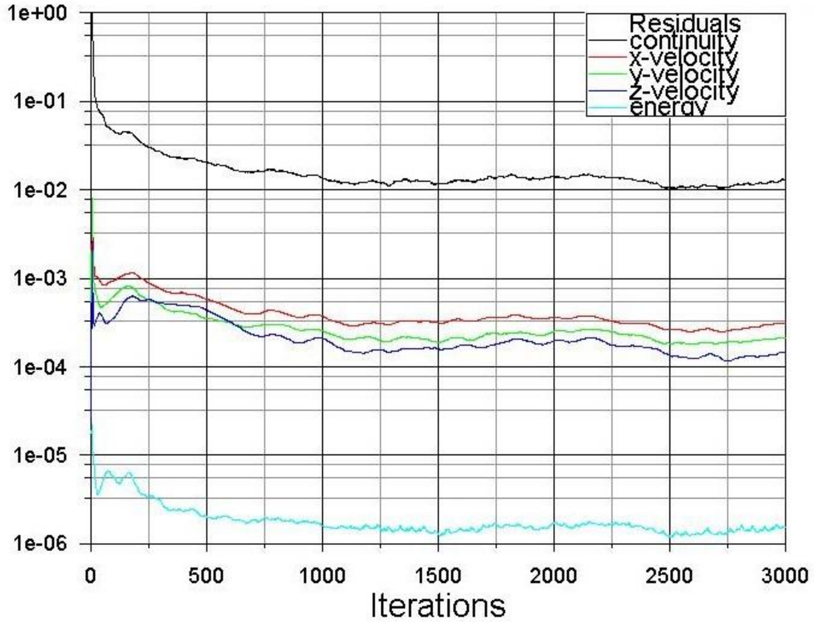


Figure 5-20 Solution residual of pre-LES RANS

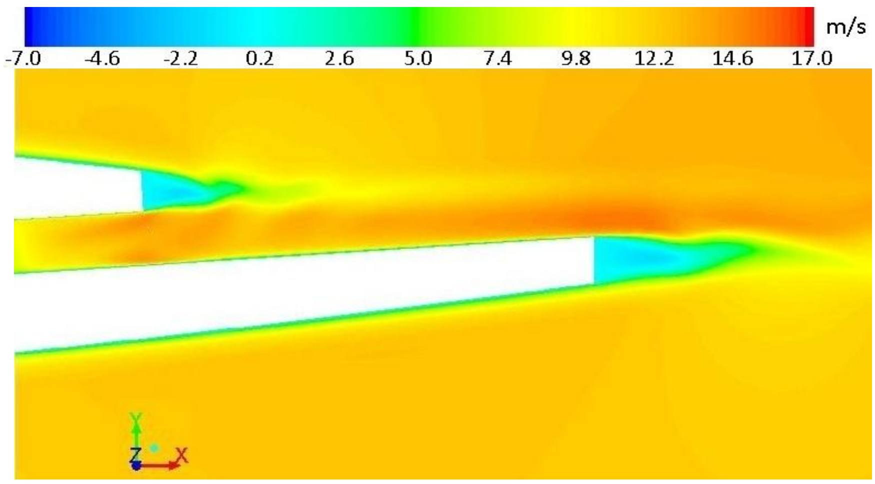


Figure 5-21 Fluctuation in the pre-LES RANS behind lips of the slot and trailing edge

After obtaining the initial solution, the transient stage of the LES is started until the turbulent flow field is statistically established, as indicated by the established pattern of the fluctuation of the temperature and pressure on the breakout surface, shown in Figure 5-22 and Figure 5-23. The establishment of the turbulent flow field takes 2×10^4 time steps, corresponding to about 24 domain through-flow times. After that, the statistic collection stage is run with 1.2×10^5 time steps, during which the solution is time-averaged with Equation (5.2). The variables that are time-averaged are the components of the velocity u_i, the static pressure p and the static temperature T.

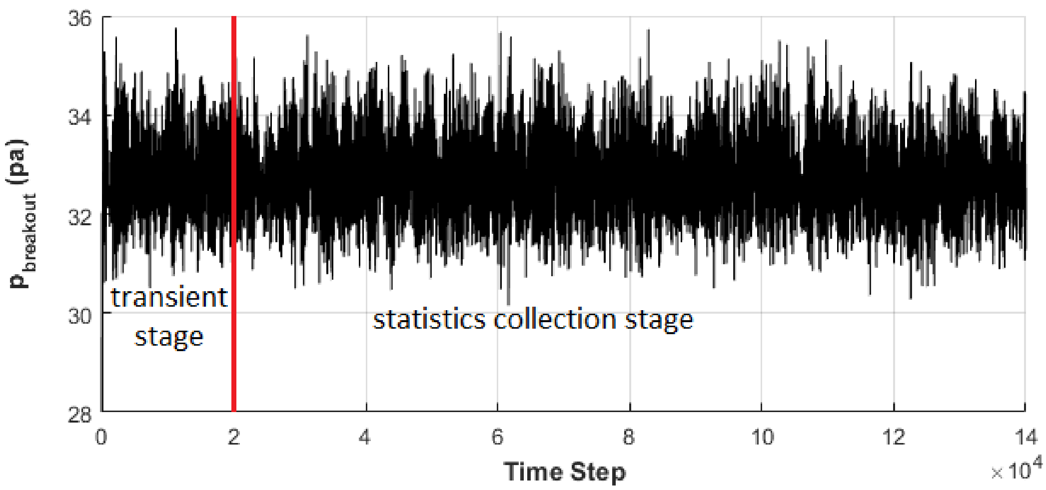


Figure 5-22 Fluctuation history of the average pressure on the breakout surface

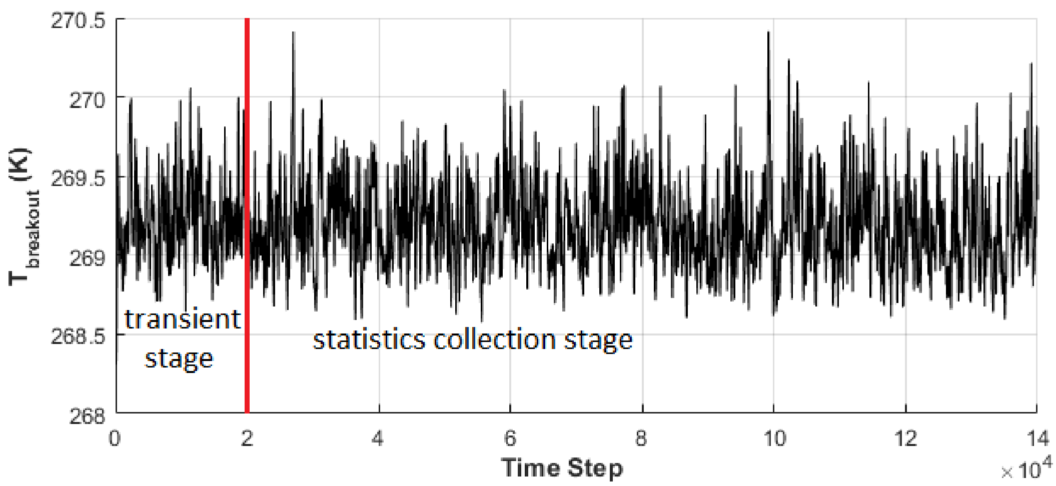


Figure 5-23 Fluctuation history of the average temperature on the breakout surface

Various aspects of the results are shown below. The instantaneous flow field at the statistics collection stage is shown in Figure 5-24 and Figure 5-25. It can be seen that the flow in the coolant chamber, the cutback region and the wake is highly unsteady and contains a wide range of turbulence scales. Figure 5-24 shows that the majority of the turbulence is generated by the pin-fins inside the chamber, the slot lip and the trailing edge lip. In Figure 5-25, vortex shedding can be seen downstream the slot lip and the trailing edge lip. The high amount of unsteadiness also explains the fluctuations in the pre-LES RANS.

Based on the time-mean LES solution, the adiabatic effectiveness η_{ad} on the trailing edge surface downstream of the slots is compared with the experiment, in terms of the surface distribution in Figure 5-26, and the spanwise-average in Figure 5-27. The comparisons show that the LES predicts the decline of the adiabatic effectiveness along the breakout surface, as also observed in the experiment. The present LES still shows less decline of η_{ad} on the breakout surface and less increase on the land surface. A possible reason could be due to no turbulence is prescribed at the main flow inlet of the LES domain in this study, leading to some under-prediction of turbulence in the cutback region. Prescription of inlet turbulence could make the present LES more accurate but is subject to future work.

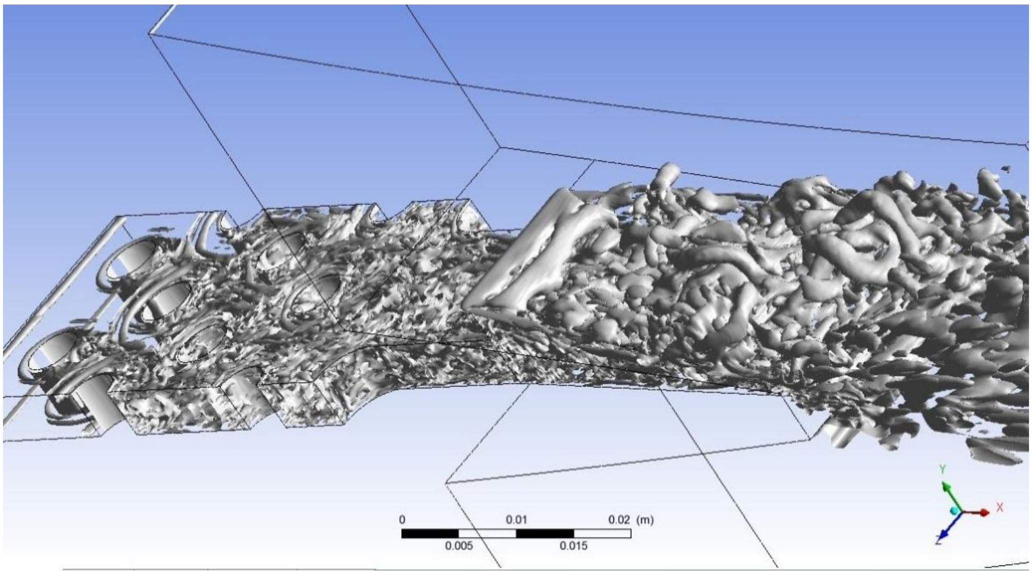


Figure 5-24 Instantaneous Q criterion of the cutback injection flow

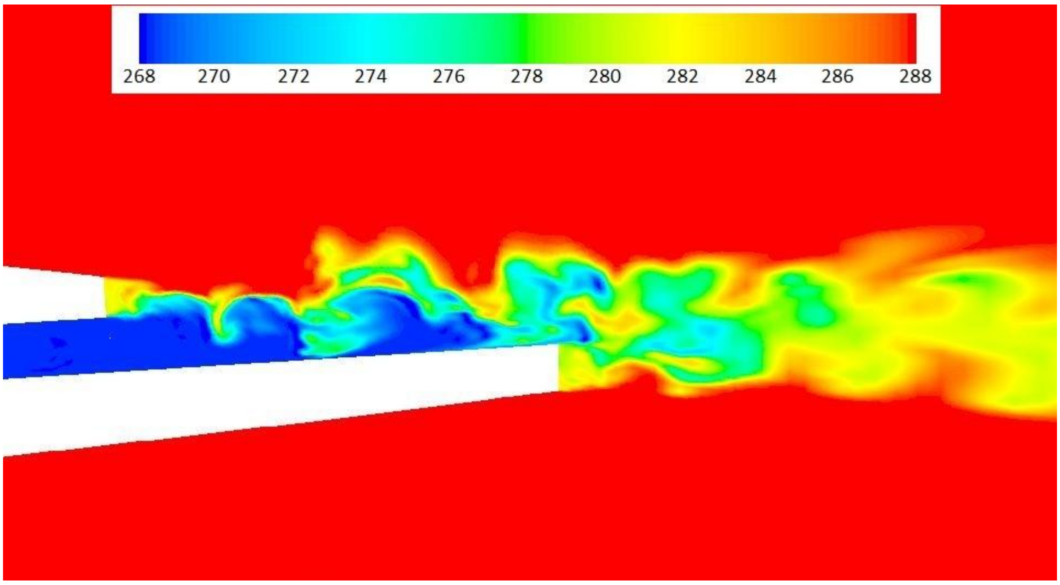
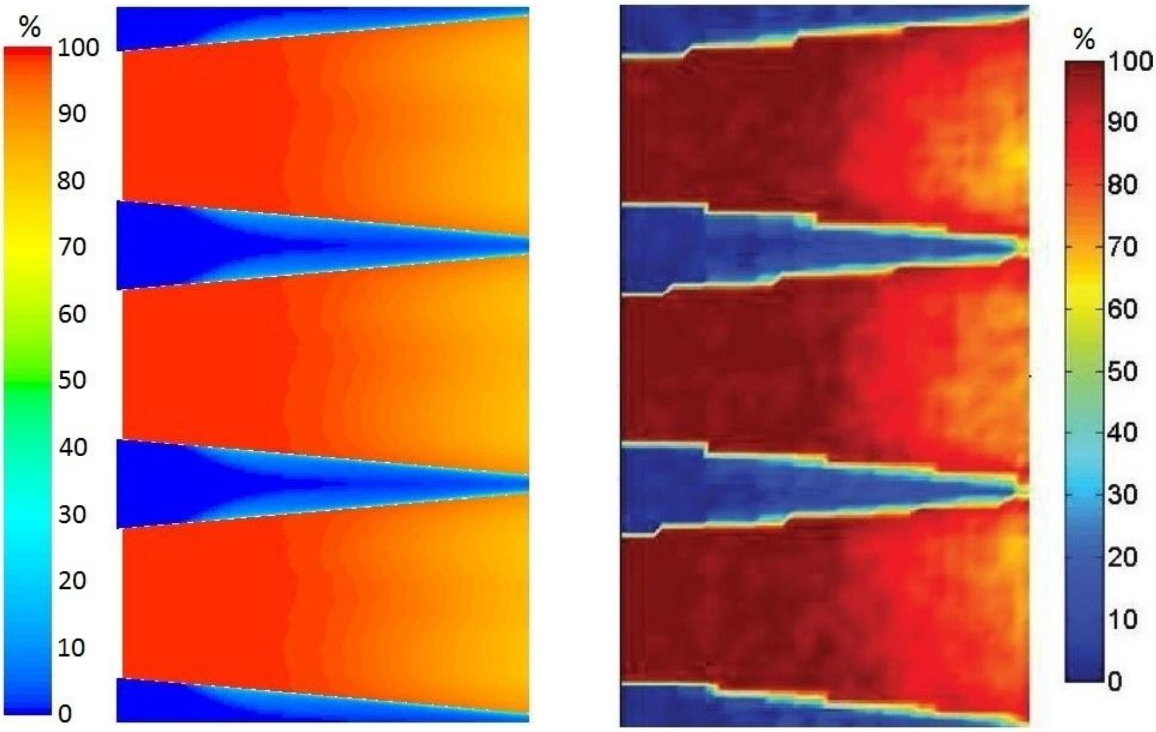


Figure 5-25 Instantaneous temperature at the mid-span cut-plane



(a) time-mean LES solution **(b) experiment (Ling et al., 2015)**

Figure 5-26 Adiabatic effectiveness η_{ad} (%) on the land top surface and on the breakout surface (flow from left to right)

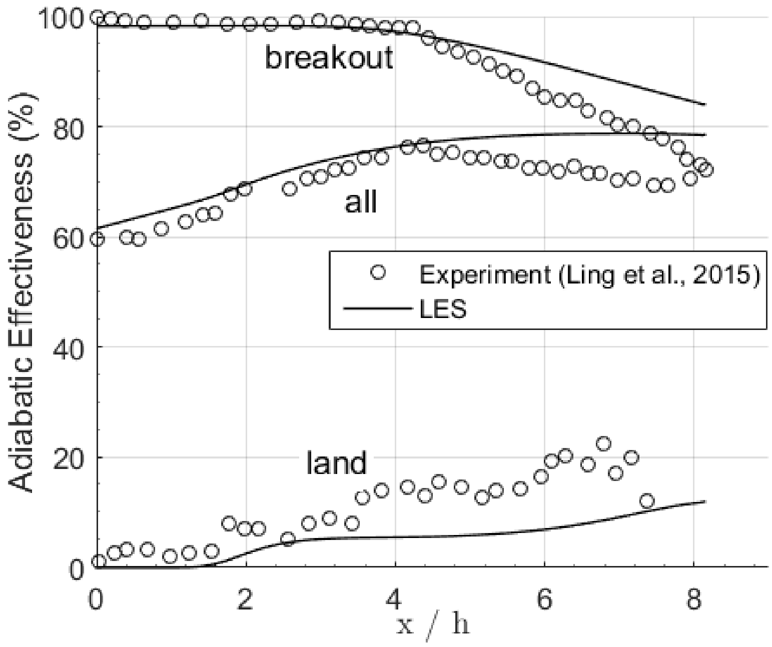
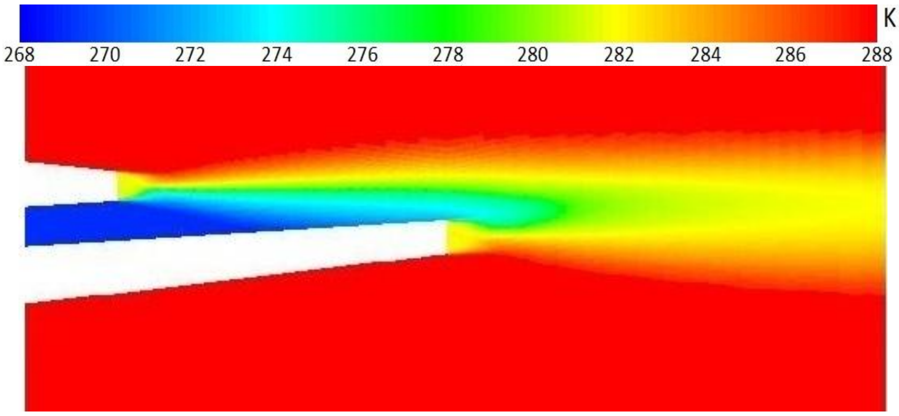


Figure 5-27 Adiabatic effectiveness η_{ad} downstream of the slot, spanwise-averaged over: the breakout surface, the land, the entire span; h: slot height

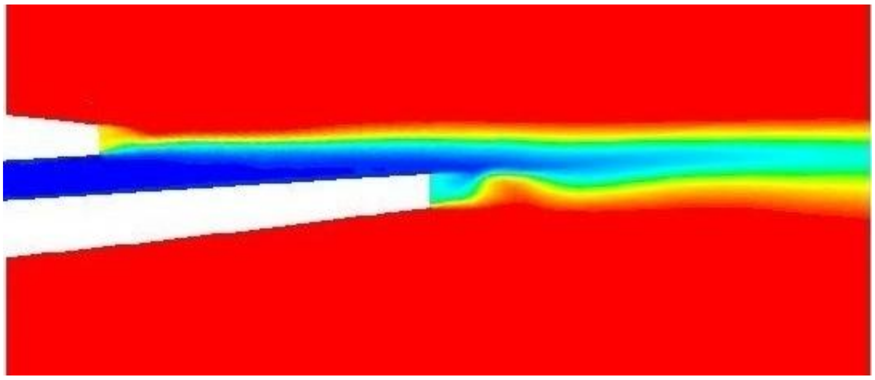
5.2.3 Comparison with RANS

The LES result is also compared with the RANS solution. The RANS computational settings including the domain and the boundary conditions are the same as the pre-LES RANS, but a much coarser mesh is used to be consistent with the normal RANS practices.

The RANS flow solutions are compared with the time-mean LES in various aspects. In the temperature in Figure 5-28, it shows that small fluctuations still exist in the RANS solution. The unsteady turbulent mixing between the main flow and the coolant also appears to be under-predicted by RANS, seen from the narrower and longer wake. As a result, Figure 5-29 and Figure 5-30 show that the adiabatic effectiveness does not decline and stays as high as near the slot on the entire surface. Overall, the RANS prediction deviates significantly from the experiment, while the LES prediction is qualitatively much more accurate than the RANS.



(a) time-mean LES



(b) RANS with k-ω SST turbulence model

Figure 5-28 Temperature prediction on the midspan cutplane, time-mean LES versus RANS

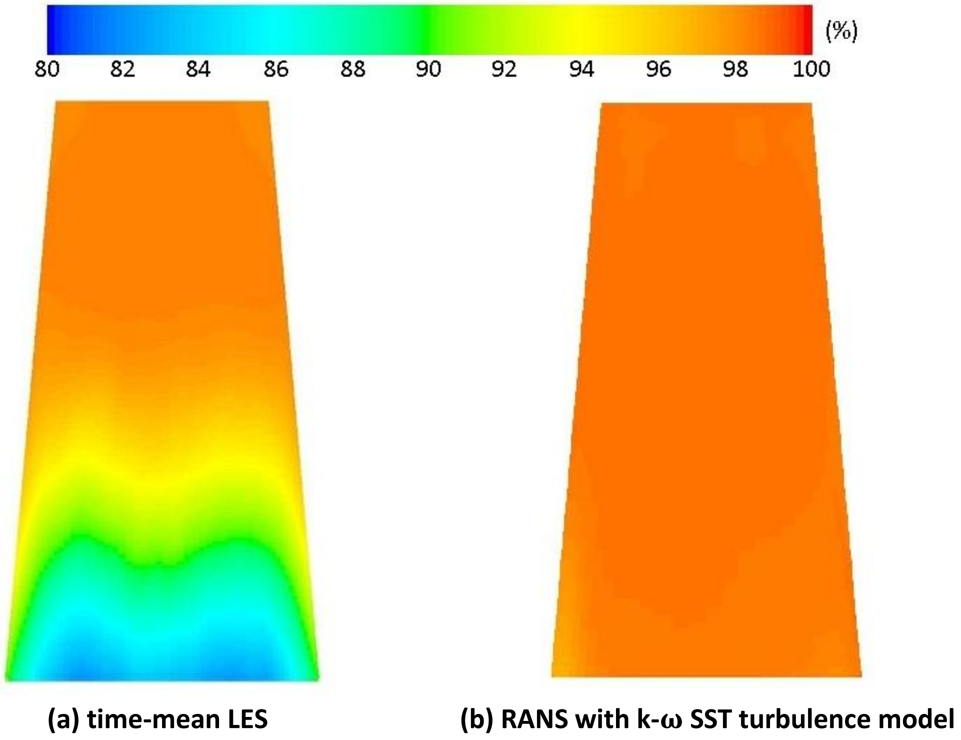


Figure 5-29 Adiabatic effectiveness on the breakout surface (flow from top to buttom)

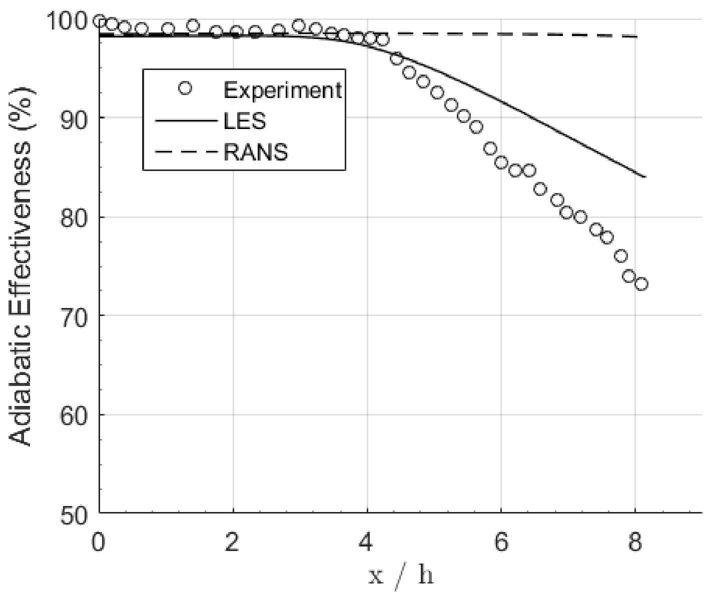


Figure 5-30 Spanwise-averaged adiabatic effectiveness on the breakout surface, comparison among LES, RANS and the experiment (Ling et al., 2015).

5.3 Summary

This chapter presents a baseline LES validation for the solver for pure aerodynamic prediction and an aero-thermal prediction. The large scale complex turbulent unsteadiness is revealed by the LES. As most of the turbulence is resolved, the LES solutions match well with the experiment.

The LES results are also compared with the RANS prediction on the trailing edge cutback injection. Significant difference is observed in the temperature distribution in the cutback region, and the wake, in which the physical unsteadiness mostly occurs. The LES agrees with the experiment in capturing the decrease of the adiabatic effectiveness on the breakout surface away from the slot. But the adiabatic effectiveness from RANS stays almost constant and is significantly over-predicted. The over-prediction is considered due to the under-prediction of the turbulent mixing in the cutback region by the RANS turbulence models, as also mentioned by other researchers (e.g. Ling et al., 2014). In addition, the RANS is unable to get a fully converged steady-state solution. Overall, the LES solution agrees well with the experiment but the RANS solution deviates qualitatively different from the experiment. This shows that the LES is required to accurately evaluate the flow control performance in the design optimization.

Chapter 6

An Efficient LES Method for Design Optimization

– Part I: Concept and Methodology for a 'Steady LES' Solution

6.1 Overall Strategy

Previous chapter has shown that very complicated flow physics may occur in a flow control related flow field which can be predicted accurately by the LES. It has also been demonstrated that the RANS predictions are not sufficiently accurate for this type of flows. These thus point to the need of using the LES in evaluating the flow control for design optimization.

However, a problem that arises is the conflict between the large computational cost and the efficiency required due to large number of performance evaluations. To address this problem, the present study aims to developing a new method to make the LES-based performance evaluation more efficient for the design optimization, while maintaining the prediction accuracy of the LES.

As the time-dependent time-mean performance and flow field are of interest in this study, the basic intent is to consider the possibility of using the steady-state simulation to predict the time-mean LES solution. Based on previous efforts in efficient analysis of the unsteady flows (e.g. Ning and He, 2001; Galbraith and Orkwis, 2008; see Chapter 2), it is considered that the key would be to model the turbulent fluctuation correlation terms (e.g. the Reynolds stresses) correctly and put them as the source terms in the steady-state equations. However, to compute the source terms correctly, essential information from the real LES may still be required beforehand. Nevertheless, it is hoped that eventually most of the designs can be evaluated by the steady-state simulation with solutions as accurately as the LES, but with only a small number of LES runs required.

Based on this general idea, the development of the efficient LES method for the design optimization can be broken down into two parts. In Part I, a small number of sample designs in the design space are chosen to be evaluated by the LES and compute the source terms. For each of these LES-evaluated designs, the computed source terms for the steady-state equations, should make the steady-state solution become the time-mean LES solution of the design itself. Then in Part II, instead of being only useful for reproducing the obtained time-mean LES solution, the source terms should be propagated from the LES-evaluated designs to the other designs, so that the efficient LES method can be effectively useful for the design optimization. Eventually, except for the few LES-evaluated sample designs, it is hope that all the remaining designs in the optimization can be evaluated by the steady-state simulations, but the steady-state solution can be as accurate as the time-mean LES solutions, if performed, on these remaining designs.

In this chapter, the development of Part I of this novel efficient LES method will be discussed, which is to compute the source terms for the steady-state equations based on the time-mean LES solution of a design, and use the source terms to reproduce the time-mean LES solution of that design through the steady-state simulation.

6.2 Steady-State Equations for the Time-Mean LES Solution

To compute the source terms of the steady-state equations and drive the steady-state solution at the time-mean LES solution, the method described below is similar to the method used by He (2008) and Lukovic et al. (2006).

First, by dropping the time-derivative terms of the unsteady Navier-Stokes (NS) equations (Equation (**3.10)** and (**3.11)** in Chapter 3), and add the source terms on the right hand side, one obtains the steady-state equations with the source terms as:

$$R_m(\boldsymbol{Q}) \equiv \frac{\partial(\rho u_j)}{\partial x_j} = S_m \tag{6.1}$$

$$R_i(\boldsymbol{Q}) \equiv \frac{\partial(\rho u_i u_j)}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} = S_i \tag{6.2}$$

$$R_e(\boldsymbol{Q}) \equiv \frac{\partial(\rho E + p)u_j}{\partial x_j} + \kappa \frac{\partial T}{\partial x_j \partial x_j} - \frac{\partial(u_i \tau_{ij})}{\partial x_j} = S_e \tag{6.3}$$

In the above equations, the left hand side of each equation is also defined as $R(\boldsymbol{Q})$ in this work. It does not contain any terms of turbulence models, and can be solved when choosing the “laminar” viscous option in Fluent. Also, denote the primitive solution variables as $\boldsymbol{Q} = [p\ \ u_1\ \ u_2\ \ u_3\ \ T]^T$. Hence the time-mean LES solution to be reproduced later can be denoted as $\boldsymbol{Q}^* = [\bar{p}\ \ \tilde{u}_1\ \ \tilde{u}_2\ \ \tilde{u}_3\ \ \tilde{T}]^T$, in which the Favre-averaging naturally becomes time-averaging for the incompressible flow.

To balance the steady-state equations at the time-mean LES solution $\boldsymbol{Q}^*$, the desired source terms $\boldsymbol{S}^* = [S_m^*\ \ S_x^*\ \ S_y^*\ \ S_z^*\ \ S_e^*]$ should be equal to $\boldsymbol{R}(\cdot)$ evaluated at $\boldsymbol{Q}^*$:

$$\boldsymbol{S}^* = \boldsymbol{R}(\boldsymbol{Q}^*) \tag{6.4}$$

where $\boldsymbol{R} = [R_m\ \ R_x\ \ R_y\ \ R_z\ \ R_e]$. The meaning of the desired source terms can be seen by time-averaging the unsteady NS equations. Thus, $\boldsymbol{R}(\boldsymbol{Q}^*)$ (or $\boldsymbol{S}^*$ equivalently) represents the Reynolds stresses in the momentum equations and the turbulent stresses in energy equation, for the time-averaged set of NS equations. In implementation, $\boldsymbol{R}(\boldsymbol{Q}^*)$ is evaluated by discretization (such as finite volume method), and the computed $\boldsymbol{Q}^*$ cannot be the exactly the time-mean flow (due to infinite time steps required). As such, $\boldsymbol{S}^*$ (or $\boldsymbol{R}(\boldsymbol{Q}^*)$ equivalently) represents the summation of the Reynolds stresses (or turbulent stresses), plus the numerical discretization error and the error between $\boldsymbol{Q}^*$ and the true time-mean flow. As such S_m^* will also takes some small values. These errors are however should be small compared to the Reynolds stresses and the turbulent stresses. Nonetheless, no matter how $\boldsymbol{R}(\boldsymbol{Q}^*)$ is evaluated, the desired source terms $\boldsymbol{S}^*$ should be equal to $\boldsymbol{R}(\boldsymbol{Q}^*)$.

The following content presents an efficient method to compute $\boldsymbol{R}(\boldsymbol{Q}^*)$, which is independent of how $\boldsymbol{R}(\cdot)$ is discretized by the solver. First, drop the source terms in the steady-state equations (6.1) to (6.3), then choose the explicit Euler pseudo-time marching, then initialize the solution as $\boldsymbol{Q}^*$, and then iterate the steady-state equations by only one pseudo-time step $\Delta\tau$. As such, the $\boldsymbol{R}(\boldsymbol{Q}^*)$ (and hence the desired source terms $\boldsymbol{S}^*$) can be obtained by:

$$\boldsymbol{S}^* = \boldsymbol{R}(\boldsymbol{Q}^*) = -\frac{\boldsymbol{U}^1 - \boldsymbol{U}^*}{\Delta\tau} \tag{6.5}$$

where $\boldsymbol{U}^1$ is the conservation form of the flow solution $\boldsymbol{Q}^1$ after the pseudo-time step. $\boldsymbol{U}^*$ is the conservation form of $\boldsymbol{Q}^*$:

$$\boldsymbol{U}^* = \left[\bar{\rho}\;\; \bar{\rho}\tilde{u}_1\;\; \bar{\rho}\tilde{u}_2\;\; \bar{\rho}\tilde{u}_3\;\; \bar{\rho}\left(\frac{1}{2}\tilde{u}_k\tilde{u}_k + c_V\bar{T}\right)\right]^T \tag{6.6}$$

6.3 Implementation Issues in Fluent

In practice, however, two issues may prevent the direct calculation of the source terms with Equation (6.5).

The first is the lack of the explicit Euler marching scheme. In Fluent, the only type of solver with explicit pseudo-time marching compatible for both the incompressible flow and the compressible flow is the density-based steady-state solver. The solution is marched using the 3-stage Runge-Kutta (RK) method. Thus $\boldsymbol{R}(\boldsymbol{Q}^*)$ cannot be obtained directly with Equation (6.5). To explain this, the steady-state equation solved by Fluent is given by:

$$\boldsymbol{\Gamma}\frac{\partial}{\partial\tau}\int \boldsymbol{Q}\,dV + \int \boldsymbol{R}(\boldsymbol{Q})\,dV = \int \boldsymbol{S}\,dV \tag{6.7}$$

where $\boldsymbol{\Gamma}$ is the precondition matrix, see (ANSYS Fluent Theory Guide) for details.

The 3-stage Runge-Kutta pseudo-time marching is expressed as:

$$\begin{aligned} &\boldsymbol{Q}_{RK}^{0} = \boldsymbol{Q}^{*} \\ &\boldsymbol{Q}_{RK}^{i} - \boldsymbol{Q}^{*} = -\alpha_i\,\Delta\tau\,\boldsymbol{\Gamma}^{-1}\boldsymbol{R}\left(\boldsymbol{Q}_{RK}^{i-1}\right)\Delta V \\ &\boldsymbol{Q}^{1} = \boldsymbol{Q}_{RK}^{3} \end{aligned} \tag{6.8}$$

in which, $i = 1, 2, 3,$ and $\boldsymbol{Q}^{*}$ and $\boldsymbol{Q}^{1}$ are the solutions at the current and the next pseudo-time level. $\boldsymbol{Q}_{RK}^{i}$ is the intermediate solution of the i-th Runge-Kutta stage. α_i is the coefficient for the i-th stage. $\Delta\tau$ is the pseudo-time step of the cell computed based-on the Courant–Friedrichs–Lewy (CFL) condition:

$$\Delta\tau = \frac{2\,CFL\,\Delta V}{\sum_f \lambda_f^{max}\,A_f} \tag{6.9}$$

where ΔV is the cell volume, A_f is the area of the cell face, λ_f^{max} is the maximum of the local eigenvalue of the preconditioned system of flow equations (Equation (6.7), see ANSYS Fluent Theory Guide for detail). A CFL number less than 1 is set by the user.

It can be seen that the solution change $\boldsymbol{Q}^{1} - \boldsymbol{Q}^{*}$ is mixed with the effect of both $\boldsymbol{R}(\boldsymbol{Q}^{*})$ and $\boldsymbol{R}\left(\boldsymbol{Q}^{i}\right)$. As the intermediate solution $\boldsymbol{Q}_{RK}^{1}$ cannot be obtained in Fluent, $\boldsymbol{R}(\boldsymbol{Q}^{*})$ and hence $\boldsymbol{S}^{*}$ cannot be obtained directly with Equation (6.5).

The second issue is, as the flow model of $\boldsymbol{R}(\boldsymbol{Q}^{*})$ is laminar, even if adding the source terms shows to stabilize the steady-state solution to some extent, it is observed that small fluctuation would still presents in the iteration for 3D configurations after the flow solution residual has dropped by one or two orders of magnitude. This prevents the solution from convergence and shows significant difference from $\boldsymbol{Q}^{*}$. It thus appears that some additional damping is required to bring the steady-state solution to convergence.

6.3.1 The Source Term Correction Iteration

For concise discussion, let $R(\cdot)$ and S denote the left hand side and the source term of one of the flow equations in Equations (6.1) to (6.3). Since $R(\boldsymbol{Q}^*)$ cannot be obtained directly, a procedure is developed to correct the source term iteratively and make it converge to $R(\boldsymbol{Q}^*)$. The correction procedure is shown in Figure 6-1 below, and each correction is expressed as:

$$\Delta S = \alpha \cdot \frac{U^* - U^1}{\Delta\tau} \tag{6.10}$$

in which α is the under-relaxation for the correction.

To explain the procedure, the source term S is first initialized (e.g. simply setting $S^0 = 0$). At the same time, the flow solution is initialized with $\boldsymbol{Q}^*$. If the source term equals $R(\boldsymbol{Q}^*)$, the solution shall stay the same after being marched by one pseudo-time step. However, as this is impossible for the initial source term, the solution change $\frac{U^1 - U^*}{\Delta\tau}$ will be large if the steady-state flow equations with the initial source terms (stay unchanged in the march) are marched by one pseudo-time step. Hence, the source term needs to be corrected. Ideally, if the explicit Euler marching is available, one can set the under-relaxation as $\alpha = 1$, so that the source term can be corrected only once to be equal to $R(\boldsymbol{Q}^*)$, that is:

$$S^* = S^0 + \Delta S = S^0 + \frac{U^* - U^1}{\Delta\tau} \tag{6.11}$$

since for the explicit Euler marching,

$$\frac{U^* - U^1}{\Delta\tau} = R(\boldsymbol{Q}^*) - S^0 \tag{6.12}$$

However, as the Runge-Kutta scheme is the only explicit marching scheme in Fluent, the source term corrected by Equation (6.11) above will not equal to $R(\boldsymbol{Q}^*)$. However, one may still assume that the source term shall approach closer to $R(\boldsymbol{Q}^*)$ each time it is corrected. One can check how close to $\boldsymbol{R}(\boldsymbol{Q}^*)$ the corrected source terms are by putting the them into the steady-state flow equations, re-initialize the flow solution with $\boldsymbol{Q}^*$ and march the flow equations (inputted source terms stay unchanged in the march) by one pseudo-time step. If the change of the flow solution $\frac{U^1-U^*}{\Delta\tau}$ is still not small, it means the source term is not yet close enough to $R(\boldsymbol{Q}^*)$ and should be corrected again with Equation (6.11). If the solution change is smaller than a certain tolerance, the source term is considered to have converged to $R(\boldsymbol{Q}^*)$ and is the desired source term S^* to balance the corresponding steady-state equation at solution $\boldsymbol{Q}^*$. Note that the source term convergence to $R(\boldsymbol{Q}^*)$ is required for every flow equation.

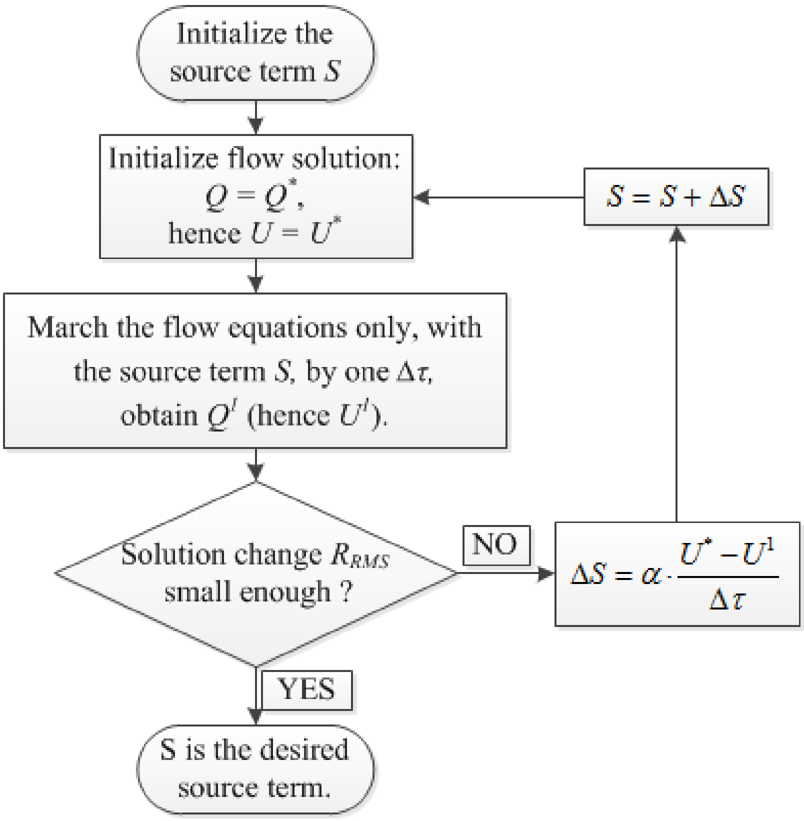


Figure 6-1 The source term correction iteration

To ensure the source term convergences to $R(\boldsymbol{Q}^*)$, the under-relaxation α needs to be chosen to ensure that the solution change becomes smaller at each source term correction.

Take any one of the steady-state flow equations for illustration, define the solution change after the m-th source term correction as:

$$D^m \equiv \frac{U^{1,m} - U^*}{\Delta\tau} = k^m(-R(\boldsymbol{Q}^*) + S^m) \tag{6.13}$$

where $k^m \neq 1$ is used to denote the difference between D^m and $(-R(\boldsymbol{Q}^*) + S^m)$ if the explicit Euler marching is not used. To converge the source term, the absolute value of D^{m+1} needs to be smaller than that of D^m. To avoid oscillation, D^{m+1} and D^m shall have the same sign. Since for the flow equation:

$$\begin{aligned} D^{m+1} &= k^{m+1}(-R(\boldsymbol{Q}^*) + S^m + \Delta S^{m+1}) = k^{m+1}\left(\frac{U^{1,m} - U^*}{k^m\Delta\tau} + \alpha \cdot \frac{U^* - U^{1,m}}{\Delta\tau}\right) \\ &= \left(\frac{k^{m+1}}{k^m} - \alpha \cdot k^{m+1}\right) D^m \end{aligned} \tag{6.14}$$

To converge the source term as well as to avoid oscillation, it is required that:

$$0 < \frac{D^{m+1}}{D^m} < 1 \tag{6.15}$$

Thus, there is:

$$0 < \frac{k^{m+1}}{k^m} - \alpha \cdot k^{m+1} < 1 \tag{6.16}$$

There the under-relaxation factor α needs to be set in the range of:

$$\frac{1}{k^m} - \frac{1}{k^{m+1}} < \alpha < \frac{1}{k^{m+1}} \tag{6.17}$$

Generally it is reasonable to assume that k^m is close to 1 and does not change much with m. So practically α can be set between 0 and 1, with slight tuning if necessary to get the source term correction iteration converge.

To monitor the convergence of the source term, one can chose to monitor the decrease of the root-mean-square (RMS) of the solution change over the entire domain across a pseudo-time step. For each flow equation, the RMS solution change is defined as:

$$R_{RMS} = \sqrt{\frac{1}{N_{cell}} \sum_{c}^{N_{cell}} \left(\frac{U^{1,m} - U^{*}}{\Delta\tau} \right)^2} \quad \textbf{(6.18)}$$

where N_{cell} is the number of mesh cells. For the continuity equation, we let $U = \rho$ for the compressible flow and $U = p/U_r^2$ for the incompressible flow, with the reference velocity U_r chosen by the user (such as the freestream velocity). In this thesis, the source term correction is considered converged when the RMS change is reduced by 4 orders of magnitude for each flow equation. This typically requires 2000 corrections, requiring the wall clock time of 6 hours with ten 2.4 GHz physical processors on a desktop workstation. From the procedure explain above one can see that the computational cost for each source term correction step is approximately the same as an explicit pseudo-time marching of the flow equations.

6.3.2 Stabilizing the Steady-State Solution

In theory, after the source terms iterations have converged to $\boldsymbol{R}(\boldsymbol{Q}^*)$, they should balance the laminar steady-state equations (6.1) to (6.3) at the time-mean LES solution $\boldsymbol{Q}^*$. However in practice, it has been observed that the laminar steady-state solution is often unable to fully converge, despite the amount of stabilization provided by the source terms.

Thus, a method to provide additional damping against the fluctuation is devised. Inspired from the ability of the turbulence model in getting the steady-state solution to converge in many physically unsteady turbulent flows, it is considered that the turbulence model could be used to provide the damping. Due to the relative simplicity for implementation, the Spalart-Allmaras (SA) model (Spalart and Allmaras, 1992) is chosen. The eddy viscosity of the model then serves as the damping viscosity μ_d to provide the remaining stabilization in addition to that from the source terms.

It should be emphasized that, the damping viscosity μ_d is used here only as a numerical damper, rather than a model for the turbulence. Thus, its detail value is not important, as long as it is a reasonable distribution (e.g. being zero in the freestream and laminar regions) that will not negatively affect the steady-state convergence.

Putting the numerical damping treatment into the laminar steady-state equations (6.1) to (6.3), the modified steady-state flow equations become:

$$R_{m,d}(\boldsymbol{Q}) \equiv R_m(\boldsymbol{Q}) = S_m \tag{6.19}$$

$$R_{i,d}(\boldsymbol{Q}, \mu_d) \equiv R_i(\boldsymbol{Q}) - \frac{\partial}{\partial x_j}\left(2\mu_d \cdot S_{v,ij}\right) = S_i \tag{6.20}$$

$$R_{e,d}(\boldsymbol{Q}, \mu_d) \equiv R_e(\boldsymbol{Q}) - \frac{\partial}{\partial x_j}\left(2\mu_d u_i S_{v,ij}\right) - \frac{\partial}{\partial x_j}\left(\frac{\mu_d}{Pr}\frac{\partial\left(c_p T\right)}{\partial x_j}\right) = S_e \tag{6.21}$$

The new left hand side is defined as $\boldsymbol{R}_d(\boldsymbol{Q}, \mu_d)$ to account for the terms of the damping viscosity μ_d. To balance the equations at $\boldsymbol{Q}^*$, the source term now needs to converge to $\boldsymbol{R}_d(\boldsymbol{Q}^*, \mu_d)$.

The $\boldsymbol{R}_d(\boldsymbol{Q}^*, \mu_d)$ needs to be determined to become the target for the convergence of the source term iteration. As such, the μ_d needs to be computed first. It can be computed by only solving the SA model transport equation, using $\boldsymbol{Q}^*$ for the coefficients of the transport equation. This is expressed as:

$$\frac{\partial\left(\bar{\rho}\tilde{u}_j \nu_d\right)}{\partial x_j} = c_{b1} S_{sa} \bar{\rho} \nu_d - c_{w1} f_w \bar{\rho}\left(\frac{\nu_d}{d}\right)^2 + \frac{1}{\sigma}\frac{\partial}{\partial x_k}\left[\left(\mu + \bar{\rho}\nu_d\right)\frac{\partial \nu_d}{\partial x_k}\right] + \frac{c_{b2}}{\sigma}\bar{\rho}\frac{\partial \nu_d}{\partial x_k}\frac{\partial \nu_d}{\partial x_k} \tag{6.22}$$

and

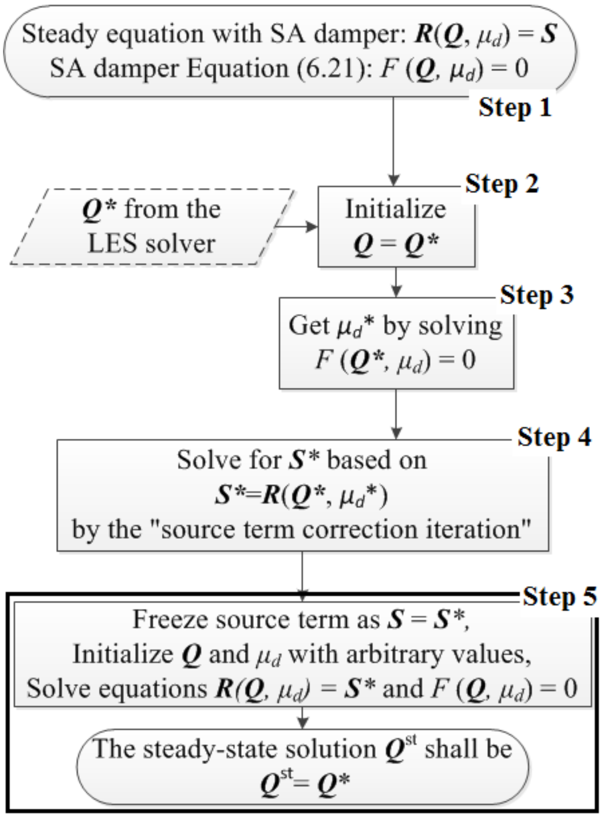
$$\mu_d = \bar{\rho}\nu_d f_{v1} \tag{6.23}$$

in which $\boldsymbol{Q}^*$ is used for $\bar{\rho}$, $\tilde{u}_j$ and the terms that dependent on the mean flow quantities, such as S_{sa}, f_w and f_{v1}. The other quantities are the model constants and fluid properties. The boundary condition for the SA transport equation (6.22) can be set to some reasonable values (e.g. turbulent viscosity ratio equals 10 at the inlet), but needs to be the same for all the designs in the optimization, as discussed in Chapter 7. Once μ_d is computed, it is fixed hence $\boldsymbol{R}_d(\boldsymbol{Q}^*, \mu_d)$ is fixed. The numerical damping procedure using the SA turbulence model as worked out in the above procedure can be denoted as the 'SA damper model'.

After that, the source term is still calculated using the iterative procedure shown before in Figure 6-1, except that $\boldsymbol{R}_d(\boldsymbol{Q}^*, \mu_d)$ is now the target for convergence. During the source term iteration the SA transport equation should be deactivated to freeze the μ_d. After the source term has converged to $\boldsymbol{R}_d(\boldsymbol{Q}^*, \mu_d)$, it is put into the steady-state solver coupled with the SA damper model. The flow and the damping viscosity are initialized with some reasonable value. The steady-state flow equations and the SA transport equation are solved together. As such, the steady-state solution in the end should be able to converge to the time-averaged LES solution $\boldsymbol{Q}^*$, and should be independent of μ_d

6.4 Summary of the Part I Procedure

The summary of the key steps in implementing the Part I of the efficient LES method is explained in Figure 6-2 below. Assume that the time-mean LES solution $\boldsymbol{Q}^*$ has been obtained from a separate LES solver.



Step 1:
Choose the density-based explicit steady-state solver. Choose the SA turbulence model, set a reasonable boundary condition for the SA transport equation.

Step 2:
Initialize the solution with $\boldsymbol{Q}^*$.

Step 3:
Initialize μ_d with a reasonable solution. Deactivate the flow equations to freeze the flow at $\boldsymbol{Q}^*$. Solve the SA transport equation to obtain μ_d^*.

Step 4:
Deactivate the SA transport equation to hold μ_d^*. Perform the iterative procedure in Figure 6-1 to make the source terms converge to $\boldsymbol{R}_d(\boldsymbol{Q}^*, \mu_d^*)$.

Step 5:
Set $\boldsymbol{S} = \boldsymbol{S}^*$ in the steady-state equations $\boldsymbol{R}_d(\boldsymbol{Q}, \mu_d)$. Initialize $\boldsymbol{Q}$ and μ_d with some values. Solve the flow equations and the SA transport equation at the same time to obtain the steady-state solution $\boldsymbol{Q}^{\mathrm{st}} = \boldsymbol{Q}^*$.

Figure 6-2 Implementation procedure to reproduce time-mean LES solution by steady-state solution with SA damping

The above steps are implemented in Fluent via the user-defined functions (UDF). The UDF are written in the C programming language, compiled and hooked to the appropriate interface in Fluent.

Some case settings of the LES are modified for the solver in the source term iteration and the steady-state flow solution. As has been mentioned, the solver is changed to density-based steady-type and the SA damper model is chosen to provide the extra damping. For the source term iteration, the explicit formulation with the pseudo-time marching is used with the CFL number set as 0.5. To increase the solution speed, the implicit formulation is used in solving the μ_d in Step 3, as well as in solving the flow and SA transport equations in Step 5. The second-order upwind spatial discretization scheme is chosen for the flow equations and the μ_d transport equation. For the SA transport equation, the inflow turbulent viscosity ratio is set as say 10 at the main flow inlets and chamber inlet. The remaining case settings such as the geometry and the mesh remain the same as the LES.

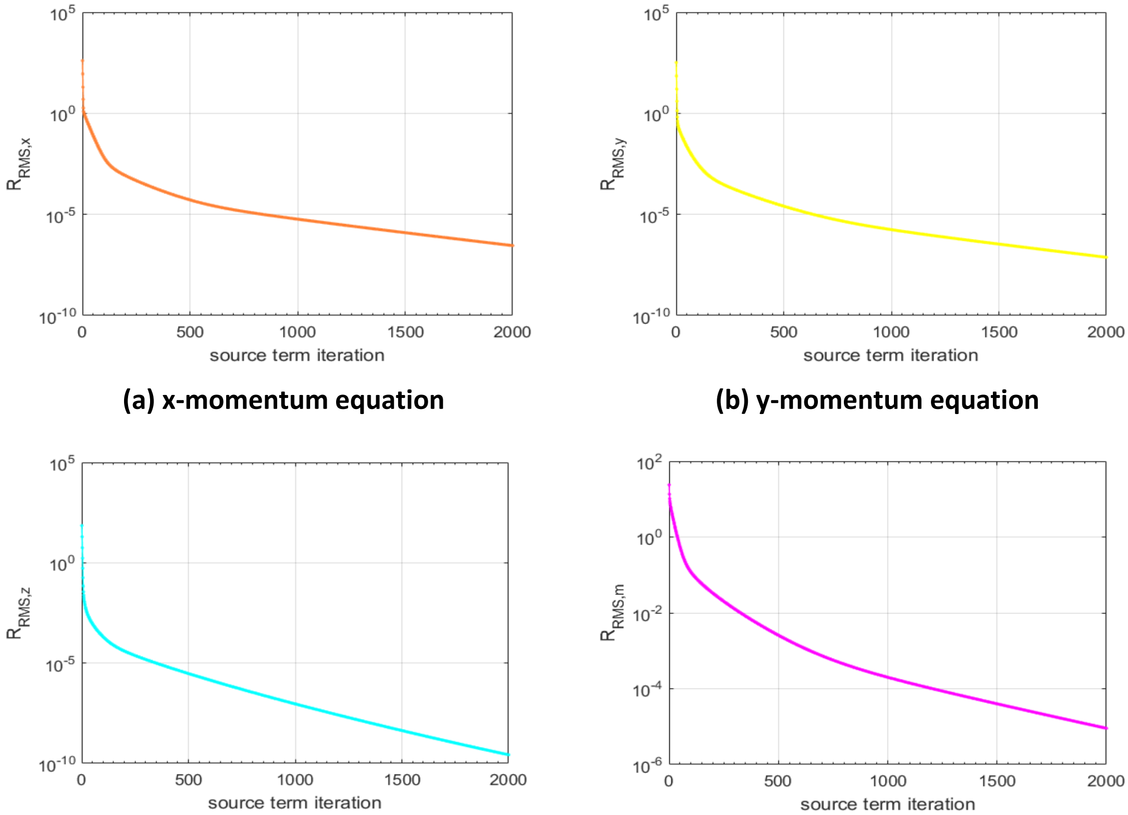
6.5 Validation Results

This section presents the validation results for the current Part I development of the efficient LES method. The aim is to reproduce the time-mean LES solution with the steady-state simulation with the source terms. The cases are chosen for an incompressible pure aerodynamic problem, a compressible aero-thermal problem and an incompressible aero-thermal problem.

6.5.1 Incompressible Cylinder Flow

The incompressible flow around the cylinder represents the incompressible pure aerodynamic problem. The LES setting for the case is the same as the cylinder case in Chapter 5. So the validated LES result there will simply be used here. The freestream Reynolds number is 3900 based on the cylinder diameter. The diameter of the cylinder is $D = 0.012\,m$, the molecular viscosity is $1.7894 \times 10^{-5}\,kg/(m \cdot s)$ constant and the density is $1.225\,kg/m^3$ constant. To match the Reynolds number, the freestream velocity is set to $4.75\,m/s$. The temperature is neglected as the energy equation is passive and the temperature variation is quite small at this low speed of the flow.

The convergence of the source term correction iteration is shown in Figure 6-3, indicated by the asymptotic decrease of the RMS change of the flow solution defined by Equation (6.18).



(a) x-momentum equation

(b) y-momentum equation

(c) z-momentum equation

(d) continuity equation

Figure 6-3 Convergence of the source term correction iteration indicated by the asymptotic decrease of the RMS change of the flow solution defined by Equation (6.18)

The damping viscosity is shown to be effective in providing a good convergence to the steady-state solution, as indicated by the continuous decrease of the solution residual in Figure 6-4. The distribution of the damping viscosity μ_d based on the time-mean LES solution is also shown in Figure 6-5. The μ_d is higher in the physically turbulent regions such as the separation region and the wake, but remains zero in the physically laminar freestream region. This indicates that the distribution of μ_d, though being only a numerical treatment, is reasonable.

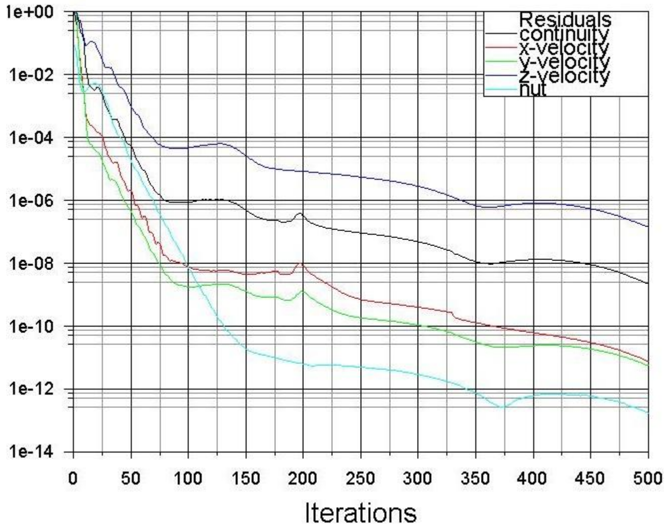


Figure 6-4 Steady-state solution residual after introducing the damping viscosity μ_d

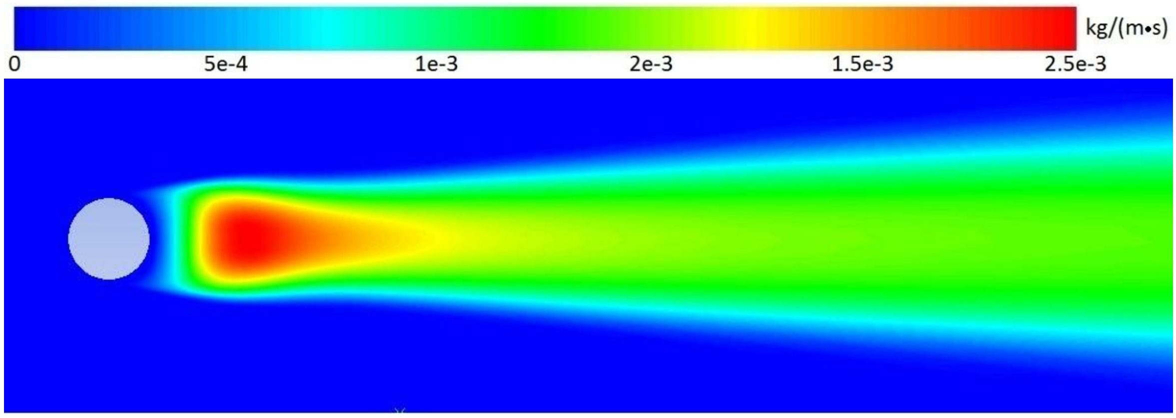


Figure 6-5 Distribution of the damping viscosity μ_d at the domain midspan, calculated based on the time-mean LES solution $\boldsymbol{Q}^*$

Assisted by the source term, the converged steady-state solution is compared with the time-mean LES solution and the RANS solution with the SA turbulence model. In Figure 6-6, it shows that the pressure on the midspan cylinder from the steady-state simulation matches very well with the time-mean LES solution, whereas the RANS prediction deviates obviously from the LES. The velocity predictions are also compared in Figure 6-7. Again the source-term-assisted steady-state solution matches well with the LES, whereas that of RANS is significantly different from the LES with a longer recirculation region and deeper wake, indicating the under-prediction of the turbulent mixing by the SA turbulence model. In practice, an absolute convergence of the time-mean LES solution cannot be obtained due to unlimited computational cost. Thus for this quasi-3D domain, the time-mean LES solution still possesses some of spanwise non-uniformity, as shown in Figure 6-8(a) by the time-mean z-velocity on the cylindrical plane concentric with the cylinder. Even so, this small spanwise non-uniformity is still well reproduced by the steady-state simulation, as shown in Figure 6-8(b).

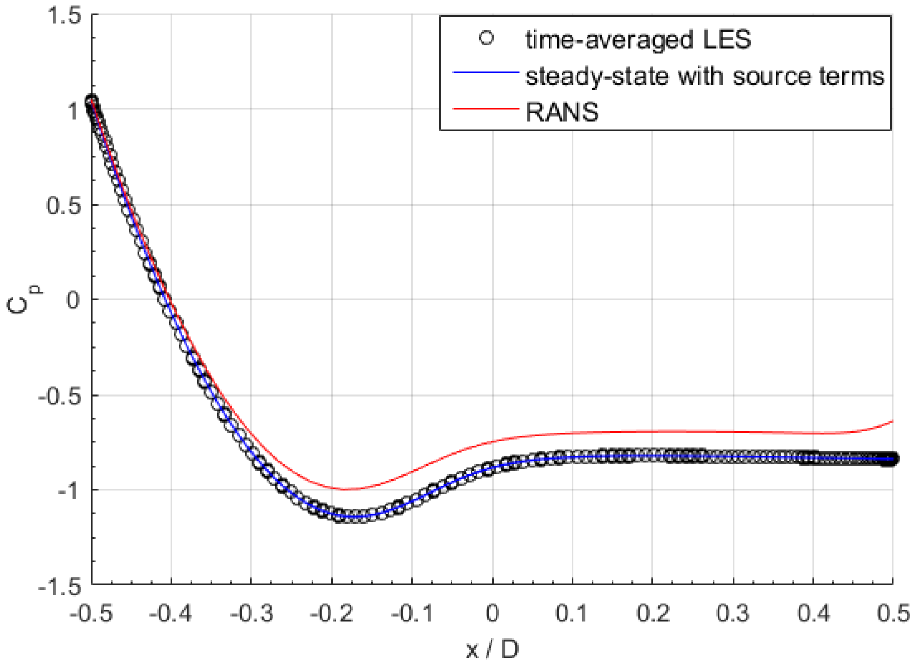
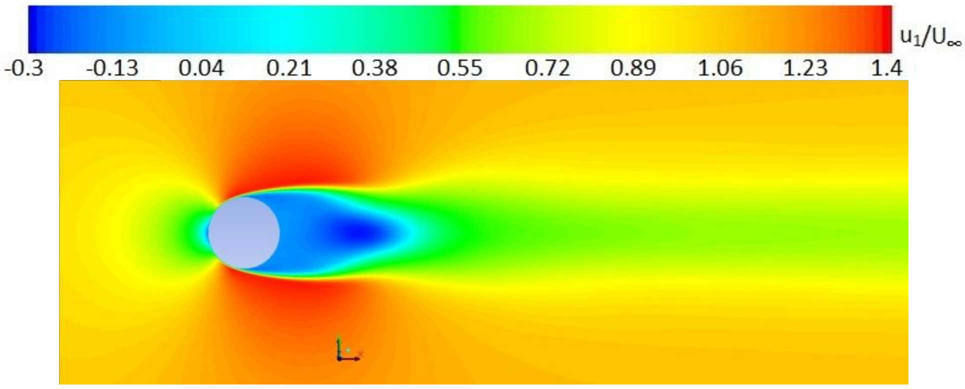
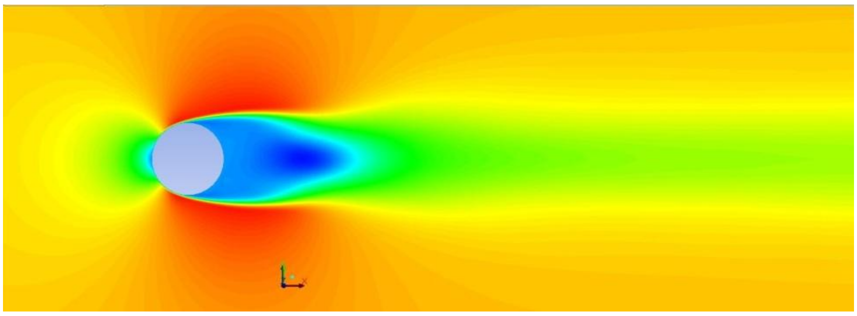


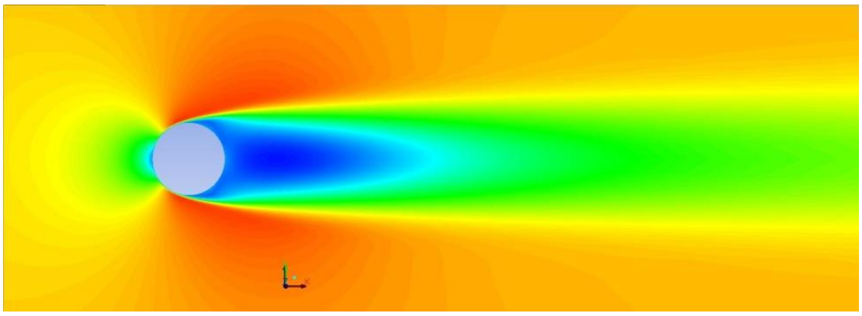
Figure 6-6 Comparison of the pressure coefficient on the midspan of the cylinder



(a) time-averaged LES solution $\bar{u}_1$

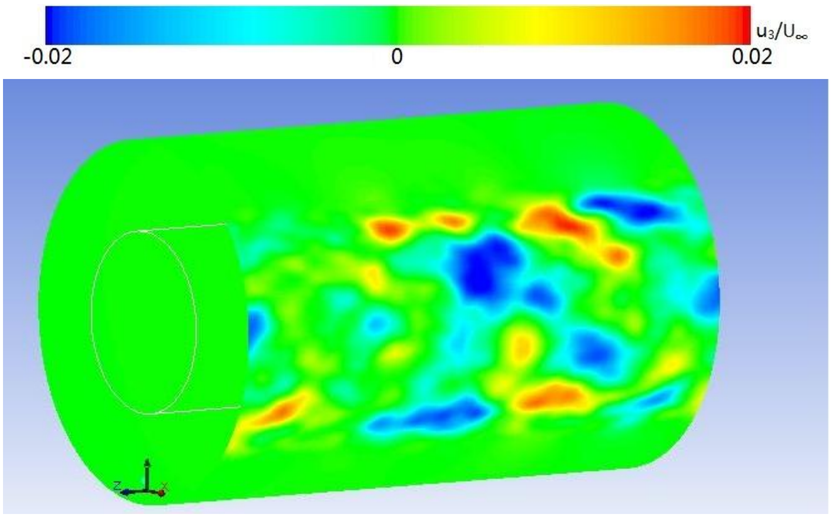


(b) steady-state solution with source terms

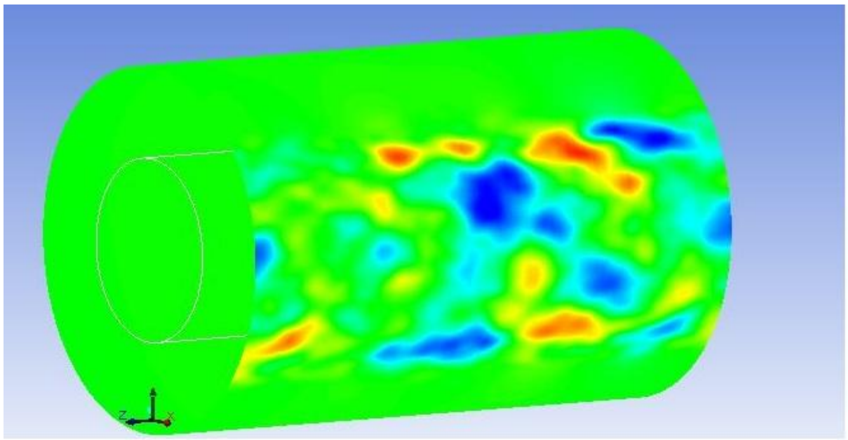


(c) RANS with SA turbulence model

Figure 6-7 Normalized x-velocity u_1 on the midspan of the domain



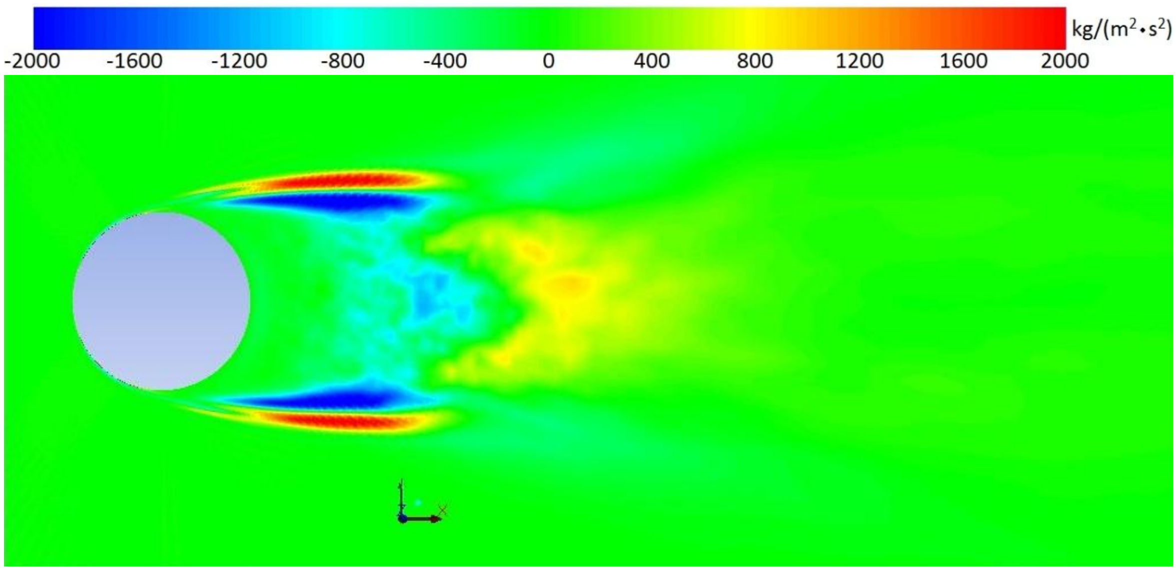
(a) time-averaged LES



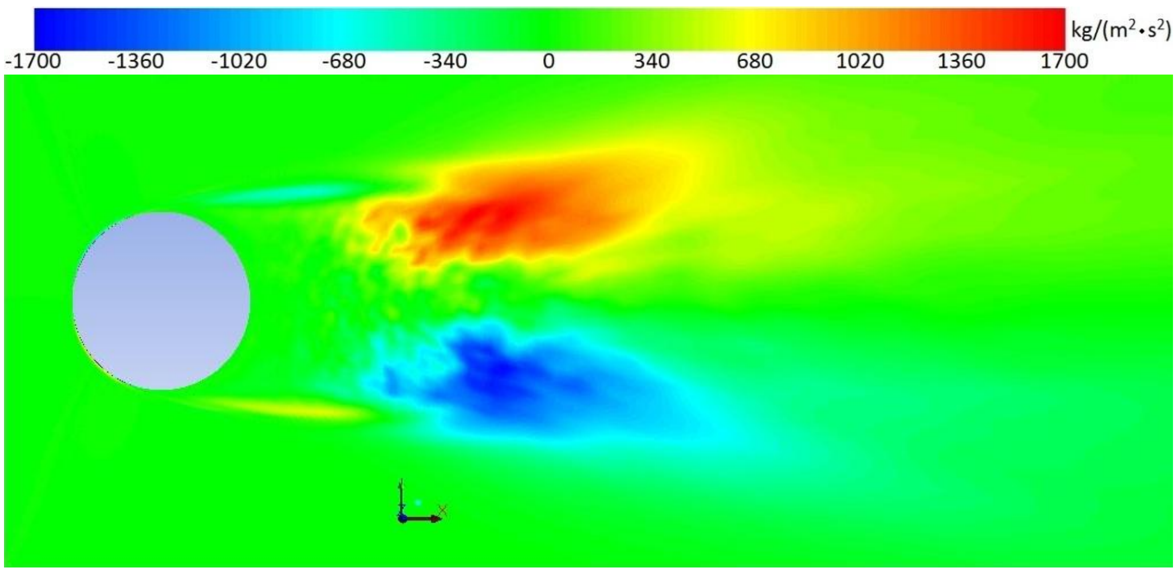
(b) steady-state simulation with source terms

Figure 6-8 Comparison of the normalized z-velocity u_3 on the cylindrical plane concentric with the cylinder

The source terms are shown in Figure 6-9. The large values mainly lie in the shear layers and the recirculation region, where most of the turbulent unsteadiness exist physically. This confirms that the source terms represent some time-mean effect of the unsteadiness, as expected.



(a) Source term for x-momentum equation: S_x



(b) Source term for y-momentum equation: S_y

Figure 6-9 Calculated source terms for the steady-state simulation of the incompressible cylinder flow

In each flow equation, the sum of the source term and the μ_d term equals the laminar residual $R(\boldsymbol{Q}^*)$. The laminar residual shall also represent the total time-mean effect of the unsteadiness, as theoretically it is equal to the sum of the spatial derivative terms of the Reynolds stresses in that flow equation. To show this, the laminar residual represented by the source term plus the μ_d term, $S_i + \frac{\partial}{\partial x_j}\left(2\mu_d \cdot S_{v,ij}\right)$, is compared to the representation by the sum of the spatial derivative terms of the Reynolds stresses $\frac{\partial\left(-\rho\overline{u_i' u_k'}\right)}{\partial x_k}$ in a particular momentum equation, for which the Reynolds stresses are computed by time-averaging the instantaneous LES solution. The comparison is shown in Figure 6-10. It can be seen that, the results of the two representations are quite similar. Slight differences can be observed, which may be attributed to the effect of the SGS terms and the numerical discretization errors.

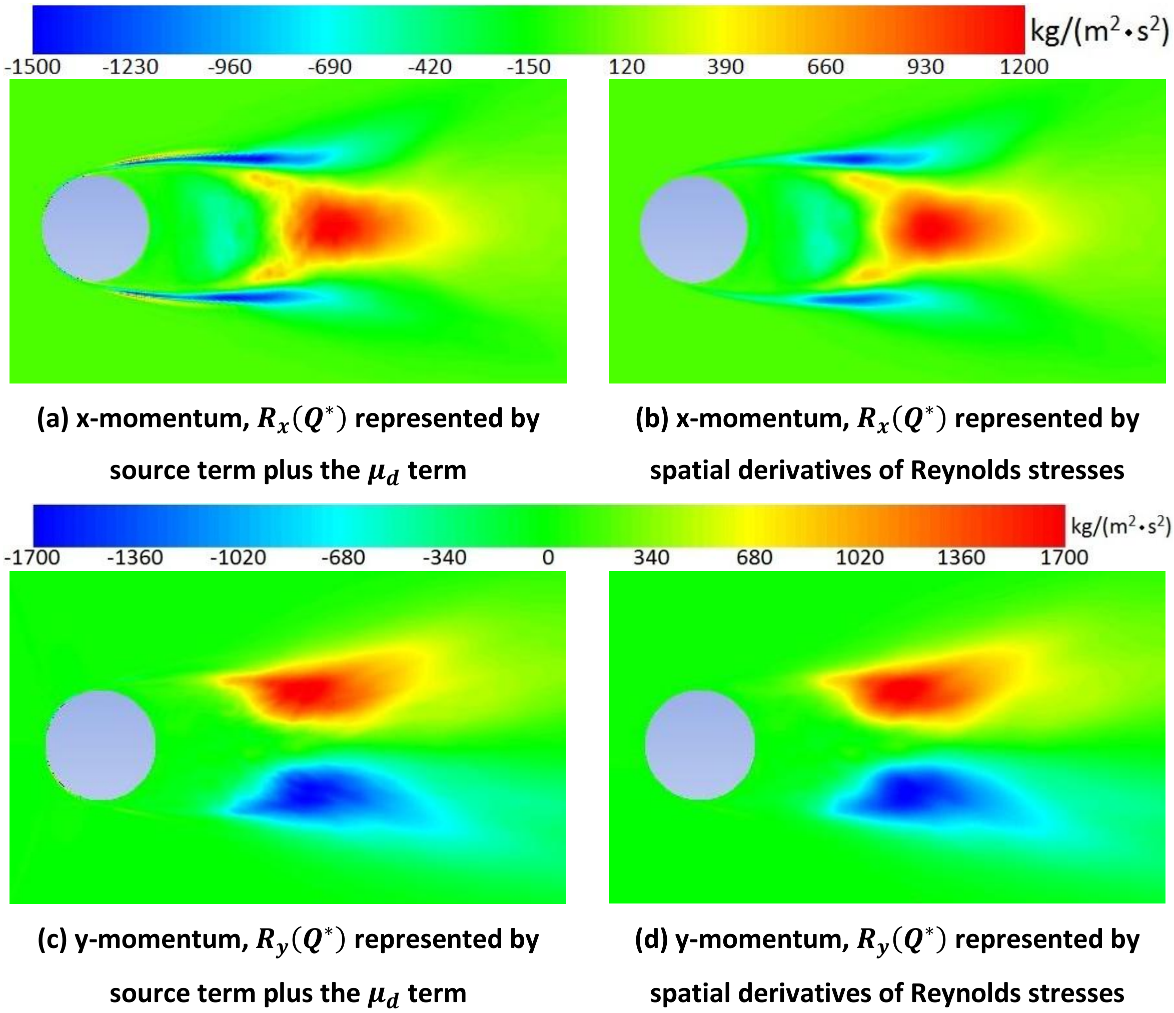
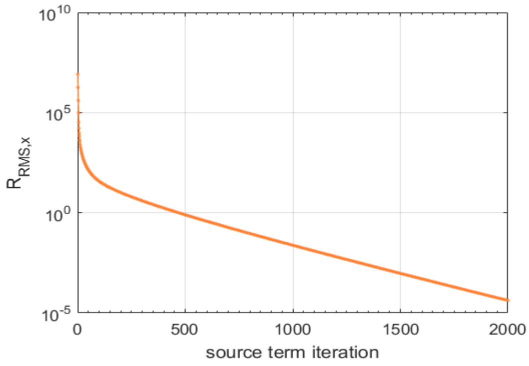


Figure 6-10 The laminar residual $\boldsymbol{R}(\boldsymbol{Q}^*)$ reconstructed from the source term plus the $\boldsymbol{\mu_d}$ term, and from the sum of the Reynolds stress derivatives.

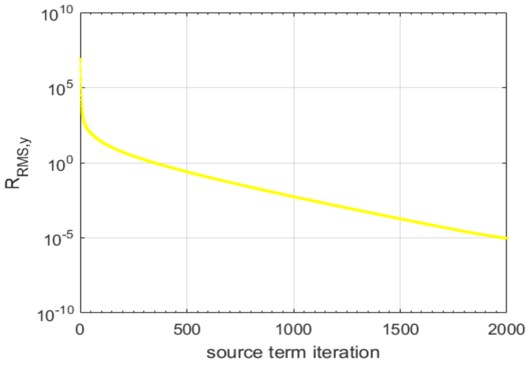
6.5.2 Compressible Cylinder Flow

The compressible flow around the cylinder is an aero-thermal problem, as the energy equation is now active has to be solved to close the flow equations. In this case, the freestream Mach number is set to 0.4. The diameter of the cylinder is adjusted to maintain the freestream Reynolds number of 3900. The LES is performed with the time step of $1.62 \times 10^{-8}\ s$. The transient stage takes 6×10^{4} time steps to establish the turbulent flow while the mean flow is averaged in the subsequent 6×10^{4} time steps of the statistic collection stage.

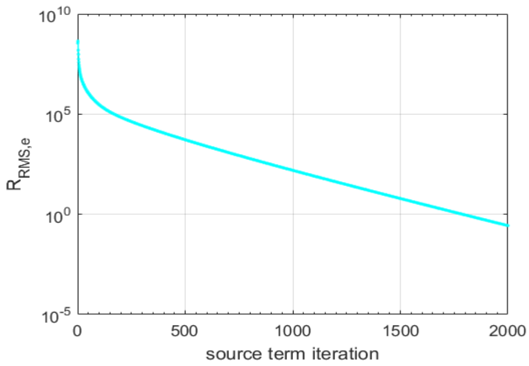
The convergence of the source terms are shown in Figure 6-11. Convergence of the steady-state solution is shown in Figure 6-12. Various aspects of the steady-state solution are compared with the time-mean LES solution and the RANS solution with the SA turbulence model. The static pressure and temperature on the cylinder surface are compared in Figure 6-13 and Figure 6-14. Both the pressure and the temperature from the source-term assisted steady-state solution match very well with the time-mean LES solution, while the RANS solution is quite different, particularly for the temperature. Contours of the velocity and the temperature in the flow field are also compared in Figure 6-15 and Figure 6-16. Again, it shows that the steady-state solution with the source terms matches well with the LES, while the RANS solution is quite different. The source terms are shown in Figure 6-17.



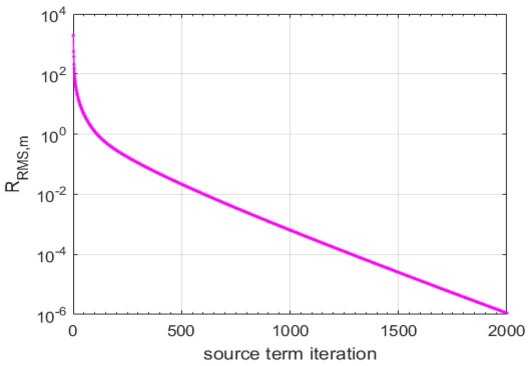
(a) x-momentum equation



(b) y-momentum equation



(c) energy equation



(d) continuity equation

Figure 6-11 Convergence of the source term correction iteration shown by the decrease of RMS solution change

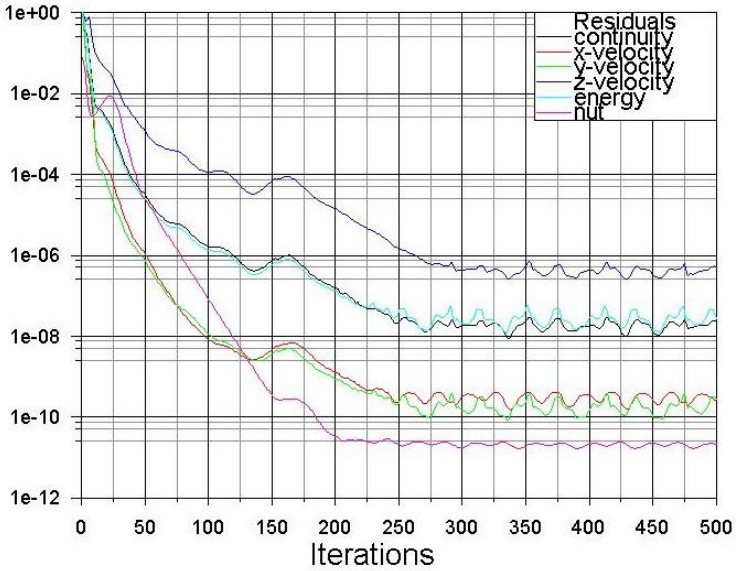


Figure 6-12 Convergence of the steady-state solution

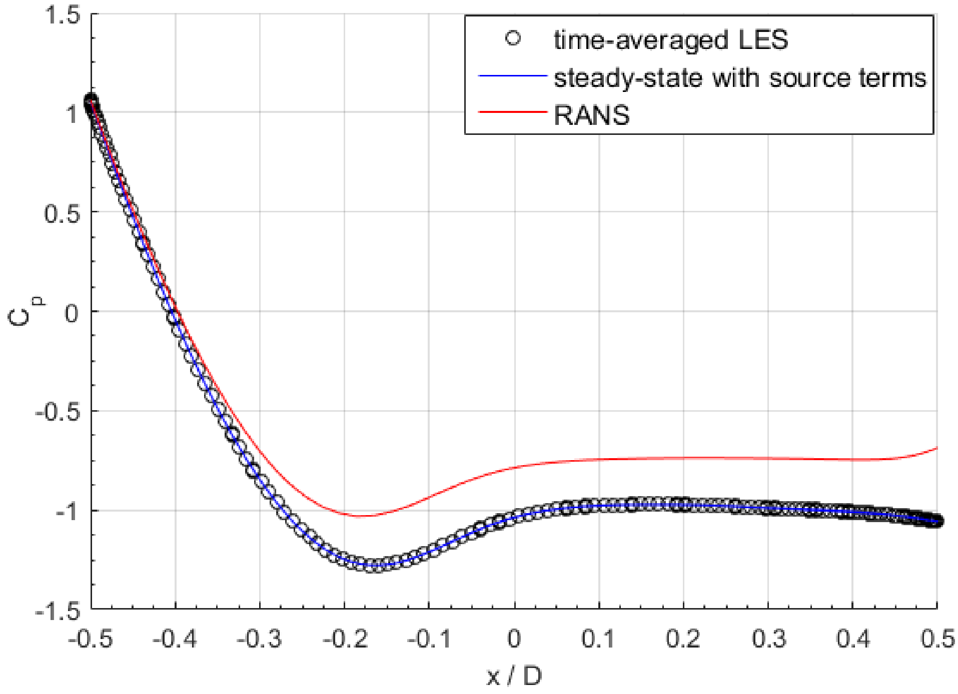


Figure 6-13 Comparison of the pressure coefficient on the midspan cylinder surface

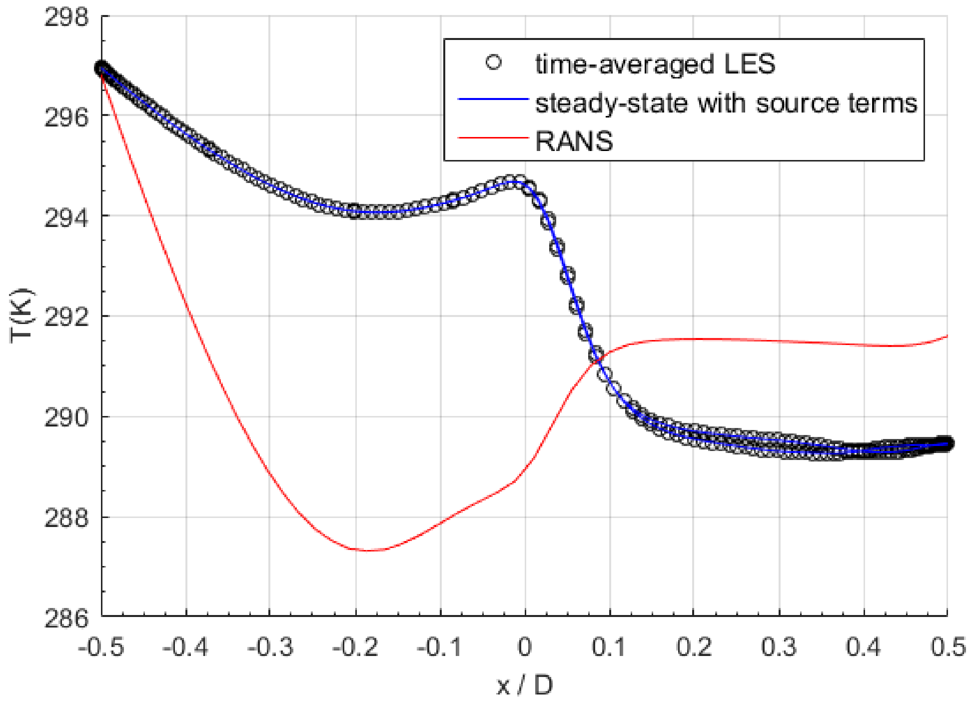
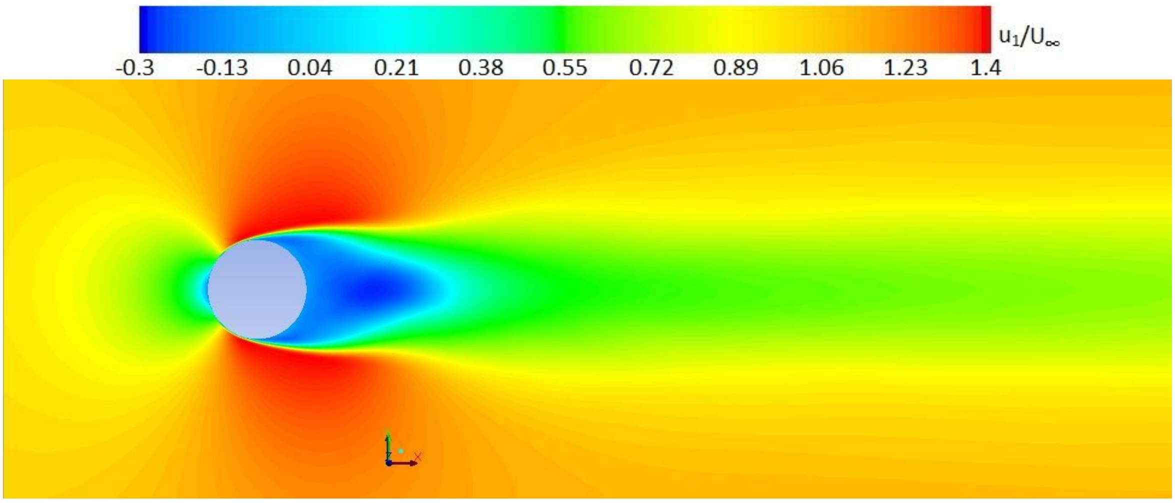
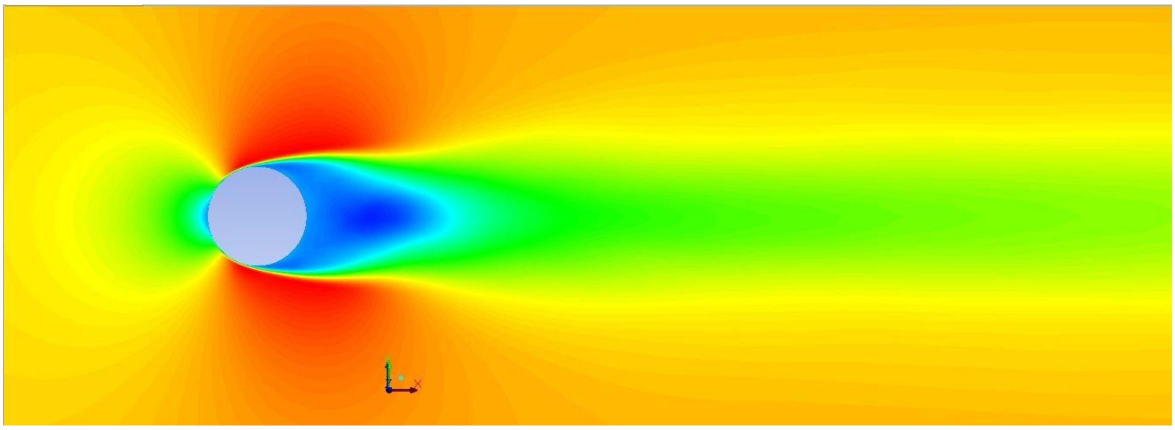


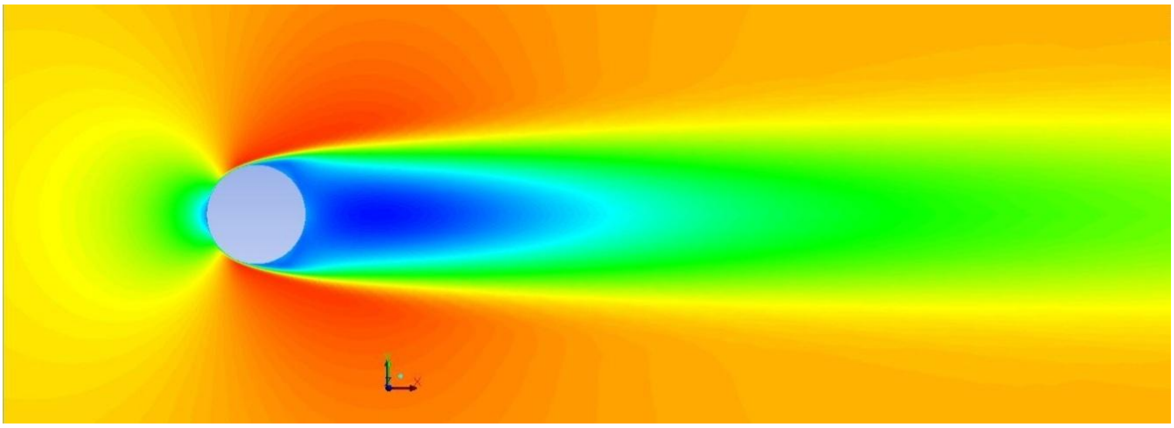
Figure 6-14 Comparison of the temperature on the midspan cylinder surface



(a) Favre-averaged LES solution $\widetilde{u}_1$

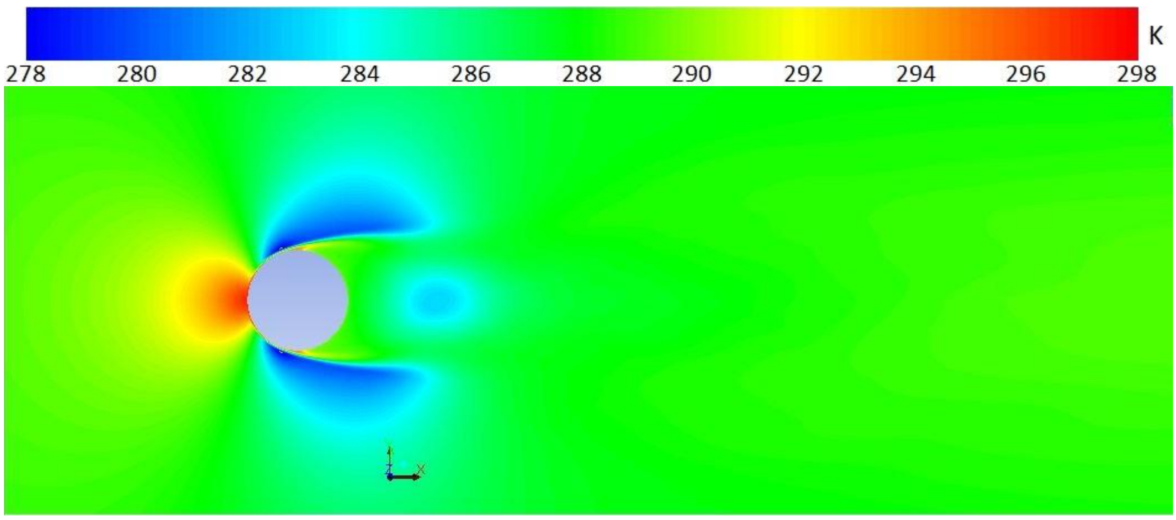


(b) steady-state solution with source terms

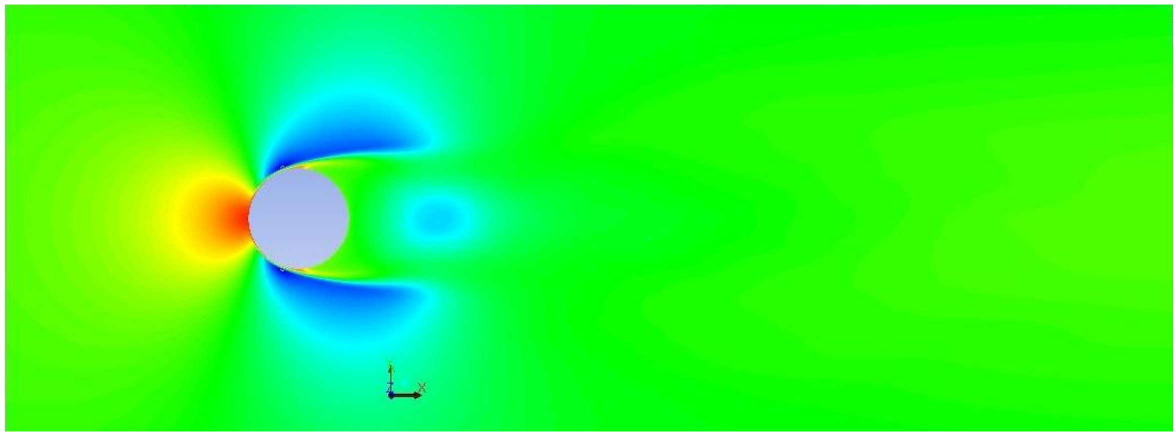


(c) RANS with SA turbulence model

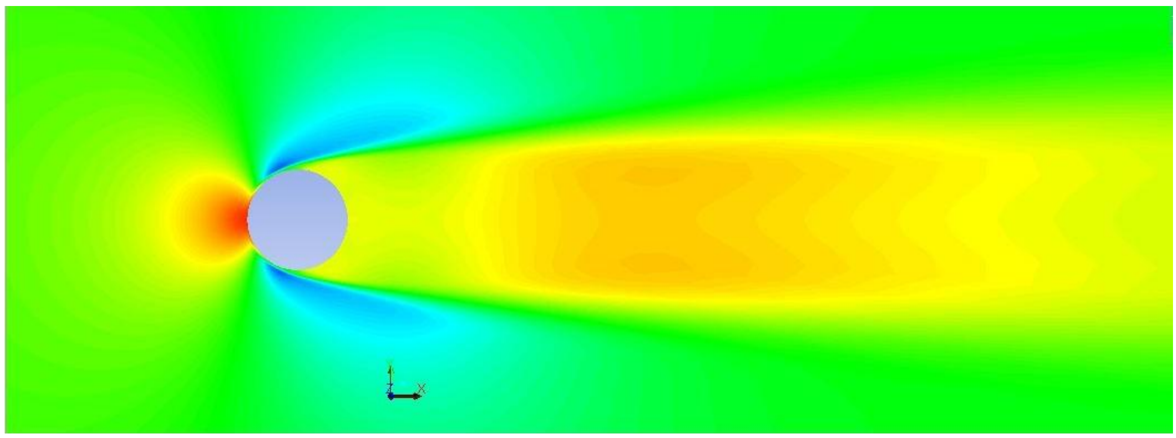
Figure 6-15 Normalized x-velocity u_1 on the midspan cutplane



(a) time-averaged LES solution $\overline{T}$

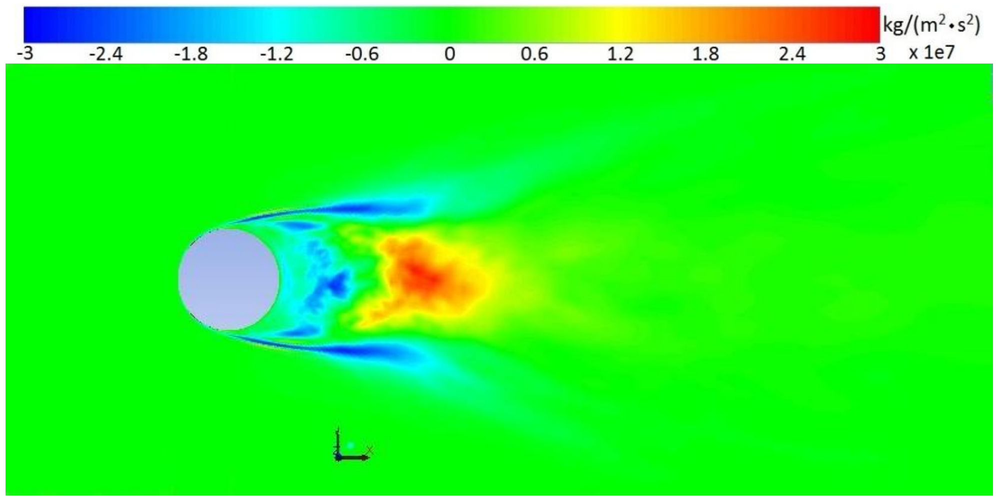


(b) steady-state solution with source terms

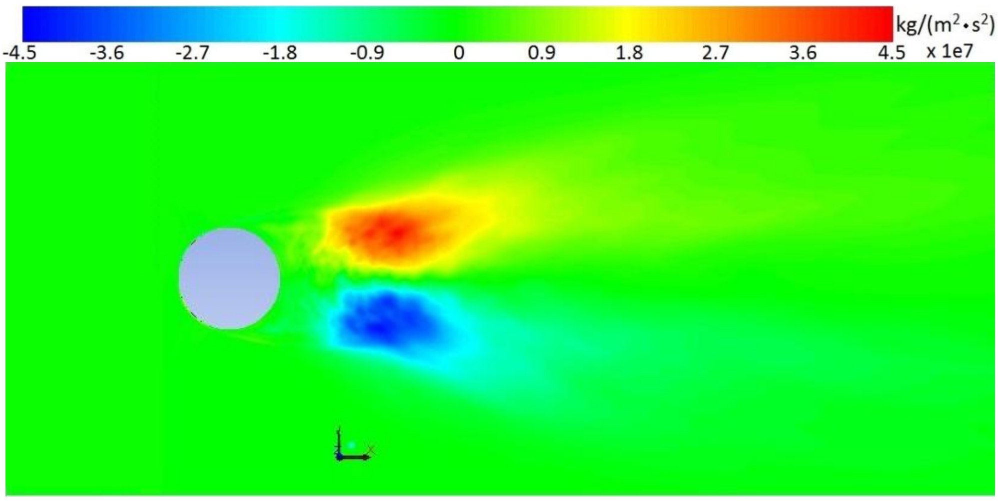


(c) RANS with SA turbulence model

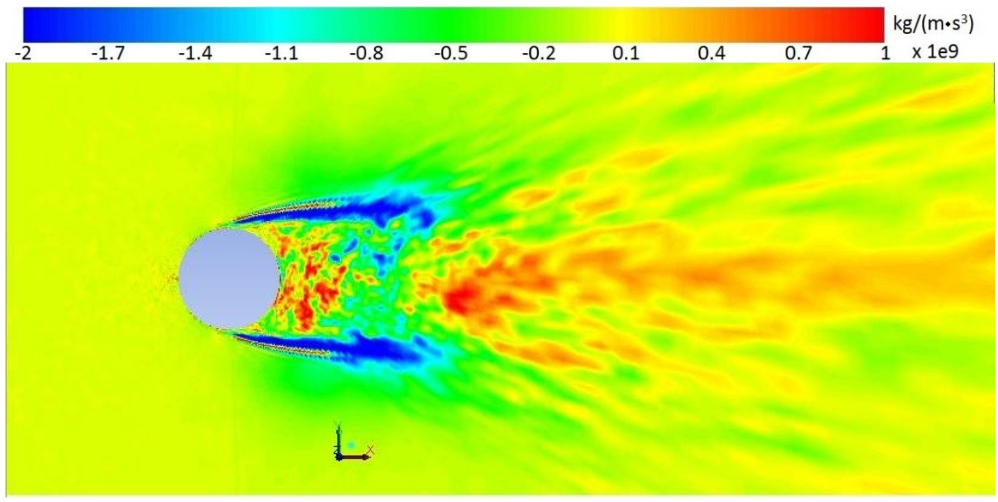
Figure 6-16 Comparison of the predicted temperature



(a) x-momentm equation source term S_x



(b) y-momentum equation source term S_y



(c) energy equation source term S_e

Figure 6-17 Source terms for the steady-state simulation of the compressible cylinder flow

6.5.3 Trailing Edge Cutback Injection

The validation is also carried out for a realistic case of the trailing edge cutback injection. This case represents the incompressible aero-thermal problem. The temperature field is passive, but is a main interest of this type of problem.

The case setting and the results of the LES will simply use those of the cutback trailing edge injection case described in Chapter 5. The experiment by Ling et al. (2015) is also used as reference, as already described in Chapter 5. The flow set incompressible, the temperature difference between the mainstream and the blowing jet is $20\ K$.

The RMS change of the flow solution during the source term iteration is shown in Figure 6-18 indicating the convergence of the source terms. The convergence of the steady-state solution after the source term is obtained is shown in Figure 6-19.

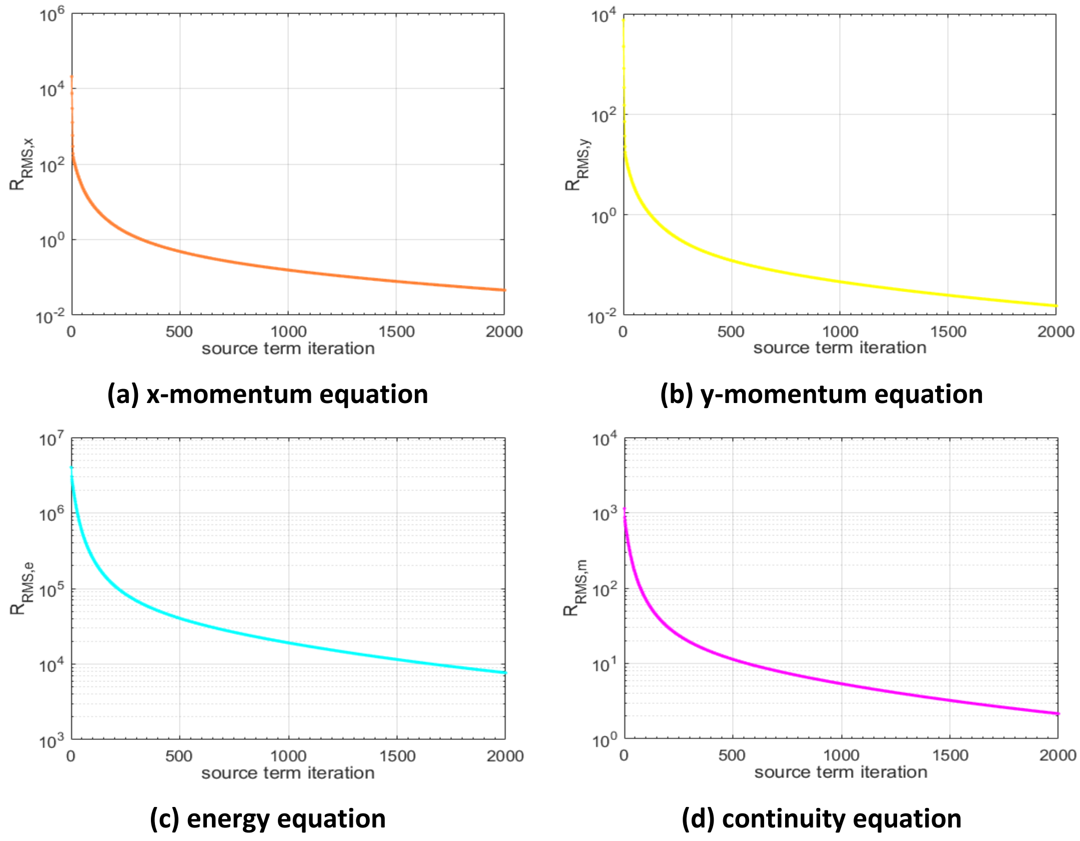


Figure 6-18 Convergence of the source term correction iteration shown by the decrease of RMS solution change

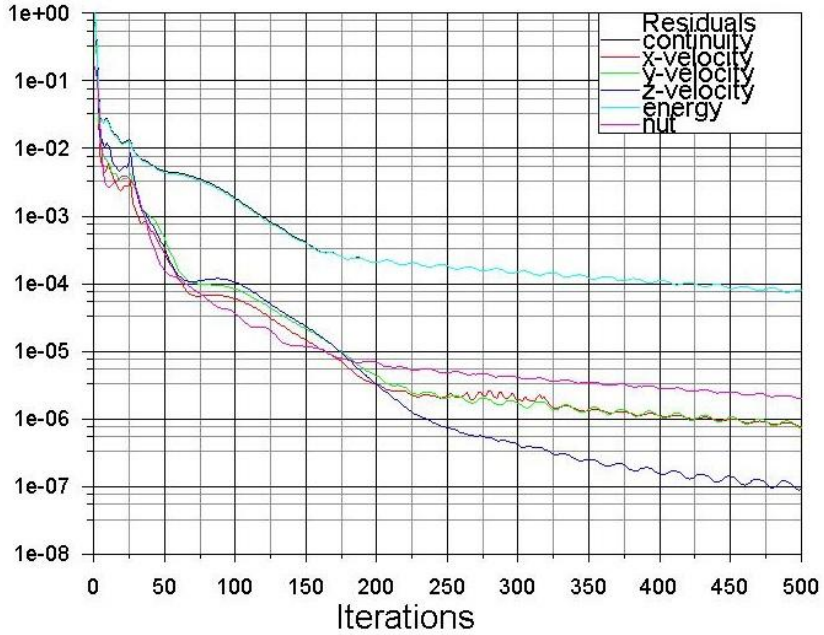


Figure 6-19 Convergence of the flow solution of the steady-state simulation with source terms

Various parameters of the steady-state solution are compared to the LES solution and the RANS solutions with different turbulence models. Comparison of the velocity and the temperature in the flow field are shown in Figure 6-20 and Figure 6-21. The steady-state solution matches the time-mean LES very well. The two RANS solutions using different turbulence models are both quite different from the LES solution. The RANS solution also shows strong dependency on the turbulence models. Small fluctuations can be observed in the RANS solution with the k-ω SST turbulent model.

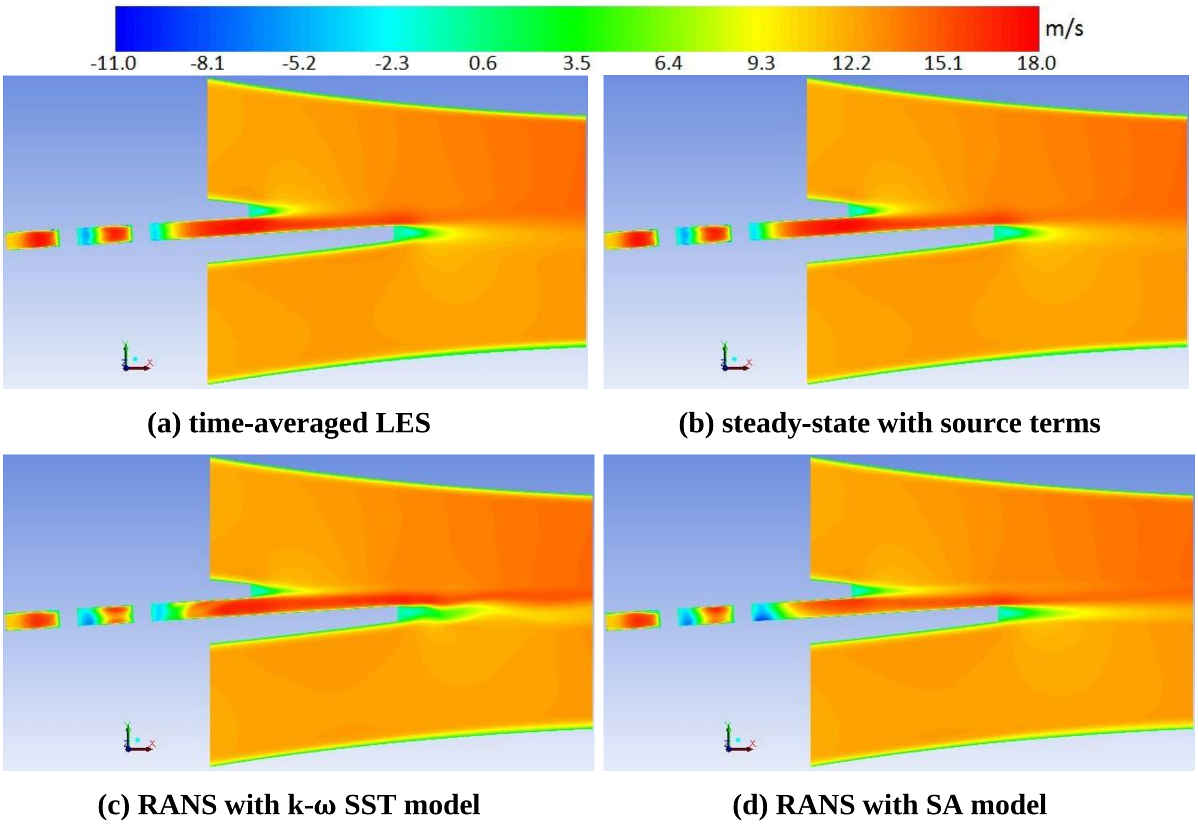
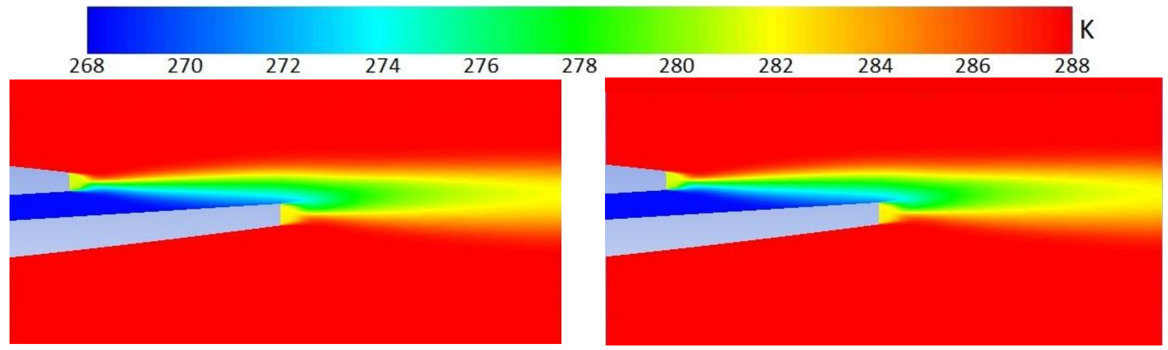


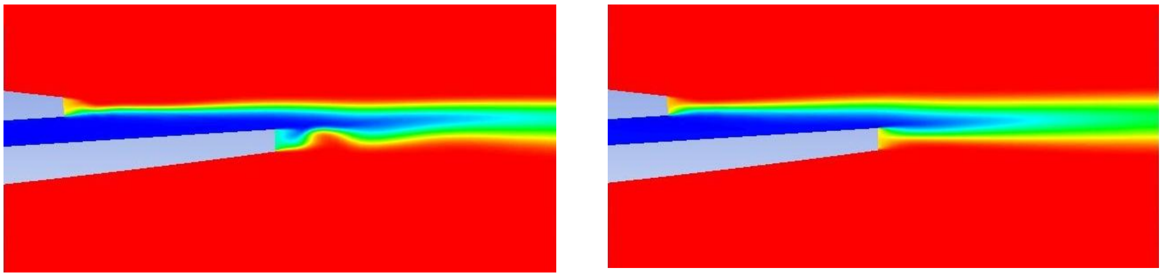
Figure 6-20 Comparison of the predicted x-velocity u_1 on the domain midspan

The temperatures are compared in Figure 6-21 and Figure 6-22. Figure 6-21 shows that the level of turbulence mixing is correctly predicted by the steady-state solution, showing almost identical wake with the LES, while both RANS turbulence models under-predicts the mixing, producing narrower and longer wakes. In Figure 6-22, the contours of the predicted temperature by the steady-state simulation are shown to be almost identical to those of the LES, capturing the increase of temperature towards the trailing edge, whereas the temperatures in both RANS stay almost constant and remain as low as the coolant on the entire breakout surface. Also, the spanwise-averaged adiabatic effectiveness on the trailing edge surface is compared in Figure 6-23. It can be seen that the steady-state solution of the source term match perfectly exactly with the time-mean LES solution, while the RANS prediction using two different turbulence models both deviate quite significantly.



(a) time-averaged LES

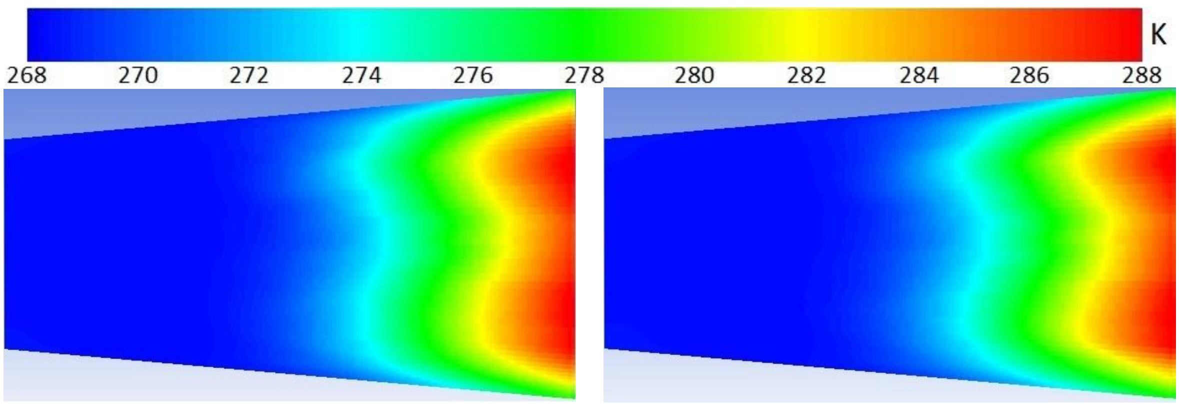
(b) steady-state with source terms



(c) RANS with k-ω SST model

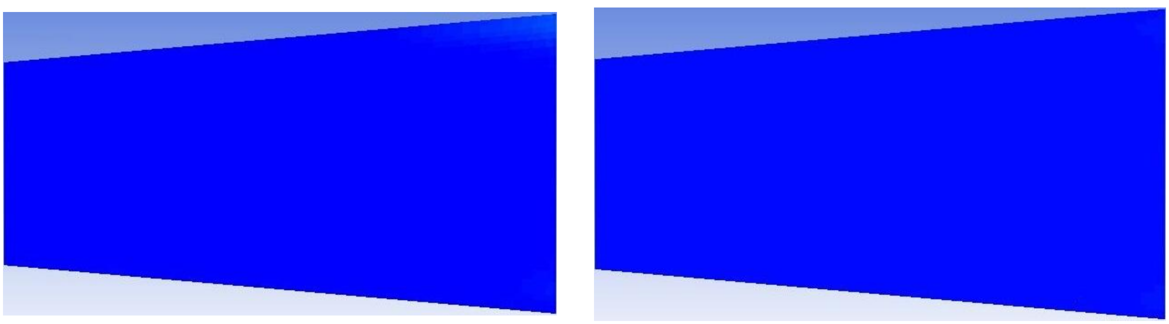
(d) RANS with SA model

Figure 6-21 Comparison of temperature prediction on the domain midspan



(a) time-averaged LES

(b) steady-state with source terms



(c) RANS with k-ω SST model

(d) RANS with SA model

Figure 6-22 Comparison of temperatures on the breakout surface (flow from left to right)

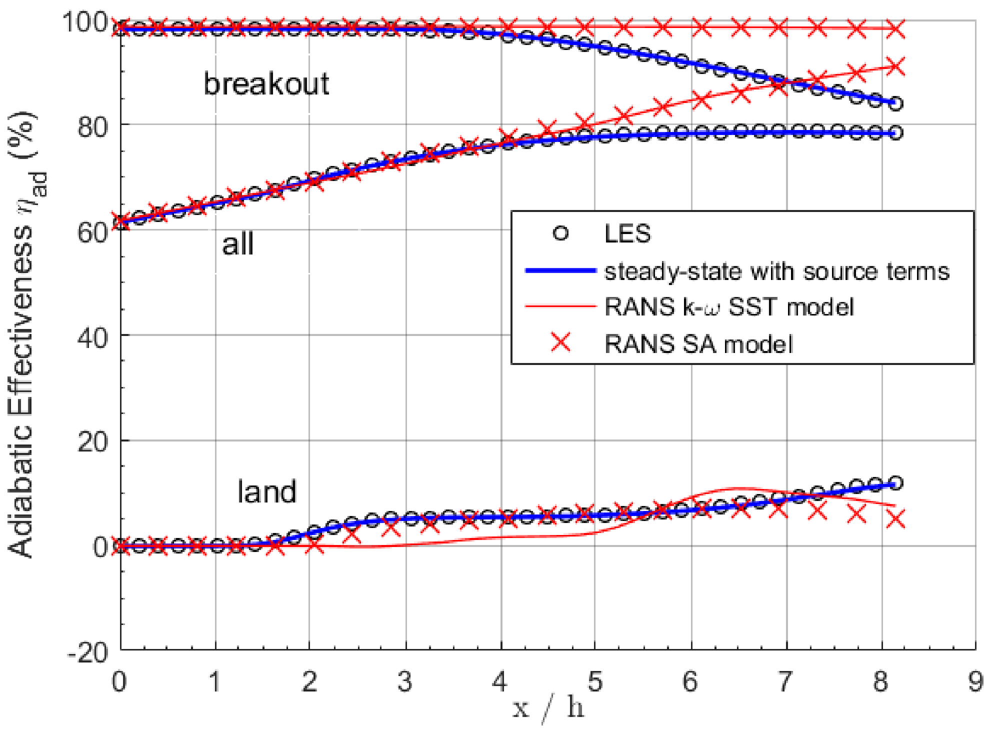
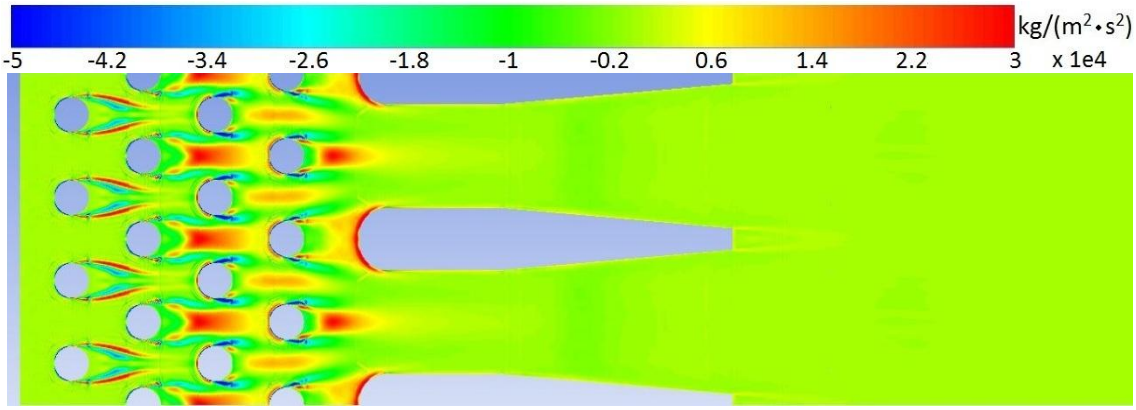


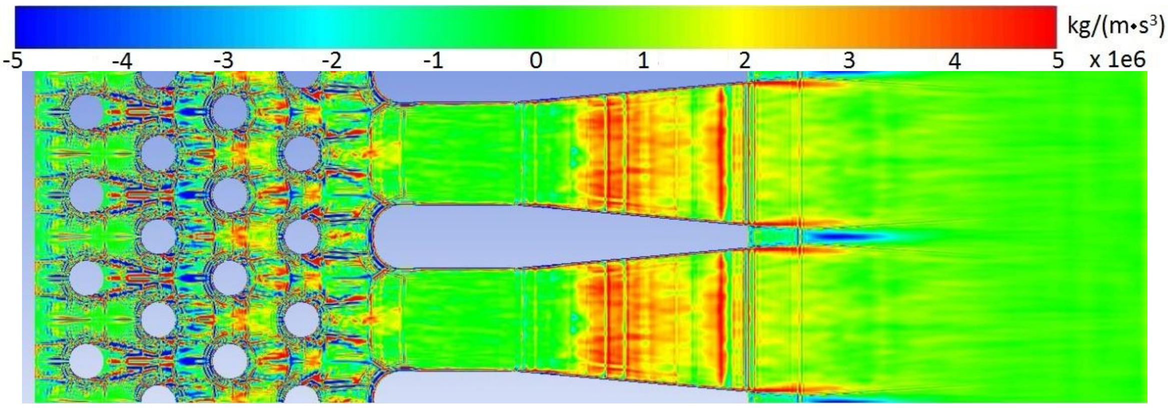
Figure 6-23 Comparison of the spanwise-averaged adiabatic effectiveness on the surface of the trailing edge

In Figure 6-24 below, the source terms for the equations are shown on the cutplane going through the mid-height of the slot and coolant chamber. The x-momentum source term in Figure 6-24a is more pronounce after the cylindrical pin-fin, and the contour is similar to that of the cylinder flow. The energy source term in Figure 6-24b is less smooth, which may be attributed to the discretization of the flow field by the mesh.

In sum, the above results show that the time-mean LES solution can be accurately reproduced by the steady-state simulation using the source terms derived from the same time-mean LES solution on the same design. This is found to be true for all the trailing edge cutback cooling designs investigated in this study.



(a) x-momentum equation source term S_x



(b) energy equation source term S_e

Figure 6-24 Calculated source terms for the trailing cutback injection configuration, on the cutplane through the mid-height of the slot and chamber.

6.6 Summary

The Part I of the development of the efficient LES method is presented in this chapter. The goal is to extract the source terms from the time-mean LES solution to enable the fast steady-state solution to reproduce the time-mean LES solution on the LES-evaluated design itself. Due to the lack of the explicit Euler pseudo-time discretization, the source term have to be computed iteratively. In addition, the SA turbulence model is used to provide additional damping to get the steady-state solution to converge effectively. It should be emphasized that the damping viscosity μ_d from the SA model is only a used as a numerical treatment for stabilization, despite that μ_d would still take up part of the effect of the unsteadiness on the time-mean flow. The μ_d terms and the source terms together represent the full effect of the unsteadiness. The entire procedure of Part I is implemented in Fluent via the interface of the user-defined functions (UDF).

The validation results show that the steady-state simulation with the source terms has successfully produced the converged steady-state solution and matches the time-mean LES solution on the LES-evaluated design itself, for the incompressible flow pure aerodynamic problem, compressible flow aero-thermal problem, and the incompressible flow aero-thermal problem. On the other hand, the RANS solutions deviate significantly from the LES, and show strong dependency on the turbulence models.

However, to make the efficient LES approach fully useful for the design optimizations, the source terms need to be propagated to the other designs, so that the fast steady-state simulation can be applied for the rest of the performance evaluation without any more LES. The source term propagation constitutes the Part II development and will be discussed in the next chapter.

Chapter 7

An Efficient LES Method for Design Optimization

- Part II: Source Term Propagation

Based on the Part I development as described in the previous chapter, the source terms extracted from the time-mean LES solution can only enable the steady-state simulation to be used in reproducing the time-mean LES solution on the LES-evaluated design itself. For the LES method to be efficiently useful for the design optimization, in particular for the combined flow control and shaping optimization, the source terms need to be propagated to the new designs so that those designs can be evaluated with the steady-state simulations without running the expensive LES on them, and also the accuracy of the steady-state solutions can still be comparable to that of the LES.

In this chapter, various source term propagation methods will be developed and validated. As the prerequisites for the propagation, the issues of the geometric similarity and the practical implementation consideration for the meshing need to be discussed first.

7.1 Flow, Geometric Similarities and Mesh Consistency

The underlying assumption that the source terms can be propagated is that the designs in the same design space are similar and vary smoothly. Consequently it can be expected that the corresponding turbulent flows will share some similar features. Specifically, it requires the shapes to be topologically similar. Changes in the influence of the turbulence should also be smooth when the design variables change. For example, the cutback injection designs with different taper angles can be seen as topologically similar, as there is no abrupt occurrence or disappearance of shape features. Currently, the maximum taper angle is limited to prevent the side of the lands meets before the trailing edge. However, it is possible that future propagation method developed may be able to account for more abrupt changes, so that the land meeting at and before the trailing edge can both be regarded as topologically similar in the perspective of the source term propagation method. Also, the flow control designs need to be similar as well. For example, the designs with different blowing ratios can be seen as similar in terms of the flow control.

In terms of the numerical implementation, a practical consideration is also given in the present work such that certain meshes are preferred for easy implementation without resorting to further interpolations. As the data are stored in the centroids of the mesh cells, the propagation of the source terms prefers the meshes to be simply consistent, so that the data can be transferred among different meshes without any grid data interpolation, so as to reduce the complexity and errors in the source term propagation. This poses two specific requirements. First, the meshes need to be topologically similar. This can only be realized when the geometries are topologically similar as well. Specifically, it requires the mesh to stretch or shrink gradually as the geometry deforms, but keeping the same number of cells, the same laws of relative node distributions and the same type of mesh, as illustrated in Figure 7-1 below, in which the scenarios that Mesh B is topologically similar or dissimilar to Mesh A are illustrated.

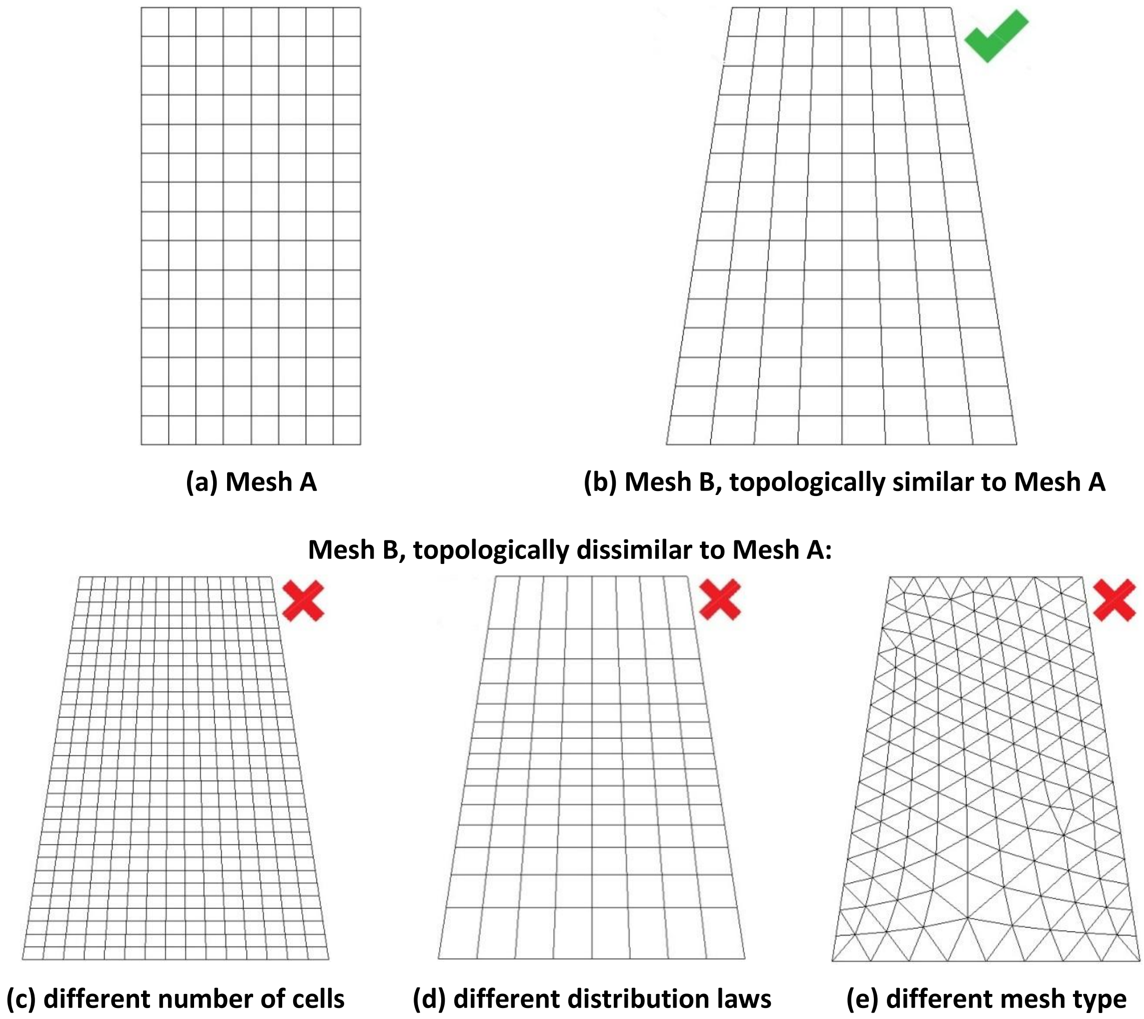


Figure 7-1 Topological similarity between Mesh A and Mesh B

Second, the mesh cells need to be indexed in a consistent sequence among different meshes. As Fluent treats all kinds of meshes as unstructured, the mesh cells are not indexed based on the topological directions. Instead, each cell is assigned a unique cell index, numbered from 1 to the total number of the cells N_{cell}. By indexing the cells in a consistent sequence, as illustrated in Figure 7-2 below, all the mesh cells at the same topological location of the different meshes will have the same index respectively, and a linkage among them can be established for a direct data transfer. The prerequisite for the consistent cell index sequence is the topological similarity among the meshes, especially the requirements for the same number of cells and the same type of mesh.

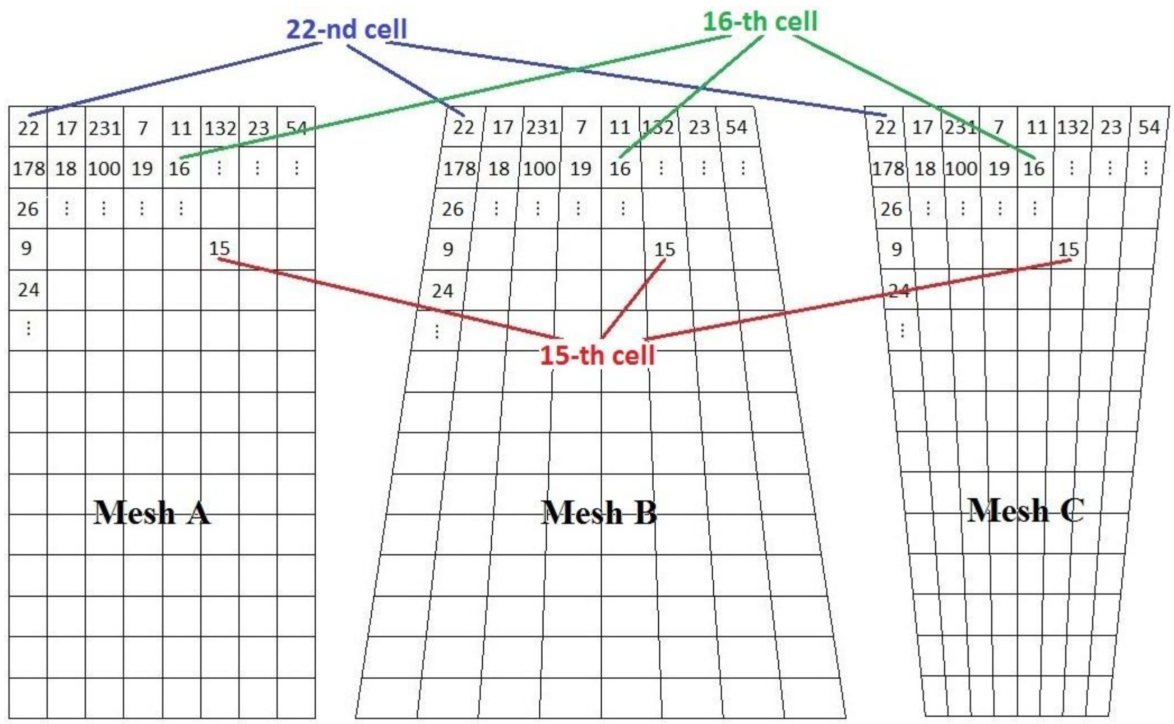


Figure 7-2 Consistent sequence of the cell index among different meshes

Generating consistent meshes for different geometries in ANSYS ICEM CFD can be accomplished by using a unified procedure or script. However, for 3D complicated geometries, it is difficult to make the scripted automatic mesh generation robust, particularly when generating the non-hexagon cells. So, it is more convenient to generate the consistent meshes manually using a unified procedure, with the same mesh distribution law accounting for the geometric deformation. Some test cases in this study confirm that the meshes generated in this manner are topologically similar and consistent in the cell index sequence.

In the following content, various methods to propagate the source terms will be discussed. These methods can be implemented in two major approaches: the cell-to-cell propagation approach and the proper-orthogonal-decomposition (POD) propagation approach.

It should be emphasised that the propagation only involves the source terms, which may be viewed as interpolated turbulence stress terms for similar designs within a reasonable range. Having had the propagated stress terms, the evaluation of the design will still be subject to the numerical solution of the conservation law at the condition concerned. This would distinguish the present method from most other surrogate models for design optimization.

7.2 Cell-to-Cell Propagation Approach

A straight forward approach to propagate the source term is the cell-to-cell approach. In this approach, the source terms on the new design are computed for one mesh cell and then another. For each cell, the value of the source term of a flow equation is propagated from the same indexed cells from the sample designs. As illustrated in Figure 7-3 below, for example, the source term values in the 22-nd cells of the new design Design C ($\boldsymbol{d}^C$) is propagated from the 22-nd cells of the sample designs Design A ($\boldsymbol{d}^A$) and Design B ($\boldsymbol{d}^B$). After that, the propagation may continue for the 17-th cells, the 231-st cells, and so on, until all the cells on Design C have received the source term values. The relationship $L(\cdot)$ between the new and sample source term for each cell depends on the design variables at the new design as well as the chosen propagation method.

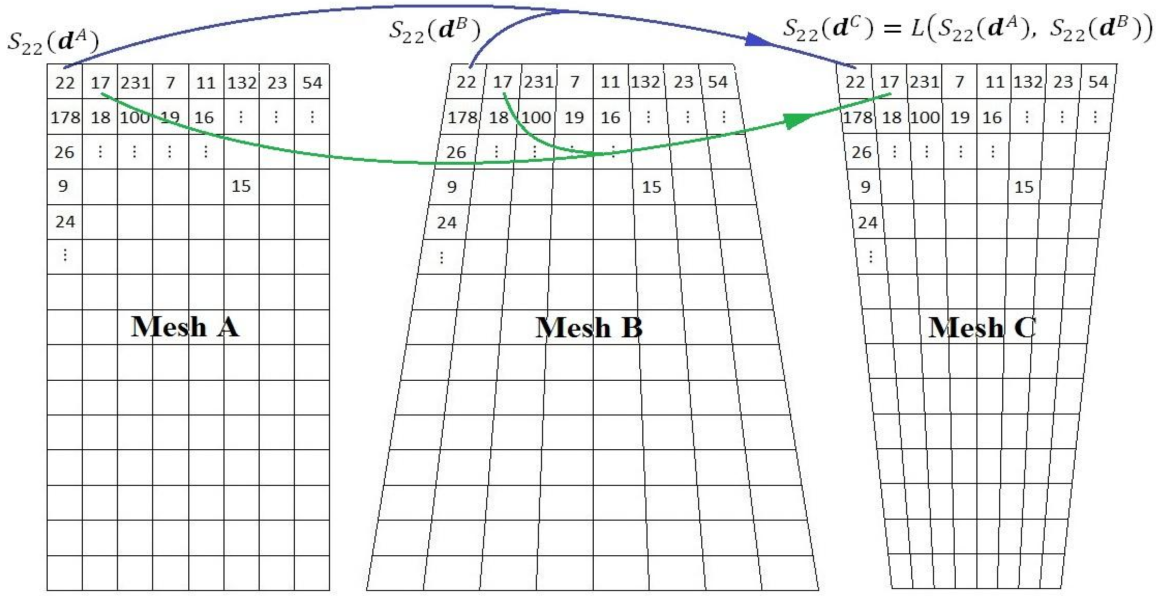


Figure 7-3 Cell-to-cell source term propagation approach, example shown for Cell 22

In the following content, several propagation methods implemented in the cell-to-cell propagation approach are discussed. These methods are based on some common interpolation/approximation methods.

7.2.1 Polynomial Regression

One possible method to compute the source term on the new design is via the polynomial regression, whose coefficients are to be determined from the sample source term values. The polynomial is a function of the N_V design variables with N degree in general. Thus, the source term of a flow equation in the c-th cell of the new design can be expressed as:

$$S_c(\boldsymbol{d}) = \sum_{p=1}^{P}\left[b_{p,c} \cdot \prod_{i=1}^{N_V}(d_i)^{n_{i,p}}\right], (c = 1, 2, \dots, N_{cell}) \quad \textbf{(7.1)}$$

or more concisely as:

$$S_c(\boldsymbol{d}) = \boldsymbol{f}^T(\boldsymbol{d})\boldsymbol{b}_c \quad \textbf{(7.2)}$$

In above, $b_{p,c}$ is the coefficient of the p-th term in the polynomial for the c-th cell, contained in the vector $\boldsymbol{b}_c = [b_{1,c} \;\cdots\; b_{P,c}]^T$. d_i is the i-th design variable contained in the vector $\boldsymbol{d} = [d_1 \;\cdots\; d_{N_V}]^T$. $n_{i,p}$ is the integer power of d_i in the p-th polynomial term, where $n_{i,p} \geq 0$ and $\sum_{i=1}^{N_V} n_{i,p} \leq N$ for each p. $\boldsymbol{f}(\boldsymbol{d})$ is vector containing P polynomial basis functions. For example, the quadratic polynomial for 2 design variables is:

$$S_c(d_1, d_2) = b_{1,c}(d_1)^2 + b_{2,c}(d_2)^2 + b_{3,c}d_1d_2 + b_{4,c}d_1 + b_{5,c}d_2 + b_{6,c} \tag{7.3}$$

With

$$\boldsymbol{f}(\boldsymbol{d}) = [(d_1)^2 \;\; (d_2)^2 \;\; d_1d_2 \;\; d_1 \;\; d_2 \;\; 1]^T \tag{7.4}$$

Based on the theory of combinatorics (e.g. Lint and Wilson, 2001), the number of terms P in a N-degree polynomial is:

$$P(N,\ N_V) = \sum_{k=0}^{N} H_k^{N_V} = \sum_{k=0}^{N} \left(\frac{(k + N_V - 1)!}{k!\,(N_V - 1)} \right) \tag{7.5}$$

To determine the polynomial, the coefficients can be computed from the least square regression of the sample source terms. This requires the number of samples N_S to be no less than P. Also when $N_S = P$, the regression naturally becomes the interpolation. Based on the linear least square regression theory, there is the design matrix $\boldsymbol{F} = [\boldsymbol{f}^T(\boldsymbol{d}^1) \;\cdots\; \boldsymbol{f}^T(\boldsymbol{d}^{N_S})]^T$ calculated based on the design variables at the samples. For the c-th cell, the polynomial coefficients can be expressed as:

$$\boldsymbol{b}_c = (\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{F}^T\boldsymbol{S}_{c,S} \tag{7.6}$$

where the vector $\boldsymbol{S}_{c,S} = [S_c(\boldsymbol{d}^1) \;\cdots\; S_c(\boldsymbol{d}^{N_S})]^T$ contains the source term values of the c-th cells at the samples. Therefore, the source term for the c-th cell on the new design can be computed as:

$$S_c(\boldsymbol{d}) = \boldsymbol{f}^T(\boldsymbol{d})\boldsymbol{b}_c = \boldsymbol{f}^T(\boldsymbol{d})(\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{F}^T\boldsymbol{S}_{c,S} \tag{7.7}$$

The above calculation can also be expressed for all the cells together. Define the ensemble matrix of the sample source term values by stacking $\boldsymbol{S}_{c,S}^T$ columnwise as:

$$\boldsymbol{S}_s = \begin{bmatrix} \boldsymbol{S}_{1,S}^T \\ \vdots \\ \boldsymbol{S}_{N_{cell},S}^T \end{bmatrix} = [\boldsymbol{S}(\boldsymbol{d}^1) \ \cdots \ \boldsymbol{S}(\boldsymbol{d}^{N_S})]_{Ncell \times N_s} \tag{7.8}$$

The source term for all the cells of the new design is then expressed as:

$$\boldsymbol{S}(\boldsymbol{d}) = \begin{bmatrix} S_1(\boldsymbol{d}) \\ \vdots \\ S_{N_{cell}}(\boldsymbol{d}) \end{bmatrix}_{Ncell \times 1} = \begin{bmatrix} \boldsymbol{S}_{1,S}^T \boldsymbol{F}(\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{f}(\boldsymbol{d}) \\ \vdots \\ \boldsymbol{S}_{Ncell,S}^T \boldsymbol{F}(\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{f}(\boldsymbol{d}) \end{bmatrix} = \boldsymbol{S}_S\boldsymbol{F}(\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{f}(\boldsymbol{d}) \tag{7.9}$$

7.2.2 Kriging Interpolation

The Kriging can also be used to interpolate the source terms. The interpolation for the new design is based on the design variables $\boldsymbol{d}$. Based on the derivation in Chapter 3, the source term of a flow equation for the c-th cell is interpolated as:

$$S_c(\boldsymbol{d}) = \boldsymbol{f}^T\boldsymbol{\beta}_c + \boldsymbol{r}_c^T\boldsymbol{R}_c^{-1}(\boldsymbol{S}_{c,S} - \boldsymbol{F}\boldsymbol{\beta}_c) \tag{7.10}$$

$$\boldsymbol{\beta}_c = (\boldsymbol{F}^T\boldsymbol{R}_c^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}^T\boldsymbol{R}_c^{-1}\boldsymbol{S}_{c,S} \tag{7.11}$$

where $\boldsymbol{f}$, $\boldsymbol{F}$, $\boldsymbol{R}_c$, and $\boldsymbol{r}_c$ are functions of the design variables at the new and sample designs, defined in Chapter 3. $\boldsymbol{S}_{c,S} = [S_c(\boldsymbol{d}^1) \ \cdots \ S_c(\boldsymbol{d}^{N_s})]$ is the vector for the sample values. Definitions of the other terms are similar to those in Chapter 3. The interpolation can also be expressed as:

$$S_c(\boldsymbol{d}) = \boldsymbol{\omega}_c^T\boldsymbol{S}_{c,S} \tag{7.12}$$

where

$$\boldsymbol{\omega}_c = \left[\omega_{c,1} \ \cdots \ \omega_{c,N_S}\right]^T = \boldsymbol{R}_c^{-1}\boldsymbol{r}_c + \boldsymbol{R}_c^{-1}\boldsymbol{F}(\boldsymbol{F}^T\boldsymbol{R}_c^{-1}\boldsymbol{F})^{-1}(\boldsymbol{f} - \boldsymbol{F}^T\boldsymbol{R}_c^{-1}) \tag{7.13}$$

It can then be seen that the Kriging interpolation is a linear combination of the sample values.

7.3 POD Propagation Approach

The cell-to-cell propagation is conceptually straightforward. However, as the interpolation/approximation function has to be built N_{cell} times for each flow equation, for very large number of cells, it can be inefficient. To increase the efficiency, it is desirable to reduce the number of constructions of the propagation function. The cell-to-cell approach effectively treats the degree of freedom (DOF) of the source term of a flow equation as N_{cell}. However, as the different designs are similar, the source terms for them should contain some common patterns. If so, the degrees of freedom may be reduced by some reduce-order models (ROM).

In this study, the proper orthogonal decomposition (POD) is considered a good choice of ROM. POD, also known as the Karhunen-Loeve decomposition, empirical orthogonal eigen-functions, the principle component analysis (PCA) in different disciplines, is a widely used technique to extract the patterns or coherent structures in the data. The POD transforms a spatial distribution into a linear combination of a series of constant basis distributions called the POD bases/modes that are 'optimal' in approximating the samples of the distribution (or called "snapshots" in other literature). In this way the POD bases are thought to represent the underlying patterns, and can usually achieve a highly accurate approximation with only a small number of POD bases. As such, the POD/PCA has been applied widely, such as in data compression (Andrews et al., 1967), image processing (Rosenfeld and Kak, 2014), facial recognition (Yang et al., 2014), signal analysis (Ephraim and Trees, 1995), etc. In fluid mechanics, the POD was first used by Lumley (1967) to extract the coherence structures in the turbulence. A comprehensive description of the POD can be found in the book of Holmes et al. (2012).

Based on the above consideration, the source term distribution can also be cast into a linear combination of a series of POD bases that represent the common patterns of the source terms among the different designs. As such, the source term distribution is determined only by the few coefficients that weight the POD bases in the linear combination, instead of N_{cell} individual cell values. As a result, the DOF is reduced from N_{cell} to only a few, which can be as high as five to seven orders of magnitude lower.

7.3.1 POD Theory

A brief description of the POD theory is provided here that centres on the context of this study. To facilitate the discussion, assume that $S(\boldsymbol{x}, \boldsymbol{d})$ is a source term distribution evaluated at the design $\boldsymbol{d}$, where $\boldsymbol{x} = [x_1 \ \ x_2 \ \ x_3]^T$ is the spatial coordinates. Denote $\langle\cdot\rangle$ as the ensemble averaging operator given by:

$$\langle S(\boldsymbol{x}, \boldsymbol{d})\rangle = \frac{1}{N_S}\sum_{m=1}^{N_S} S(\boldsymbol{x}, \boldsymbol{d}^m) \tag{7.14}$$

where $S(\boldsymbol{x}, \boldsymbol{d}^m)$ is the source term of the m-th sample design. $|\cdot|$ is the modular. $\|\cdot\|$ is the L^2-norm defined as:

$$\|f\| = (f, f)^{\frac{1}{2}} \tag{7.15}$$

where $(\cdot\,,\ \cdot)$ denotes the inner product, which is the spatial integration of the function over the spatial domain:

$$(f, g) = \int_D f(\boldsymbol{x})g(\boldsymbol{x})\, d\boldsymbol{x} \tag{7.16}$$

The POD seeks to approximate $S(\boldsymbol{x}, \boldsymbol{d})$ as a linear combination of a finite number of bases as:

$$S(\boldsymbol{x}, \boldsymbol{d}) = \sum_{i=1} a_i(\boldsymbol{d}) \cdot \Phi_i(\boldsymbol{x}) \tag{7.17}$$

To compute the bases $\Phi_i(\boldsymbol{x})$, the ensemble of samples $\{S(\boldsymbol{x}, \boldsymbol{d}^m)\}$ are needed. They can be chosen by the design of experiment methods and evaluated from the experiments or the high-fidelity numerical simulations such as the LES.

It is wished that the source term distribution can be represented as accurately by as fewer bases as possible. Therefore the bases must be computed in some sense of optimality. In POD, an optimal basis is defined as the basis onto which the projected ensemble best approximates the original one by minimizing the mean squared error between the projected ensemble and the original ensemble. This is expressed as:

$$\min\left\langle\left\| S(\boldsymbol{x},\boldsymbol{d}) - \frac{\big(S(\boldsymbol{x},\boldsymbol{d}),\Phi(\boldsymbol{x})\big)\Phi(\boldsymbol{x})}{\|\Phi(\boldsymbol{x})\|^2}\right\|\right\rangle \tag{7.18}$$

To simplify the expression, it is often cast into an equivalent problem, which is to maximize the averaged "energy" of the ensemble projection on the POD basis. Also, the basis is required to be normalized. Thus, a POD basis is a solution to the maximization of the following Lagrange function:

$$J[\Phi] = \left\langle\left|\big(S(\boldsymbol{x},\boldsymbol{d}),\Phi(\boldsymbol{x})\big)\right|^2\right\rangle - \lambda(\|\Phi(\boldsymbol{x})\|^2 - 1) \tag{7.19}$$

For discrete distribution as in this study, the data of $S(\boldsymbol{x},\boldsymbol{d})$ and $\Phi(\boldsymbol{x})$ stored in the cells can be re-arranged into a $N_{cell} \times 1$ vector, sequenced according to the cell index. For example, for the m-th sample distribution, $S(\boldsymbol{x},\boldsymbol{d}^m)$ is equivalent to $\boldsymbol{S}(\boldsymbol{d}^m) = [s_1^m \ \cdots \ s_{Ncell}^m]^T$, where s_c^m is the value for the c-th cell on the m-th sample. And $\Phi(\boldsymbol{x})$ is equivalent to:

$$\boldsymbol{\varphi} = \left[\varphi_1 \ \cdots \ \varphi_{N_{cell}}\right]^T \tag{7.20}$$

As such, the discretized Lagrange function is expressed as:

$$J = \frac{1}{N_S}\sum_{m=1}^{N_S}\left(\sum_{i=1}^{N_{cell}} s_i^m\varphi_i\right)^2 - \lambda\left(\sum_{i=1}^{N_{cell}}\varphi_i^2 - 1\right) \tag{7.21}$$

This function contains multiple extrema. The solution to each of them is a POD basis. At each extremum, the derivative of the Lagrange function with respect to each component of the basis vector and the Lagrange multiplier λ must be zero, which is given by:

$$\frac{\partial J}{\partial \varphi_k} = \frac{1}{N_S}\sum_{m=1}^{N_S}\left[2\left(\sum_{i=1}^{N_{cell}} s_i^m \varphi_i\right)s_k^m\right] - 2\lambda\varphi_k \tag{7.22}$$

$$\frac{\partial J}{\partial \varphi_k} = \sum_{i=1}^{N_{cell}} \varphi_i^2 - 1 = 0 \tag{7.23}$$

Introducing the $N_{cell} \times N_{cell}$ correlation matrix computed based on tensor product as:

$$\boldsymbol{R} = \langle \boldsymbol{S}(\boldsymbol{d}) \otimes \boldsymbol{S}(\boldsymbol{d})\rangle = \frac{1}{N_S}\boldsymbol{S}_S\boldsymbol{S}_S^T = \begin{bmatrix} \ddots & & \\ & \frac{1}{N_S}\sum_{m=1}^{N_S}\left(s_i^m s_j^m\right) & \\ & & \ddots \end{bmatrix} \tag{7.24}$$

where the $\boldsymbol{S}$ is the $N_{cell} \times N_S$ ensemble matrix $\boldsymbol{S}_S = [\boldsymbol{S}(\boldsymbol{d}^1) \ \cdots \ \boldsymbol{S}(\boldsymbol{d}^{N_S})]$.

Thus, it can be shown that the solutions to the above zero-gradient problem of Equation **(7.22)** and **(7.23)** are the eigenvectors and eigenvalues of the following eigenvalue problem:

$$\boldsymbol{RQ} = \boldsymbol{Q\Lambda} \tag{7.25}$$

where $Q = \left[\varphi_1 \ \cdots \ \varphi_{N_{cell}}\right]$ contains the eigenvectors $\boldsymbol{\varphi}_i$. The diagonal matrix $\boldsymbol{\Lambda}$ is:

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{N_{cell}} \end{bmatrix} \tag{7.26}$$

It contains the eigenvalues λ_i corresponding to each eigenvector $\boldsymbol{\varphi}_i$. Thus, the POD bases are the eigenvectors $\boldsymbol{\varphi}_i$ of the correlation matrix $\boldsymbol{R}$. As $\boldsymbol{R}$ is positive semi-definite, the eigenvalues λ_i are real and non-negative. Also, as $\boldsymbol{R}$ is symmetric, based on the Hilbert-Schmidt theory, the eigenvectors are to be orthogonal to each other. To satisfied Equation **(7.23)**, the eigenvectors are further scaled to be orthonormal. Hence the POD bases are orthonormal to each other. Once $\boldsymbol{\varphi}$ is solved, $\Phi(\boldsymbol{x})$ is obtained naturally by rearranging the data.

It can be shown that $\lambda_i = \delta_{ij}\langle a_i(\boldsymbol{d})a_j(\boldsymbol{d})\rangle = \delta_{ij}\frac{1}{N_S}\left(a_i(\boldsymbol{d}^m)a_j(\boldsymbol{d}^m)\right)$ (Holmes et al., 2012). So, the eigenvalue λ_i is the maximized averaged "energy" of the ensemble projection on the basis, i.e. the maximum of Equation **(7.21)** under the orthonormal constrain. As such, a larger eigenvalue corresponds to the basis that better approximate the ensemble. For the POD expansion:

$$\boldsymbol{S}(\boldsymbol{d}) = \sum_{i=1}^{N_{cell}} a_i(\boldsymbol{d})\boldsymbol{\varphi}_i \tag{7.27}$$

The terms in the expansion are usually arranged with descending eigenvalues $(\lambda_i \geq \lambda_{i+1})$. So, if a truncated series with only n terms are used for approximation, it captures the largest amount of the averaged "energy" of the ensemble projection. Also, it has been shown by Homes et al. (2012) that for any n, the first n POD bases arranged this order captures more "energy" on average than the first n terms of a linear expansion using any other set of orthogonal bases.

7.3.2 POD Computation by the SVD

Solving the POD bases from a direct solution of the eigenvalue problem of Equation (7.25) is usually computationally expensive, as the dimension of the correlation matrix can be extremely large due to the large number of cells (usually millions) in a typical computational domain of the present work. Therefore, efficient calculation methods have been developed for large dataset. One of them is the singular value decomposition (SVD) (e.g. Lay et al., 2016).

The efficiency of the SVD method lies in that, as it has well-developed computation algorithms, even if the ensemble matrix $\boldsymbol{S}_S = [\boldsymbol{s}^1 \ \cdots \ \boldsymbol{s}^{N_S}]$ is very large, the POD bases can still be computed very fast, e.g. a few seconds in this study. The POD bases are only computed once and can be used for all other new designs, provided that the ensemble matrix does not take in new samples.

The ensemble matrix is decomposed by SVD as:

$$\boldsymbol{S}_S = \boldsymbol{U\Sigma V^T} \tag{7.28}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1 \;\; \cdots \;\; \boldsymbol{u}_{N_{cell}}]$ and $\boldsymbol{V} = [\boldsymbol{v}_1 \;\; \cdots \;\; \boldsymbol{v}_{N_{cell}}]$ are unitary matrices (i.e. $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_{N_{cell}\times N_{cell}}$, $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_{N_S\times N_S}$). $\boldsymbol{\Sigma}$ is the $N_{cell} \times N_s$matrix containing the singular values:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_1 \\ \boldsymbol{0} \end{bmatrix}, \boldsymbol{\Sigma}_1 = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{N_S} \end{bmatrix} \tag{7.29}$$

in which $\boldsymbol{\Sigma}_1$ is a diagonal matrix. The singular values σ_i are real and non-negative and sorted in descending order. As the sample distributions are generally linearly independent, the rank of the ensemble matrix is N_S, and the singular values are all positive. The matrix $\boldsymbol{U}$ can be partitioned as $\boldsymbol{U} = [\boldsymbol{U}_1 \;\; \boldsymbol{U}_2]$, where the $N_{cell} \times N_S$ matrix $\boldsymbol{U}_1$ is:

$$\boldsymbol{U}_1 = [\boldsymbol{u}_1 \;\; \cdots \;\; \boldsymbol{u}_{N_S}] \tag{7.30}$$

in which all the $\boldsymbol{u}_i$ are mutually orthonormal.

Substituting the SVD (Equation **(7.28)**) into $\boldsymbol{R}$ (Equation **(7.24)**), and right-multiplied by $\boldsymbol{U}_1$, one obtains:

$$\boldsymbol{RU}_1 = \frac{1}{N_S}\boldsymbol{SS}^T\boldsymbol{U}_1 = \frac{1}{N_S}[\boldsymbol{U}_1 \;\; \boldsymbol{U}_2]\begin{bmatrix} \boldsymbol{\Sigma}_1 \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{V}^T\boldsymbol{V}[\boldsymbol{\Sigma}_1 \;\; \boldsymbol{0}]\begin{bmatrix} \boldsymbol{U}_1^T \\ \boldsymbol{U}_2^T \end{bmatrix}\boldsymbol{U}_1 = \boldsymbol{U}_1\frac{\boldsymbol{\Sigma}_1^2}{N_S} \tag{7.31}$$

It can then be seen immediately that the first N_S eigenvectors of $\boldsymbol{R}$ are the columns of $\boldsymbol{U}_1$ and the first N_S eigenvalues are the squared singular values divided by N_S, i.e. $\boldsymbol{\varphi}_i = \boldsymbol{u}_i$ and $\lambda_i = \sigma_i^2/N_S$ $(1 \leq i \leq N_S)$. As the rank of the ensemble matrix is N_S, so is that of $\boldsymbol{R}$. Thus the remaining $N_{cell} - N_S$ eigenvalues are zero. The number of eigenvectors required to span the subspace spanned by the ensemble is then the rank of the ensemble N_S.

In summary, the POD can be efficiently computed by the SVD of the ensemble matrix and taking the first N_S columns of $\boldsymbol{U}$ as the POD bases. The values of the POD coefficient of the i-th basis at the samples can be expressed as:

$$\boldsymbol{a}_{i,S} = \begin{bmatrix} a_i(\boldsymbol{d}^1) \\ \vdots \\ a_i(\boldsymbol{d}^{N_S}) \end{bmatrix} = \sigma_i \boldsymbol{v}_i \quad \textbf{(7.32)}$$

7.3.3 POD for Source Term Propagation

Once the POD bases are constructed, they are applicable to all the designs in the design space. The dimension of the source term is then reduced to the number of POD coefficients which is no more than N_S. The source term on the new design is form by the POD expansion, with the coefficients propagated from the sample values. A propagation function needs to be constructed for each POD coefficient respectively.

A. Polynomial Regression

The polynomial regression can be used to propagate the POD coefficients. For each POD coefficient, a polynomial is constructed respectively. The k-th POD coefficient at the new design is calculated based on the design variable $\boldsymbol{d}$ as:

$$a_k(\boldsymbol{d}) = \sum_{p=1}^{P} \left[b_{p,k} \prod_{i=1}^{N_V} (d_i)^{n_{i,p}} \right], (k \le N_S) \quad \textbf{(7.33)}$$

or concisely,

$$a_k(\boldsymbol{d}) = \boldsymbol{f}^T \boldsymbol{b}_k \quad \textbf{(7.34)}$$

$b_{p,k}$ is the coefficient of the p-th term in the polynomial for the k-th POD coefficient, contained in vector $\boldsymbol{b}_k = [b_{1,k} \ \cdots \ b_{P,k}]^T$. d_i is the i-th design variable. $n_{i,p}$ is the integer power of d_i in the p-th polynomial term, where $n_{i,p} \ge 0$ and $\sum_{i=1}^{N_V} n_{i,p} \le N$ for each p. $\boldsymbol{f}^T(\boldsymbol{d})$ is a $P \times 1$ vector containing the polynomial bases functions.

Based on the linear least square regression theory, the polynomial coefficients for the k-th POD coefficient can be calculated as:

$$\boldsymbol{b}_k = (\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{F}^T\boldsymbol{a}_{k,S} \tag{7.35}$$

where $\boldsymbol{F} = [\boldsymbol{f}^T(\boldsymbol{d}^1) \ \cdots \ \boldsymbol{f}^T(\boldsymbol{d}^{Ns})]^T$ is the $N_S \times P$ design matrix calculated based on the design variables at the samples. $\boldsymbol{a}_{k,S} = [a_k(\boldsymbol{d}^1) \ \cdots \ a_k(\boldsymbol{d}^{Ns})]^T$ contains the values of the k-th POD coefficient at the samples. The k-th POD coefficient at the new design is thus calculated as:

$$a_k(\boldsymbol{d}) = \boldsymbol{f}^T(\boldsymbol{d})\boldsymbol{b}_k = \boldsymbol{f}^T(\boldsymbol{d})(\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{F}^T\boldsymbol{a}_{k,S} \tag{7.36}$$

Also, all the POD coefficients can be expressed as a whole as:

$$\boldsymbol{a}(\boldsymbol{d}) = \begin{bmatrix} a_1(\boldsymbol{d}) \\ \vdots \\ a_{N_S}(\boldsymbol{d}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_{1,S}^T(\boldsymbol{F}^{-1})^T\boldsymbol{f}(\boldsymbol{d}) \\ \vdots \\ \boldsymbol{a}_{N_{cell},S}^T(\boldsymbol{F}^{-1})^T\boldsymbol{f}(\boldsymbol{d}) \end{bmatrix} = \boldsymbol{A}_S\boldsymbol{F}(\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{f}(\boldsymbol{d}) \tag{7.37}$$

where $\boldsymbol{A}_S = \boldsymbol{\Sigma}_1\boldsymbol{V}^T = \left[\boldsymbol{a}_{1,S}^T \ \cdots \ \boldsymbol{a}_{k,S}^T \ \cdots \ \boldsymbol{a}_{N_S,S}^T\right]^T$ is the ensemble of all the POD coefficients at the samples.

Therefore, on the new design, the source term values in all the cells are simultaneously computed as:

$$S(\boldsymbol{d}) = \sum_{k=1}^{N_S} a_k(\boldsymbol{d}) \cdot \boldsymbol{\varphi}_k(\boldsymbol{x}) = \boldsymbol{U}_1\boldsymbol{a}(\boldsymbol{d}) \tag{7.38}$$

B. Kriging Interpolation

The Kriging method can also be used to interpolate the POD coefficients on the new design. For each POD coefficient, a Kriging interpolant is constructed respectively. The k-th POD coefficient is calculated based on the design variable $\boldsymbol{d}$ as:

$$a_k(\boldsymbol{d}) = \boldsymbol{f}^T\boldsymbol{\beta}_k + \boldsymbol{r}_k^T\boldsymbol{R}_k^{-1}\left(\boldsymbol{a}_{k,S} - \boldsymbol{F}\boldsymbol{\beta}_k\right) \tag{7.39}$$

$$\boldsymbol{\beta}_k = \left(\boldsymbol{F}^T\boldsymbol{R}_k^{-1}\boldsymbol{F}\right)^{-1}\boldsymbol{F}^T\boldsymbol{R}_k^{-1}\boldsymbol{a}_{k,S} \tag{7.40}$$

where $\boldsymbol{f}$, $\boldsymbol{F}$, $\boldsymbol{R}_k$ and $\boldsymbol{r}_k$ are functions of $\boldsymbol{d}$ at the new design and sample designs. $\boldsymbol{a}_{k,S}$ is the sample values for the k-th POD coefficient.

The POD coefficient can also be expressed as a linear combination of the sample values as:

$$a_k(\boldsymbol{d}) = \boldsymbol{\omega}_k^T \boldsymbol{a}_{k,S} \tag{7.41}$$

where:

$$\boldsymbol{\omega}_k = \boldsymbol{R}_k^{-1}\boldsymbol{r}_k + \boldsymbol{R}_k^{-1}\boldsymbol{F}\left(\boldsymbol{F}^T\boldsymbol{R}_k^{-1}\boldsymbol{F}\right)^{-1}\left(\boldsymbol{f} - \boldsymbol{F}^T\boldsymbol{R}_k^{-1}\right) \tag{7.42}$$

After the coefficients are calculated, source term values in all the cells are computed simultaneously as:

$$S(\boldsymbol{d}) = \sum_{k=1}^{N_S} a_k(\boldsymbol{d}) \cdot \boldsymbol{\varphi}_k(\boldsymbol{x}) = \boldsymbol{U}_1\boldsymbol{a}(\boldsymbol{d}) \tag{7.43}$$

7.4 Implementation Process

The detail implementation process to propagate the source terms and use them for the steady-state solution is presented below.

Step 1: store the source terms data extracted from the time-mean LES solutions of the samples and stored in text file

Step 2 - compute new source terms: **a)** Load all the extracted sample source terms to the workspace of Matlab. **b)** Perform the source term propagation based on the values of the design variables of the new design with the selected Matlab code based on the selected propagation method and approach. **c)** Save the computed new source terms into the text file.

Step 3 - case setting for the fluent solver: **a)** Choose the steady-state density-based solver, set the solver setting the same as the that used in the source term extraction (as discussed in Chapter 6), including the spatial discretization, boundary conditions, viscosity setting (the SA damper model with same options and coefficients as in Chapter 6), etc. **b)**

However, the solver can be set to implicit type with pseudo-transient option. This will require a much fewer number of iterations (say 300) to get the steady-state solution converged. **c)** Compile and load the relevant UDF into the case. **d)** Enable source term option in the cell-zone condition, hook the UDF that links the source terms of the equation with the user-defined-memory (UDM), where the source term data are stored. **e)** Activate the flow equations and the SA damper model transport equations, as both are to be solved together.

Step 4 - flow solution: **a)** load the propagated source terms and store them in the allocated UDM (already linked to the source terms by Step 3), via a UDF. The source term for each flow equation will need one UDM. Each UDM will have the same number of data points as the number of cells, with each data point linked to a cell. **b)** Initialize the flow field and damping viscosity μ_d with some reasonable value, and start the iteration. e) Finally, with the effect of the source terms and the damping viscosity, the steady-state solution should be able to converge quickly after around 300 iterations.

Step 5: post-process the solution data and feed the required quantities back into the optimization procedure.

The C and Matlab programming codes for the above procedure were intended to but weren't finally included as an appendix due to the page limit of the present thesis.

7.5 Validation Results

The validation for the various source term propagation methods are carried out for both the cell-to-cell approach and POD approach. The case chosen for the validation is the trailing edge cutback injection configuration in Chapter 5 and Chapter 6. To keep focused, some other aspects are simplified. In total, only two design variables are chosen. The taper angle of the land is chosen as the shape design variables, ranging from 0° to 5°. The blowing ratio (BR) is chosen as the flow control design variables, ranging from 1 to 1.3. A 3 by 3 uniform distribution of the samples are chosen, as shown in Figure 7-4 below.

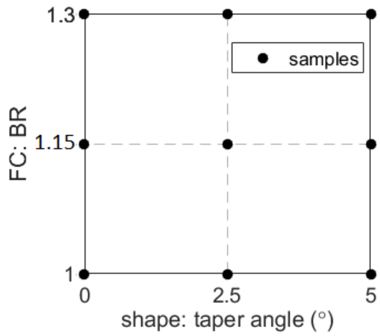


Figure 7-4 Samples in the 2D design space used for the validation

In the following validation, the steady-state solution with the propagated source terms at a new test design are compared to the reproduced LES solution of that test design by the steady-state solution that uses the LES-extracted source terms (as explained in Chapter 6), so as to consistently distinguish the propagation error from the reproduction error. From Chapter 6 it can be assured that the reproduction error is always negligibly small. So the steady-state solutions with the LES-extracted source terms for the comparison of this chapter can be regarded as to the time-mean of the direct LES solutions, and will simply be called the “LES”.

7.5.1 Solution Sensitivity to the Source Terms

Firstly the author would like to address the issue whether the steady-state solution is sensitive enough to the variation of the source terms. Because otherwise, it could be possible that, for instance, the source terms of a particular design can be used unchanged on another design to sufficiently provide the accurate steady-state solutions there, as the accuracy will chiefly be accounted for by the geometry and boundary conditions of those designs. If that’s the case, the importance of the propagation methods to adapt the source terms will diminish.

To investigate this, the source terms of the sample design at the centre of the design space (i.e. 2.5° taper angle and $BR = 1.15$) are applied unchanged to the other samples respectively, as shown in Figure 7-5 below. The other samples are chosen such that the effects of the difference in the shape and the flow control are studied both separately and altogether.

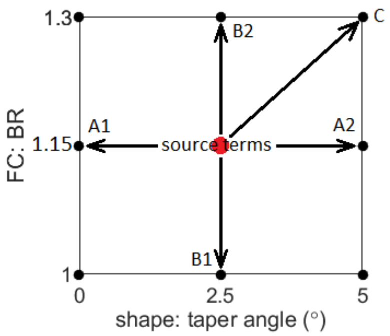


Figure 7-5 Central sample source terms applied to other samples

First, the source terms of the central sample are applied to the samples of different taper angles (i.e. Sample A1 and A2 of Figure 7-5). The resulting steady-state solutions are shown in Figure 7-6 below. For both A1 and A2, Figure 7-6b shows the η_{ad} are quite similar to that of the central sample, despite being stretched by the geometries. Both η_{ad} also deviates significantly from their time-mean LES solutions (Figure 7-6a, reproduced by the steady-state solutions with the LES-extracted source terms). This shows that the source terms can exert more influence than the taper angle on the steady-state solutions.

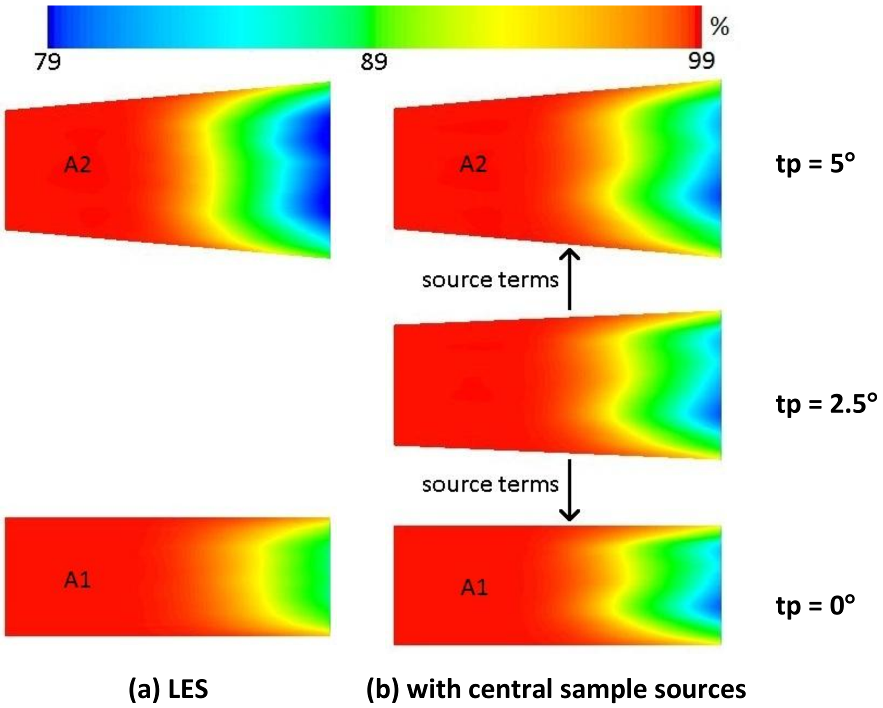


Figure 7-6 Source term sensitivity on varied taper angle (Sample A1 and A2), regarding adiabatic effectiveness η_{ad} on the breakout surface

Next, the source terms of the central sample are also applied to the two samples with different blowing ratios (i.e. Sample B1 and B2 of Figure 7-5). The resulting steady-state solutions are compared in Figure 7-7 below. The results show that the source terms change the steady-state solutions of the two samples significantly compared to their LES solutions (Figure 7-7a). However the steady-state solutions are still not close enough to that of the central sample. This indicates that source terms are influential enough on steady-state solutions, though not much more than the currently chosen range of blowing ratio.

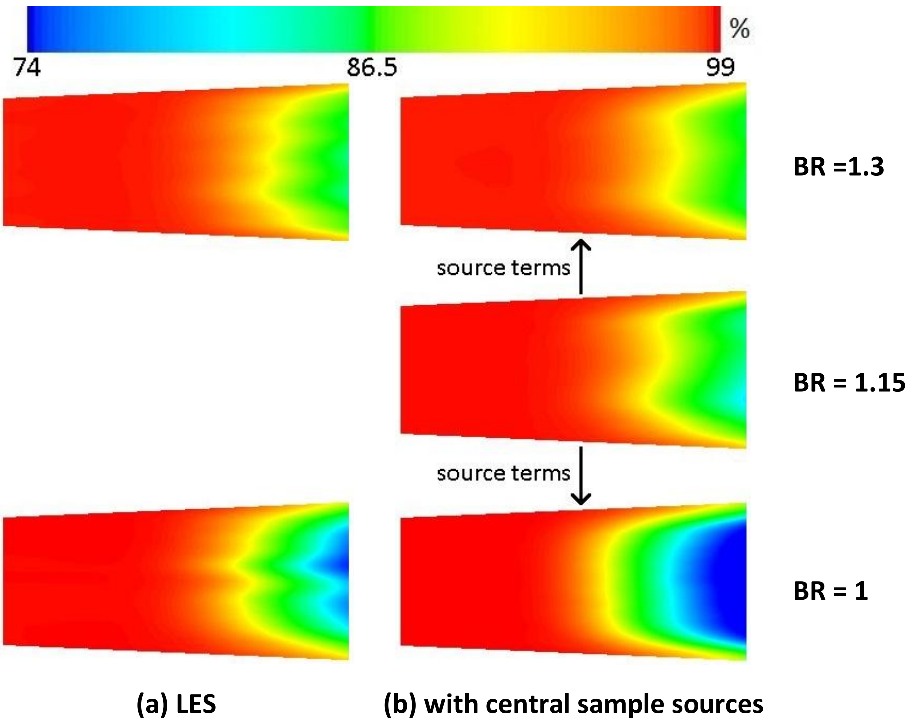


Figure 7-7 Source term sensitivity on varied blowing ratio (Sample B1 and B2), regarding adiabatic effectiveness η_{ad} on the breakout surface

Finally, the source terms of the central sample are also applied to the sample with both taper angle and blowing ratio varied (i.e. Sample C of Figure 7-5). In Figure 7-8 below it shows that the resulting steady-state solution deviates obviously from its LES solution (Figure 7-8a), but is also quite different from the solution of the central sample.

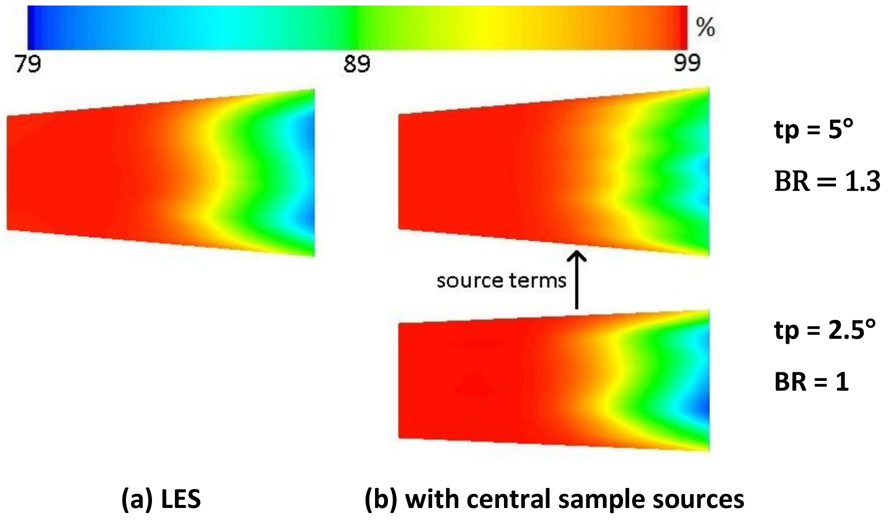


Figure 7-8 Steady-state solutions of the design with different taper angle and blowing ratio using the source terms of the central sample

All the above results show that the source terms are as influential as the geometry and the boundary conditions, and play an important role in making the steady-state solution accurate. So it is important that they should be adapted properly during the propagation.

7.5.2 Validation of the Propagation Methods

In the following, the propagation methods to be validated are the quadratic regression carried out in the cell-to-cell-approach (CTC-Quad), and the kriging interpolation carried out in the POD approach (POD-Krg).

The quadratic regression in the POD approach is not validated since it can be shown that it will produce the same results as the cell-to-cell quadratic regression. From Equation (7.9), the source term values for all the cells computed by the cell-to-cell quadratic regression are expressed together as:

$$\boldsymbol{S}(\boldsymbol{d}) = \boldsymbol{S}_S \boldsymbol{F}(\boldsymbol{F}^T \boldsymbol{F})^{-1} \boldsymbol{f}(\boldsymbol{d})$$

Whereas for the POD approach, from Equation (7.37) shown previously, all the POD coefficients can be expressed as a whole as:

$$\boldsymbol{a}(\boldsymbol{d}) = \boldsymbol{A}_S \boldsymbol{F}(\boldsymbol{F}^T \boldsymbol{F})^{-1} \boldsymbol{f}(\boldsymbol{d})$$

And since by POD linear expansion:

$$\boldsymbol{S}(\boldsymbol{d}) = \boldsymbol{U}_1\boldsymbol{a}(\boldsymbol{d}) = \boldsymbol{U}_1\boldsymbol{A}_S\boldsymbol{F}(\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{f}(\boldsymbol{d}) = \boldsymbol{S}_S\boldsymbol{F}(\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{f}(\boldsymbol{d})$$

One can see that the computed source term values by the two approaches are identical in the polynomial regression.

Also, the Kriging interpolation carried out in the cell-to-cell approach is not adopted, because some testing shows that it would take more than one day to compute all the source terms for a new design, which is 8 times longer than a steady-state simulation. While the Kriging interpolation carried out in the POD approach only takes several seconds. To be specific, assume that all the 9 POD bases derived using 9 samples in this validation case are used to propagated the 5 source terms for a new design, then there are 45 kriging models to be constructed for the 45 POD coefficients respectively. This will take 0.3 seconds in this work. Scaling the computational speed linearly, for Kriging source term interpolation in the cell-to-cell approach, $5N_{cell}$ Kriging models will be needed for each source term value in each mesh cell. This will require about 33 hours, which is a computational cost nearly 400,000 times larger than the Kriging source term interpolation in the POD approach. Thus it can be seen that the POD approach can be a much more efficient approach to propagate the source terms.

Some integral quantities used in the validation are defined here. First, the mean of the adiabatic effectiveness η_{ad} on the breakout surface is area-weighted averaged over the rear $2/3$ portion, as illustrated in Figure 7-9 below, because η_{ad} on the front $1/3$ part is shown to be almost constant in all the designs.

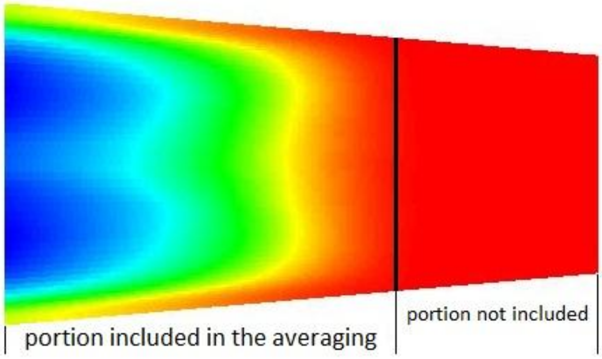


Figure 7-9 Portion of the breakout surface over which η_{ad} is averaged (viewed along y-axis)

The second integral quantity is the loss of the flow field. Two losses are defined. The first is the aerodynamic loss. The second is the thermal dynamic loss to take into account of the energy expenditure of the injection. The aerodynamic loss coefficient is defined as usual:

$$\zeta_{aero} = \frac{p_{t1} - p_{t2}}{p_{t2} - p_2} \quad \textbf{(7.44)}$$

For the thermal dynamic loss, based on the work of Osnaghi et al. (1997), the definition of the coefficient is expressed as:

$$\begin{aligned}\zeta_{th} &= 1 - \frac{\text{actual exit kinetic energy (KE)}}{\text{isentropic mainstream KE at exit} + \text{isentropic injection KE at exit}} \\ &= 1 - \frac{(\dot{m}_1 + \dot{m}_c)(p_{t2} - p_2)}{\dot{m}_1(p_{t1} - p_2) + \dot{m}_c(p_{tc} - p_2)}\end{aligned} \quad \textbf{(7.45)}$$

In above loss definitions, p_{t1} is taken to be the averaged total pressure at the test section inlet in the full-airfoil simulation (see Chapter 5), so as to be consistent with the common loss calculation that takes into account of the whole airfoil or blade. p_{t2} and p_2 are the averaged total and static pressure at the LES domain outlet. p_{tc} is the averaged total pressure at the coolant chamber inlet. The above averaging for the pressures are all mass-flow-averaged. $\dot{m}_1$ and $\dot{m}_c$ are the total mass flow rates of the main flow and the coolant injection. To achieve consistency, for the time-mean LES solution, the pressures to be averaged are the time-mean pressures, and the mass flow rates are calculated based on the time-mean velocity. For the steady-state solution, the averaged pressures and the mass flow rates are simply calculated from the solution. One can also see that when the injection mass flow rate is zero, the thermal dynamic loss naturally becomes the aerodynamic loss.

A. Variation of the Taper Angle

In this section, the source terms are propagated to the new designs with different taper angles. The blowing ratio stays at $BR = 1.3$. The three samples from which the source terms are propagated have the taper angles of 0°, 2.5° and 5°. The new designs are chosen with taper angle of 1.25° and 3.75°, at the midpoint of the sample intervals. In the cell-to-cell propagation approach, the quadratic polynomial is chosen as the interpolant. In the POD propagation approach, the Kriging model is chosen as the interpolant.

Table 7-1 Comparison of predicted performance between LES and the steady-state solutions at the new designs with different taper angles

Taper	mean η_{ad}			ζ_{th}			ζ_{aero}		
	LES	CTC-Quad	POD-Krg	LES	CTC-Quad	POD-Krg	LES	CTC-Quad	POD-Krg
0°	0.9566			0.1961			0.03826		
1.25°	0.9492	0.9477	0.9477	0.1956	0.1954	0.1954	0.03844	0.03844	0.03844
2.5°	0.9392			0.1949			0.03855		
3.75°	0.9301	0.9312	0.9312	0.1941	0.1942	0.1942	0.03855	0.03842	0.03842
5°	0.9237			0.1940			0.03844		

The main performances, which are the mean adiabatic effectiveness η_{ad} on the rear breakout surface, the thermal dynamic loss ζ_{th} and the aerodynamic loss ζ (Equation **(7.44)**), are compared in Table 7-1 above, between the LES and the steady-state solutions with CTC-Quad and POD-Krg source term propagations. Variations of the above performance with respect to the taper angle are also plotted below in Figure 7-10 for the mean adiabatic effectiveness and Figure 7-11 for the thermal dynamic loss. Both plots show that the performance of the new designs are predicted well by the steady-state solution using both ways of source term propagation, as seen by the close match with the direct LES solutions of the new designs. But the difference between the two source term propagation methods: CTC-Quad and the POD-Krg are almost absent, which might due to the relatively simple variation of the performances with this range of taper angle, though the two methods are not mean to always produce the same results.

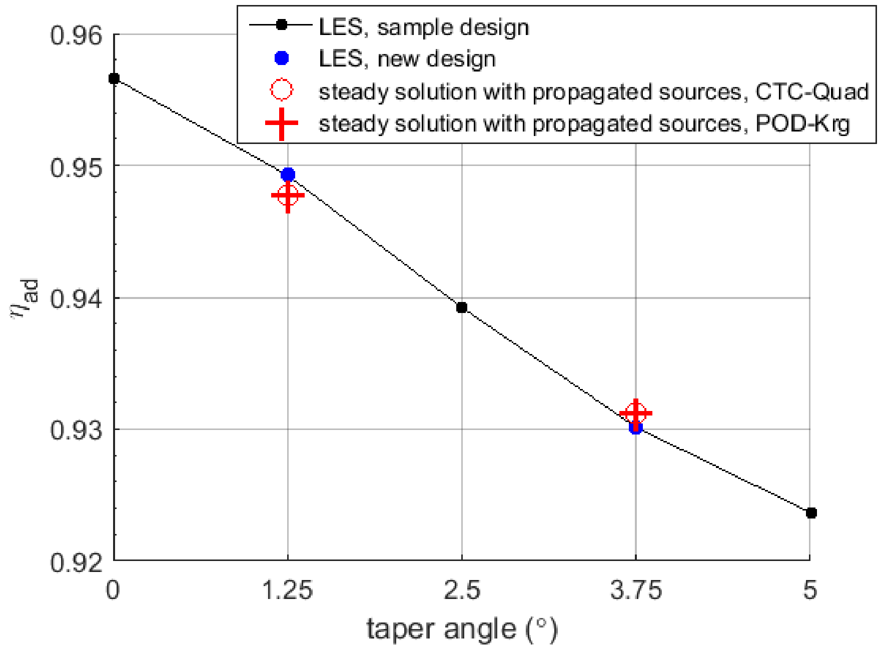


Figure 7-10 Variation of mean η_{ad} on breakout surface at different taper angles

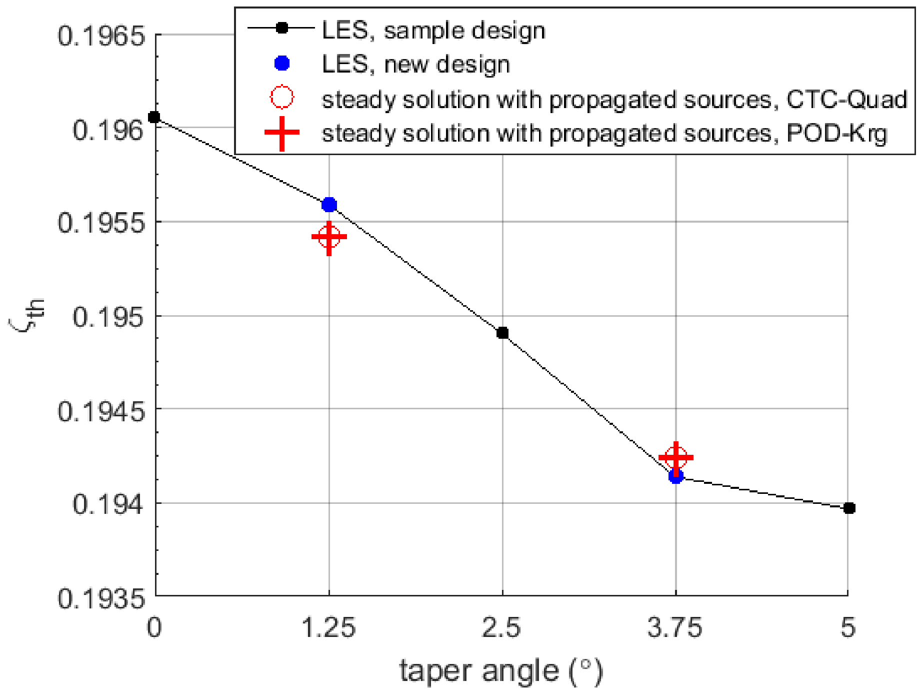


Figure 7-11 Variation ζ_{th} at different taper angles

Apart from the mean quantities, the contours of the flow fields also compared in Figure 7-12 regarding the adiabatic effectiveness on the breakout surface. The LES solutions show that the η_{ad} decreases as the taper angle increases, particularly at the end of the surface. This is considered due to the enlarged exposure area of the breakout surface that increases the mixing between the mainstream and the coolant. The steady-state solutions with both types of propagated source terms fit well into this trend of variation of the LES solution. At both new designs, the contour distributions of η_{ad} are quite similar to the LES prediction in terms of both the magnitude and the shape.

The source terms extracted from the LES as well as propagated by the two methods are compared in Figure 7-13, in which the energy source on the plane passing through the mid-heights of the slot and chamber is presented. The window is shifted half period in z-direction to center at the midspan of the land. Only the part that shows noticeable differences among the designs is displayed. It can be seen that the LES-extracted source terms show common features among the designs. Also, a clear trend of variation with the taper angle can be seen, especially in the breakout region and downstream of the land. The propagated source terms fit well, and match well with the LES-extracted source terms in term of both the overall magnitude and the distribution.

Overall, the validation shows that source term propagations are able to adapt to the change of the taper angle and produce accurate enough source terms for the new designs. As such, the predicted performance and the flow field from the steady-state solution match well with the LES prediction at the designs with new taper angles. However, the two propagation methods do not show obvious difference, probably due to the relatively simple variation of the flow field with only the difference in the taper angle.

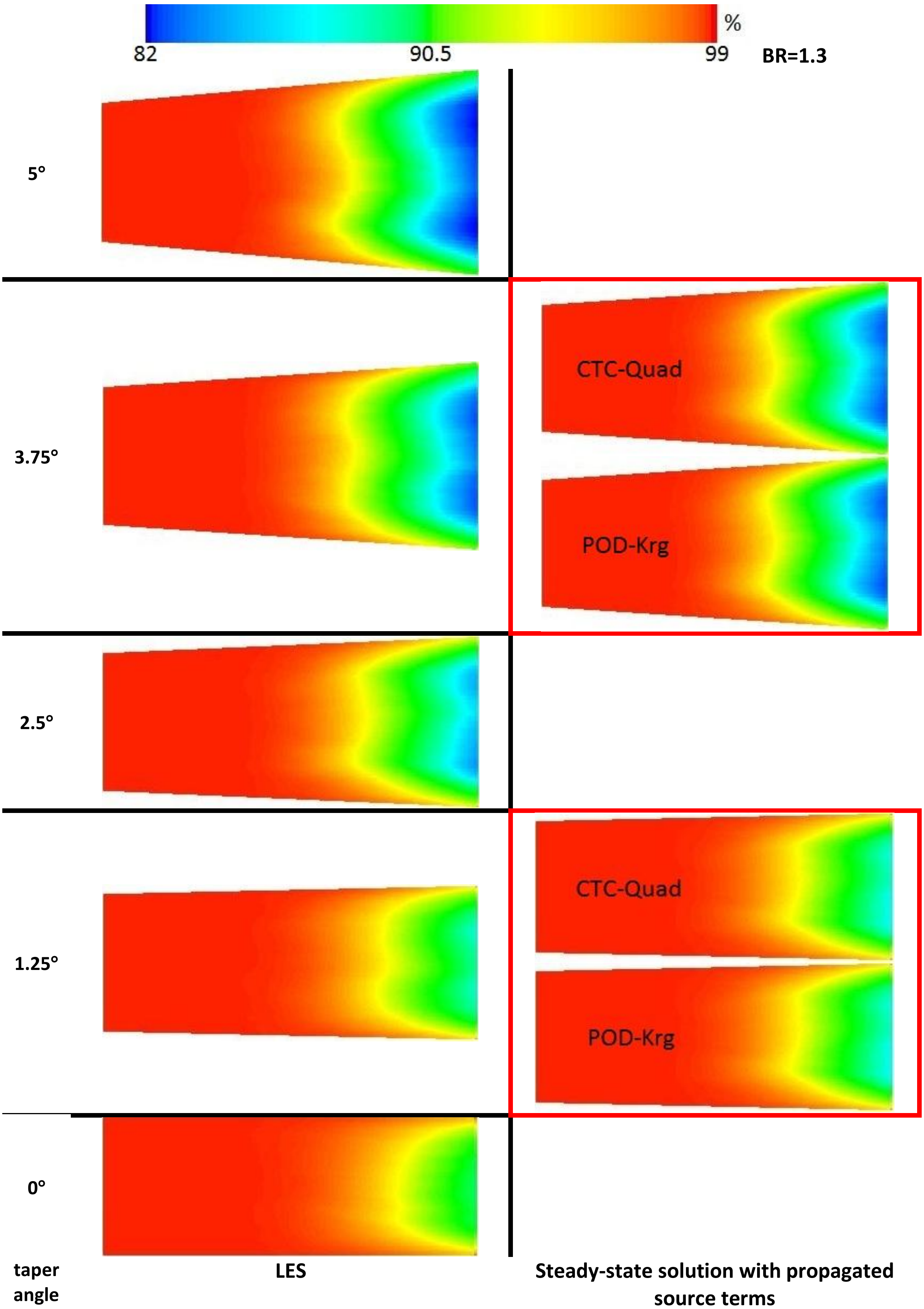


Figure 7-12 Comparison of η_{ad} on breakout at various taper angles

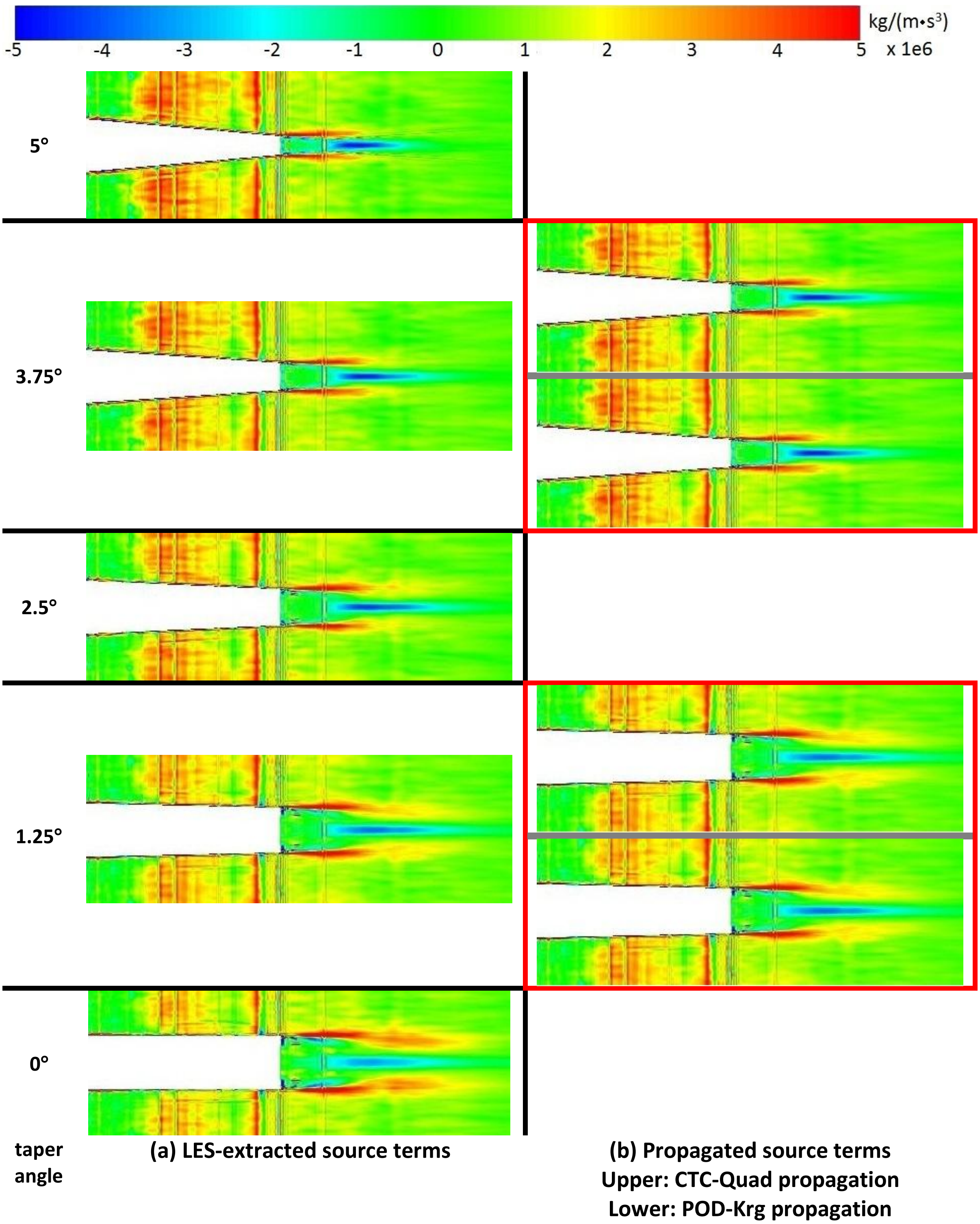


Figure 7-13 Propagated source terms (for the energy equation) compared with LES-extracted source terms

B. Different Blowing Ratios

In this section the source terms are propagated to the new designs with different blowing ratios. The taper angle stays at 2.5°. The three samples from which the source terms are propagated have the blowing ratios of 1, 1.15 and 1.3. The new design to be validated has the blowing ratio of 1.075. In the cell-to-cell propagation approach, the quadratic polynomial is chosen as the interpolant. In the POD propagation approach, the Kriging model is chosen as the interpolant.

Table 7-2 Comparison of predicted performance between LES and the steady-state solutions at the new designs with different blowing ratios

BR	mean η_{ad}			ζ_{th}			ζ_{aero}		
	LES	CTC-Quad	POD-Krg	LES	CTC-Quad	POD-Krg	LES	CTC-Quad	POD-Krg
1.3	0.939			0.1949			0.03855		
1.15	0.884			0.1325			0.06673		
1.075	0.804	0.843	0.843	0.1128	0.1131	0.1131	0.07958	0.08016	0.08016
1	0.785			0.1021			0.09350		

The main performances between the LES and the steady-state solutions are compared in Table 7-2 above. Variations of the above performances with respect to the blowing ratio are plotted in Figure 7-14 and Figure 7-15 below. In Figure 7-14, the adiabatic effectiveness from the steady-state solution using the propagated source terms fit closely with the LES prediction. In Figure 7-15, the thermal dynamic loss from the steady-state solution also fit well to LES prediction. The results of both the CTC-Quad and the POD-Krg propagations are quite the same, which may also be attributed to the simple variations of the performance with the only the blowing ratio being different. As the blowing ratio increases, the aerodynamic loss decreases. This can be attributed to the filling of the wake momentum deficit by the injection. But instead the thermodynamic loss increases, indicating that the reduction of the aerodynamic loss comes at the cost of the energy expenditure of the injection.

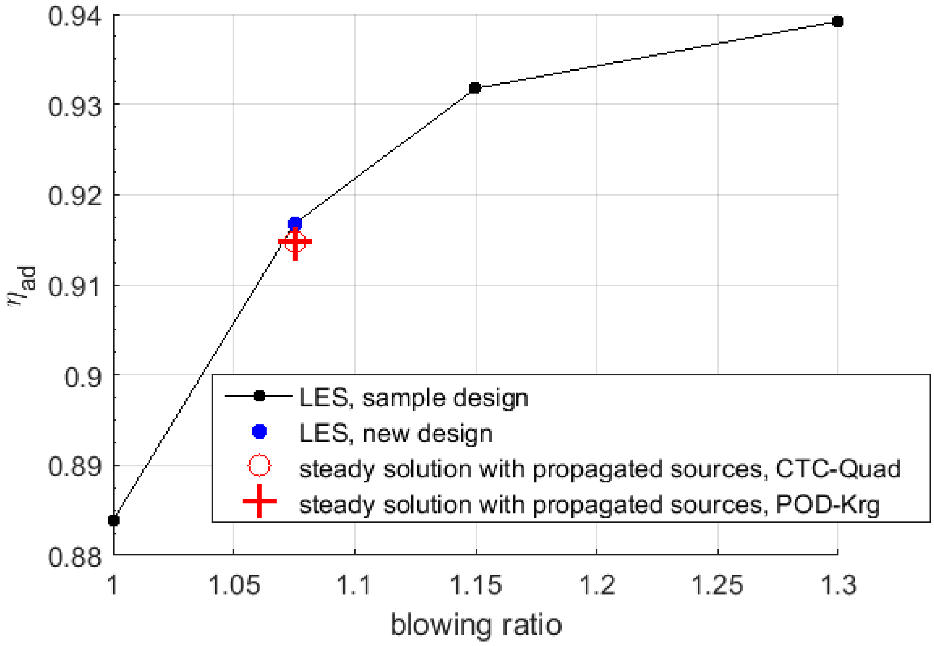


Figure 7-14 Variation of mean η_{ad} on breakout surface at different blowing ratios

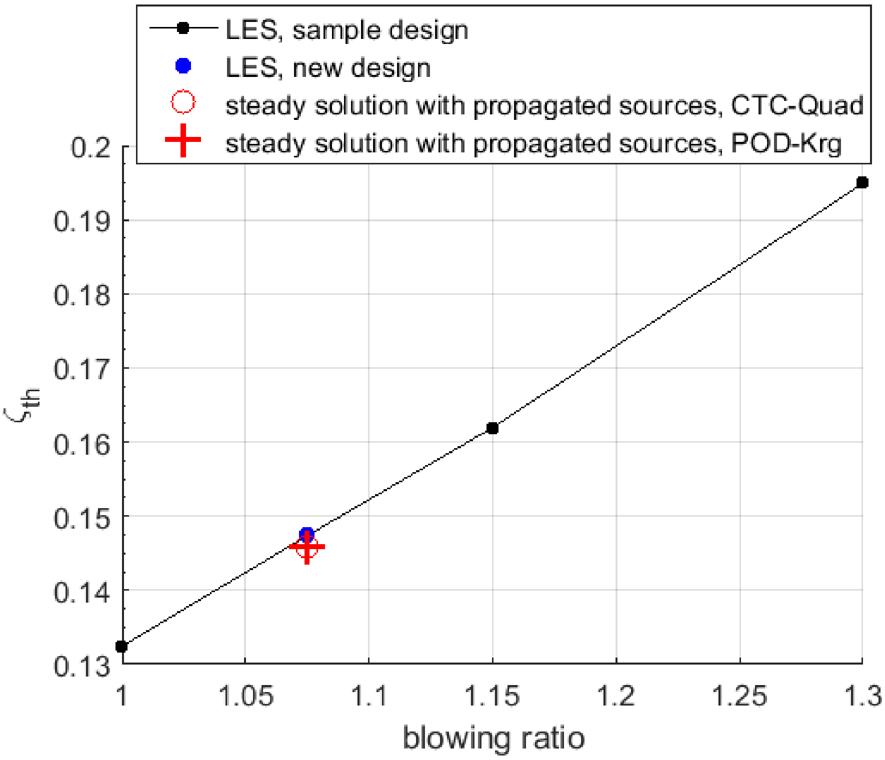


Figure 7-15 Variation ζ_{th} at different blowing ratios

The contours of the adiabatic effectiveness are compared in Figure 7-16. From the LES prediction, it can be seen that the adiabatic effectiveness increases as the blowing ratio is increased. The steady-state solutions at the new design (BR=1.075) with the propagated source terms fit well into the trend of variation predicted by the LES. Also, the contour distributions of η_{ad} are quite similar to the LES prediction in terms of both the magnitude and the shape.

The source terms are shown in Figure 7-17, for the energy source on the plane at the mid-height passing the mid-heights of the slot and chamber. The LES-extracted source terms on different blowing ratios show similar pattern, and a trend of variation can be seen. The propagated source terms fit into the trend.

Overall, the validation shows that the source term propagation is able to take into account of the variation of the blowing ratio and produce steady-state solution that match well with the LES at the design with the new blowing ratio.

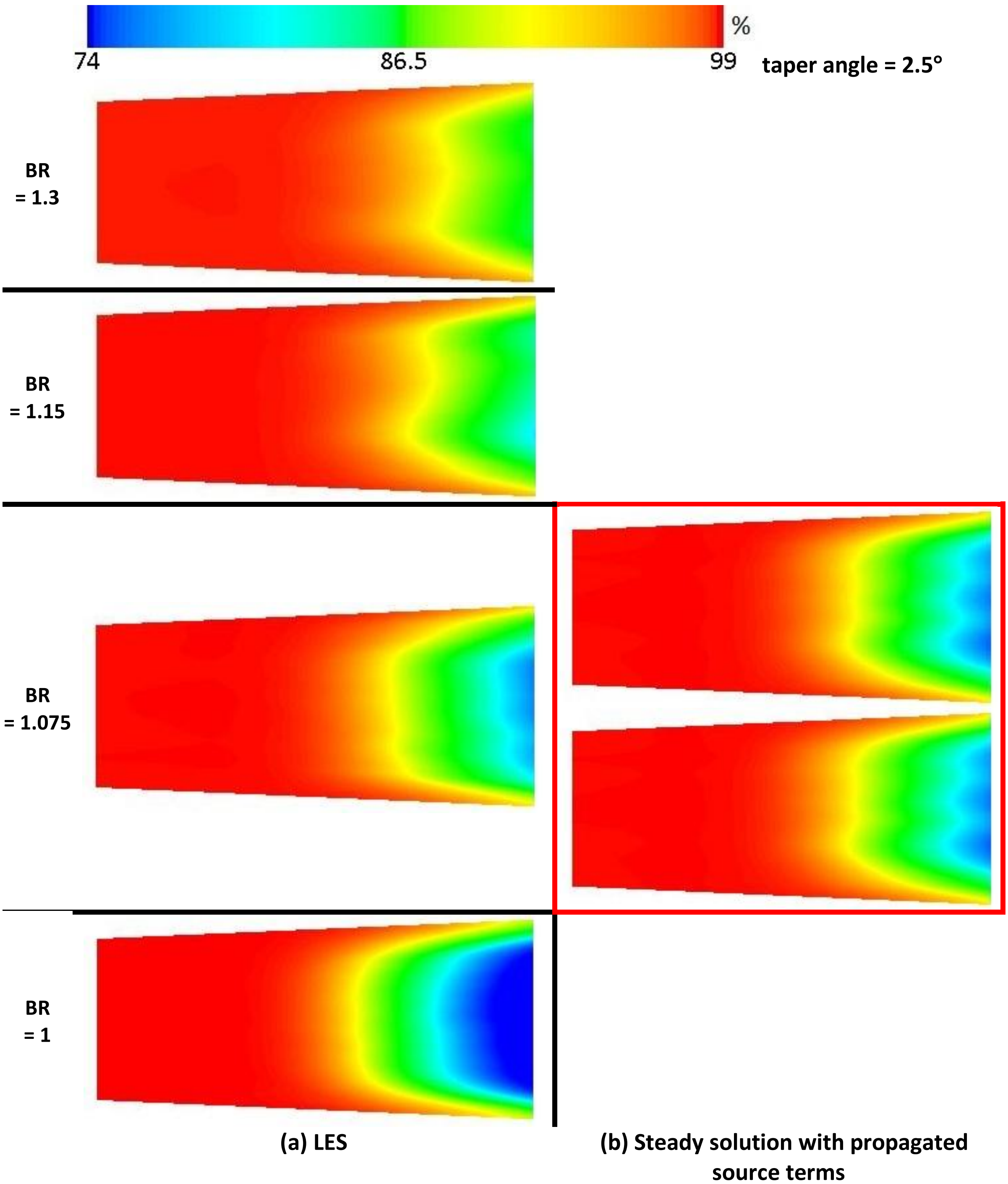


Figure 7-16 η_{ad} on breakout from steady solutions with various types of source terms

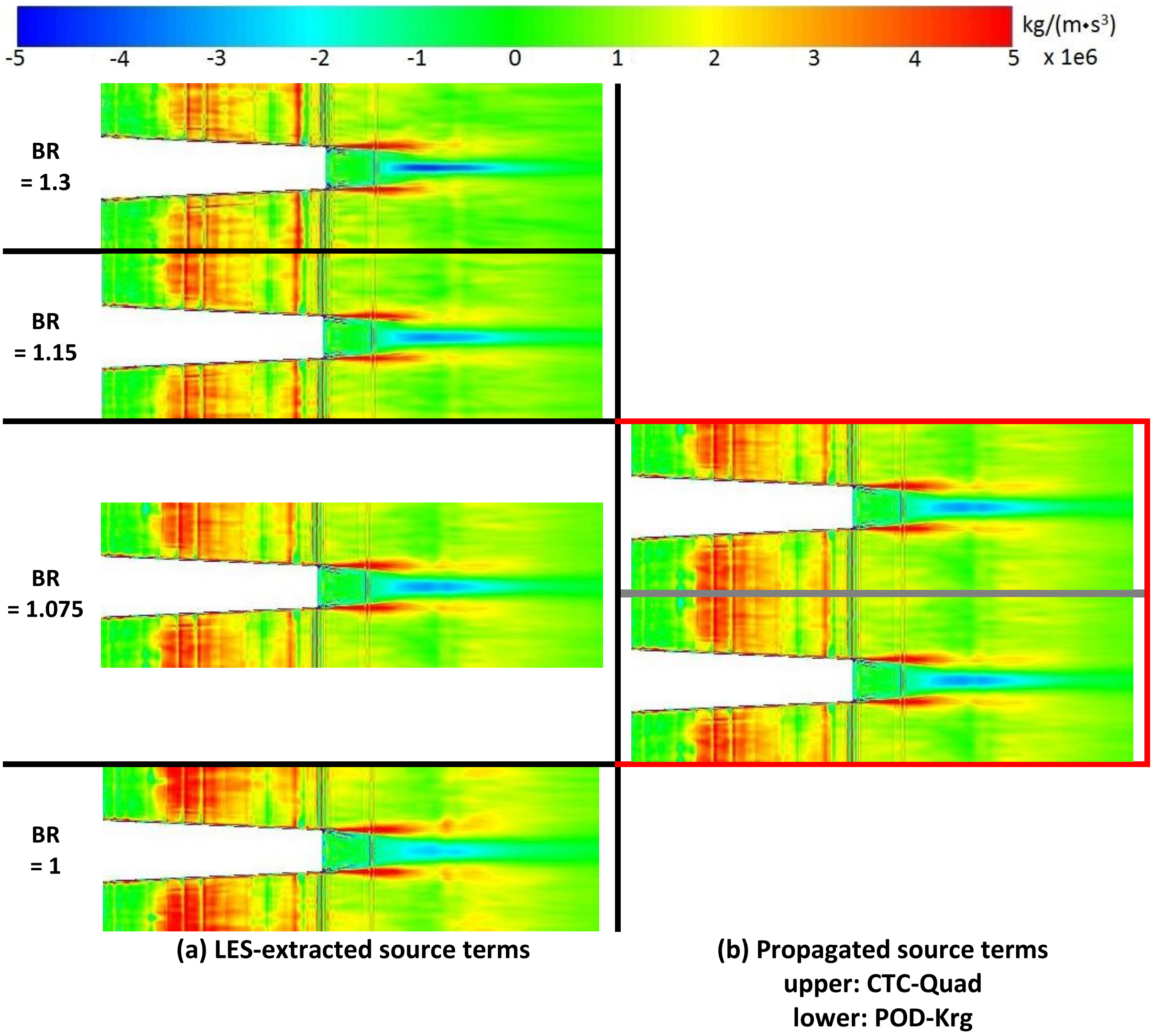


Figure 7-17 Propagated source terms compared with LES-extracted source terms

C. Variation of Both Taper Angles and Blowing Ratios

Finally, the validation is performed for the new designs with combined variation of the shape and the flow control in the two-dimensional design space. The LES is performed for the uniformly-distributed 9 samples from which the sample source terms are extracted and propagated. The taper angle ranges from 0° and 5°. The blowing ratio ranges from 1 to 1.3. The three new designs for the validation are located at [taper BR] = [1.25° 1.075; 2.5° 1.075; 3.75° 1.225], lying at the centre or the middle of the surrounding samples.

The main performances, which are the mean adiabatic effectiveness on the rear breakout surface η_{ad} and the thermal dynamic loss coefficient ζ_{th} are compared in Table

7-3 below (new designs in red frame). In addition, on the new designs, the relative error between the LES and the steady-state solution for each performance are shown in the parenthesis below each respective performance. The relative error are the difference of the performance between the LES and steady-state predictions, which is then normalized by the difference between the maximum and minimum values of the performance of the sample designs. It can be seen that, on all the new designs, the maximum of the relative errors for both performances is around 2.5%, which can be seen as small.

Table 7-3 Comparison of predicted performances (η_{ad}, ζ_{th}) between the LES and the steady-state solutions with propagated source terms

BR \ taper	0°	1.25°	2.5°	3.75°	5°
1.3	η_{ad}: 0.9566 ζ_{th}: 0.1961		0.9392 0.1949		0.9237 0.1940
1.225				LES: 0.9289 0.1773 CTC-Quad: 0.9293 (0.34%) 0.1774 (0.21%) POD-Krg: 0.9293 (0.35%) 0.1774 (0.20%)	
1.15	0.9520 0.1630		0.9318 0.1620		0.9107 0.1607
1.075		LES: 0.9292 0.1479 CTC-Quad: 0.9285 (0.70%) 0.1462 (2.67%) POD-Krg: 0.9285 (0.71%) 0.1462 (2.67%)	LES: 0.9168 0.1474 CTC-Quad: 0.9141 (2.55%) 0.1459 (2.37%) POD-Krg: 0.9141 (2.54%) 0.1459 (2.38%)		
1	0.9239 0.1336		0.8838 0.1325		0.8539 0.1329

Variations of the two performances in the above table are also plotted in Figure 7-18 and Figure 7-19 in the combined shape-FC design space below. For both performances the steady-state predictions with both source term propagation methods lies closely to the LES prediction for all the new designs. The trend of variation indicates that the mean η_{ad} is sensitive to both the land taper angle and the blowing ratio, while the thermal dynamic loss mainly varies with the blowing ratio.

The contour of η_{ad} on the rear breakout surface is shown in Figure 7-20, for the new design at [taper BR] = [1.25° 1.075] and the surrounding four samples. It shows that the magnitude of η_{ad} from the steady-state prediction is close to the LES, while the exact shape of the contour still shows some difference.

The present validation shows that for the combined variation of the shape and flow control in the present case, the source term propagation is able to take into account of the change of the shape and flow control simultaneously and produce accurate enough source terms at the new designs which make the steady-state solutions of the new design match well with their LES solutions.

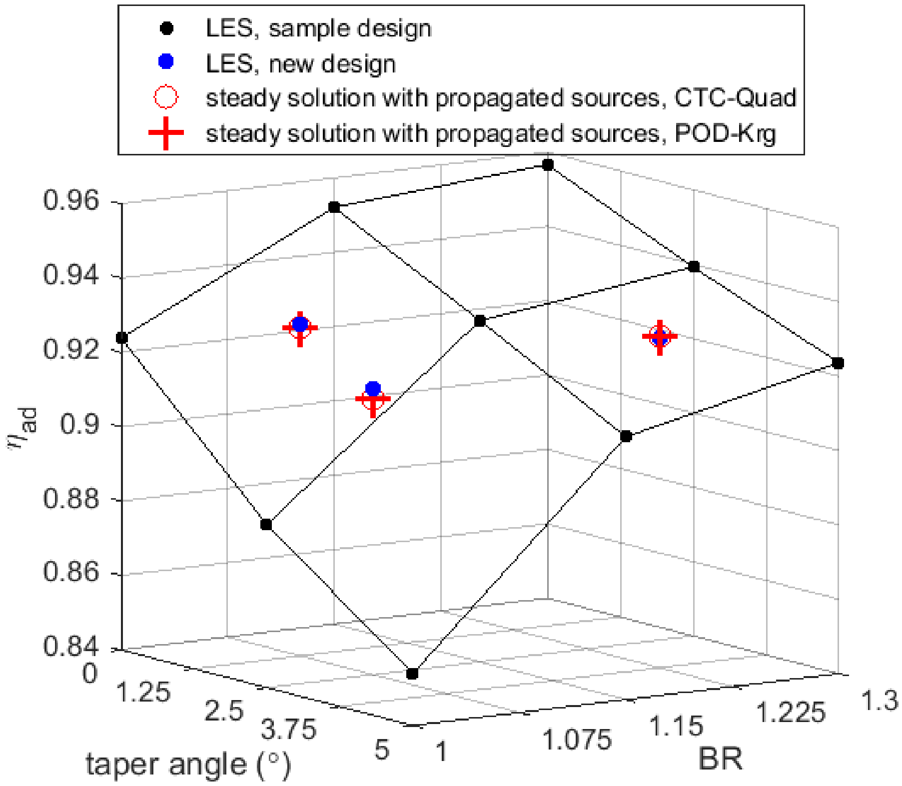


Figure 7-18 Comparison of the mean adiabatic effectiveness of the breakout surface, in the combined shape-FC design space

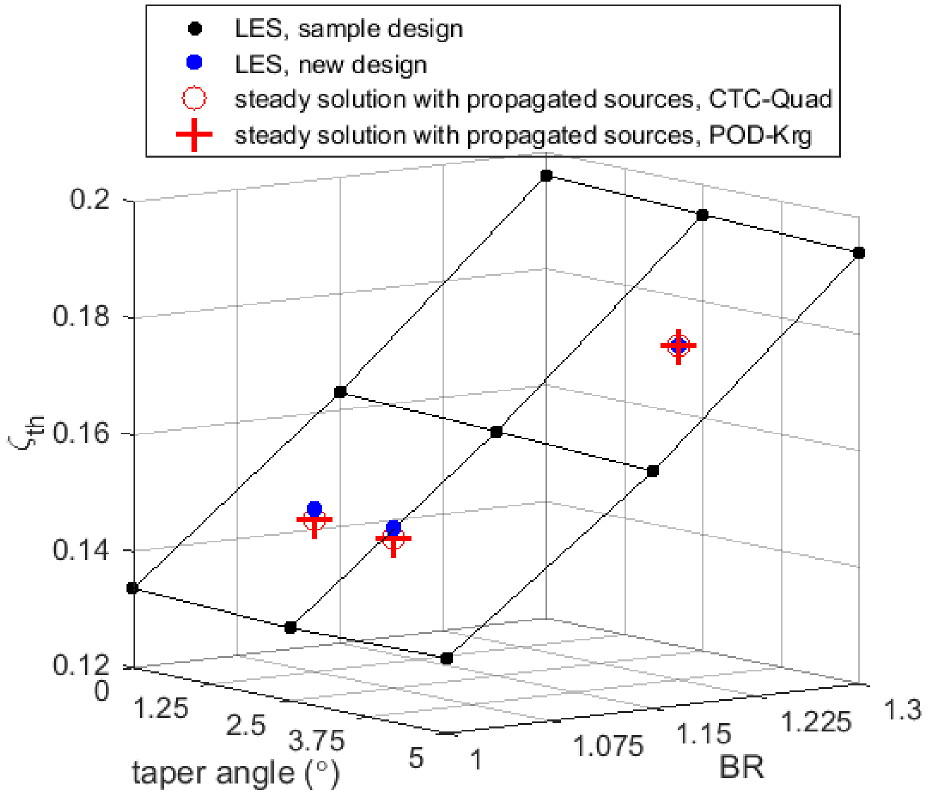


Figure 7-19 Comparison of the thermodynamic loss coefficient, in the combined shape-FC design space

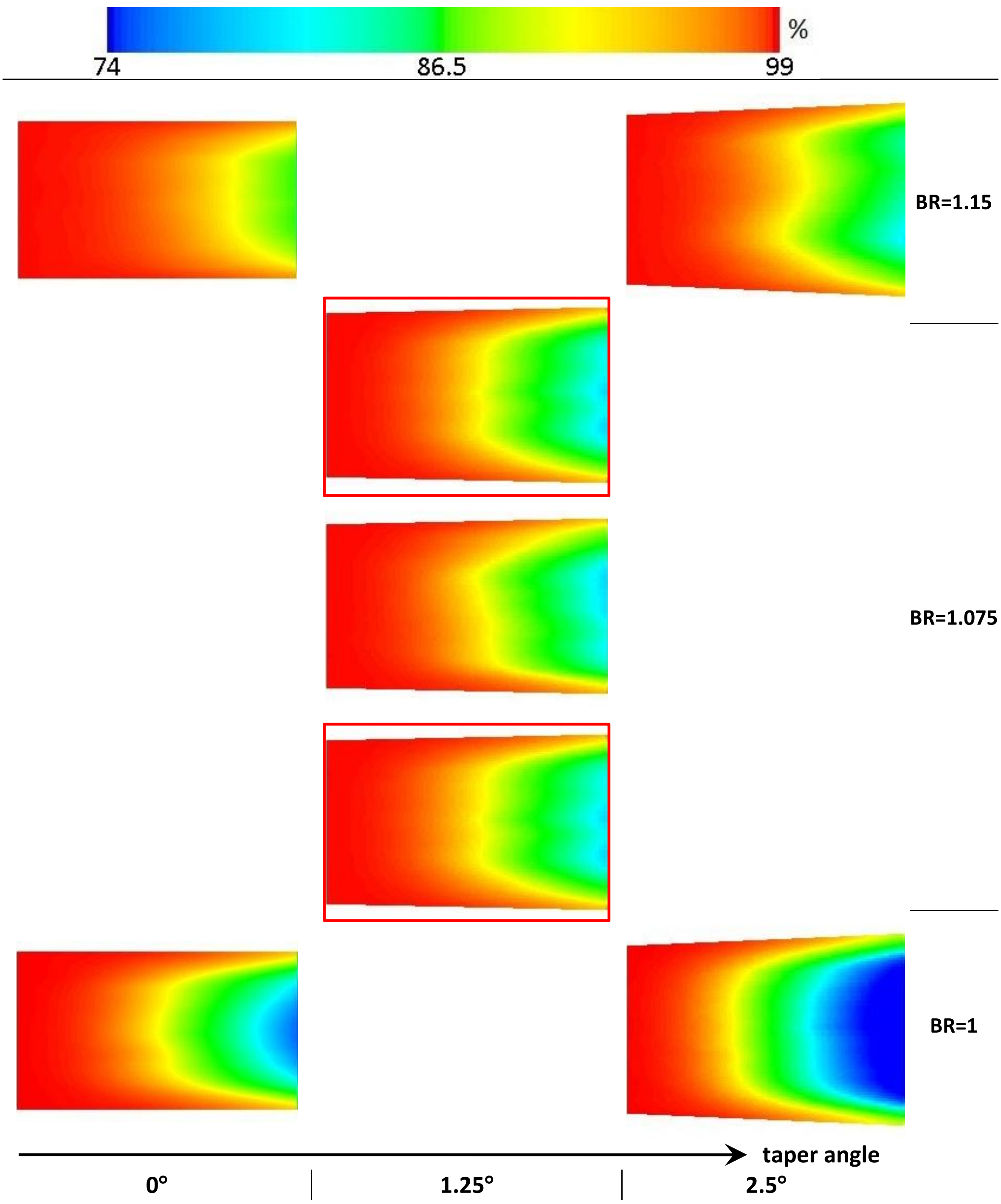


Figure 7-20 Contour of η_{ad} on the new design and the surrounding four samples in the 2D design space, using extracted sources (default) and propagated sources (red frame)

7.6 Summary

In this chapter, the Part II development of the efficient LES method for design optimization – the source term propagation is presented. The propagation can be carried out in two approaches: the cell-to-cell propagation approach and the POD propagation approach. The cell-to-cell approach is conceptually straight forward, but requires the construction of regression or interpolation functions for each of the large number of cells, which can be inefficient in some cases. The POD approach cast the source terms into a linear combination of only a few POD bases and the regression/interpolation is performed on only a few POD coefficients to construct the new source terms, which is computationally much more efficient, particularly for the kriging interpolation for example.

The validation studies are carried out for a blade trailing edge coolant injection case with two basic design variables (taper angle and blowing ratio) being included to represent the essential ingredients in combined shaping and flow control optimizations. The results of the validation studies show that the developed source term propagation methods is able to adapt the source terms properly based on the LES-extracted source terms from the samples, to the new designs that are different in both the shape and the flow control. The resulting steady-state solutions are shown to be close enough to the test LES solution at the new designs, and the predicted integral quantities and flow field distribution also fit well into the trend of variation predicted by the LES.

It is also shown that the mean adiabatic effectiveness on the breakout surface is sensitive to the taper angle but also more to the blowing ratio. The downstream thermal dynamic loss is not very sensitive to the land taper angle but sensitive to the blowing ratio.

At this point, the key ingredients of the efficient LES method have all been developed. The validations show that it can be a reliable method to provide efficient and high-fidelity performance evaluation for the optimizations that combines the shaping and flow control design. A demonstration of how to apply the efficient LES method in the combined optimizations of the shape and flow control is presented in the next chapter.

Chapter 8

Integrating the Efficient LES Method to the Optimization

As the efficient LES method has been developed in the previous chapters, this chapter presents how to integrate it into the optimization procedure. The case used for the demonstration combines the optimization of the shape and the flow control for a three-dimensional realistic configuration.

8.1 Case Settings

The configuration to be optimized is the trailing edge cutback injection, as already been investigated extensively in previous chapters. The detailed case setup (Chapter 5), numerical methods (Chapter 3) and definition of quantities, e.g. adiabatic effectiveness (Chapter 5), loss (Chapter 7), etc., have already been explained in those chapters hence will not be repeated here.

Two design variables are chosen, one for the shape and the other for the flow control. The shape design variable is the taper angle of the land as illustrated in Figure 5-14 in Chapter 5, ranging from 0° to 5°. The flow control design variable is the blowing ratio, which is set to vary between 0.7 and 1.3. For a demonstrative case as this one, selection of only two design variables shall be a balance between making the case realistic and making the analysis of the result convenient.

As the concurrent optimization approach has produced the highest performance gain in Chapter 4, it is used in this demonstration case, where the shape and flow control are optimized simultaneously. The source term propagation method is chosen to be the kriging interpolation in the POD propagation approach.

8.1.1 Objective Function

In the trailing edge cutback injection configuration, both the thermal and aerodynamic performances are important. On the thermodynamic side, the concern is to improve the cooling performance on the trailing edge, including both the breakout and the land surfaces, quantified measured by the adiabatic film effectiveness η_{ad}, as defined in Chapter 5. On the aerodynamic side, the aerodynamic mixing loss can seriously affect the overall performance. The injection may reduce the aerodynamic loss by filling the wake, but may also increase the loss due to the turbulence in the chamber and cutback region. Also, the energy expenditure is required to raise the total pressure of the injection. To account for both aspects, the loss is quantified by the 'thermodynamic loss coefficient' ζ_{th} as shown by Osnaghi et al. (1997), and adapted in Chapter 7 for the incompressible flow of this case.

Accordingly, the optimization objectives are to maximize the mean adiabatic effectiveness $\eta_{ad,m}$ on the rear 2/3 portion of the breakout and land surfaces (Figure 8-1) and minimize the thermodynamic loss ζ_{th}. Thus the objective function to be maximized is a weighted sum of these performances, defined as:

$$I = w_1\eta_{ad,m,brk,norm} + w_2\eta_{ad,m,land,norm} + w_3\left(1-\zeta_{th,norm}\right) \tag{8.1}$$

Normalization is required as the order of magnitude of different types of performances are quite different. Subscript "norm" denotes the normalization using the lower and upper bound values in the initial samples. For example, the normalized adiabatic effectiveness for the breakout surface is:

$$\eta_{ad,m,brk,norm} = \frac{\eta_{ad,m,brk} - \eta_{ad,m,brk,low}}{\eta_{ad,m,brk,up} - \eta_{ad,m,brk,low}} \tag{8.2}$$

The weighting coefficients are chosen as $w_1 = 0.45$, $w_2 = 0.15$ and $w_3 = 0.4$, approximately based on the relative area between the breakout and the land. As two η_{ad} are considered, the sum of their weights is set slightly bigger than the weight for the loss.

No constraints are applied in the present demonstrating case.

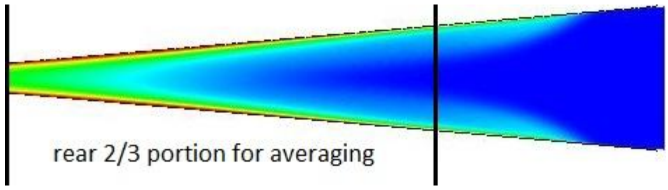


Figure 8-1 Portion over which η_{ad} on the land surface is averaged (y-axis view)

8.1.2 Integration into the Optimization Procedure

To integrate the efficient LES method into the optimization, the optimization procedure described in Chapter 3 is modified. The resulting procedure is shown in Figure 8-2 below, based on the POD-Krg source term propagation approach. For the present demonstrative case, some unimportant aspects are simplified but can still be made more sophisticated when required, such as the uniform sampling in the DOE, the omission of the constraint functions, the omission of the statistics lower bound (SLB) method in updating the sample pool. However the following description of the procedure is still based on the general case. The actual procedure with any simplification or sophistication should still be compatible with the general description.

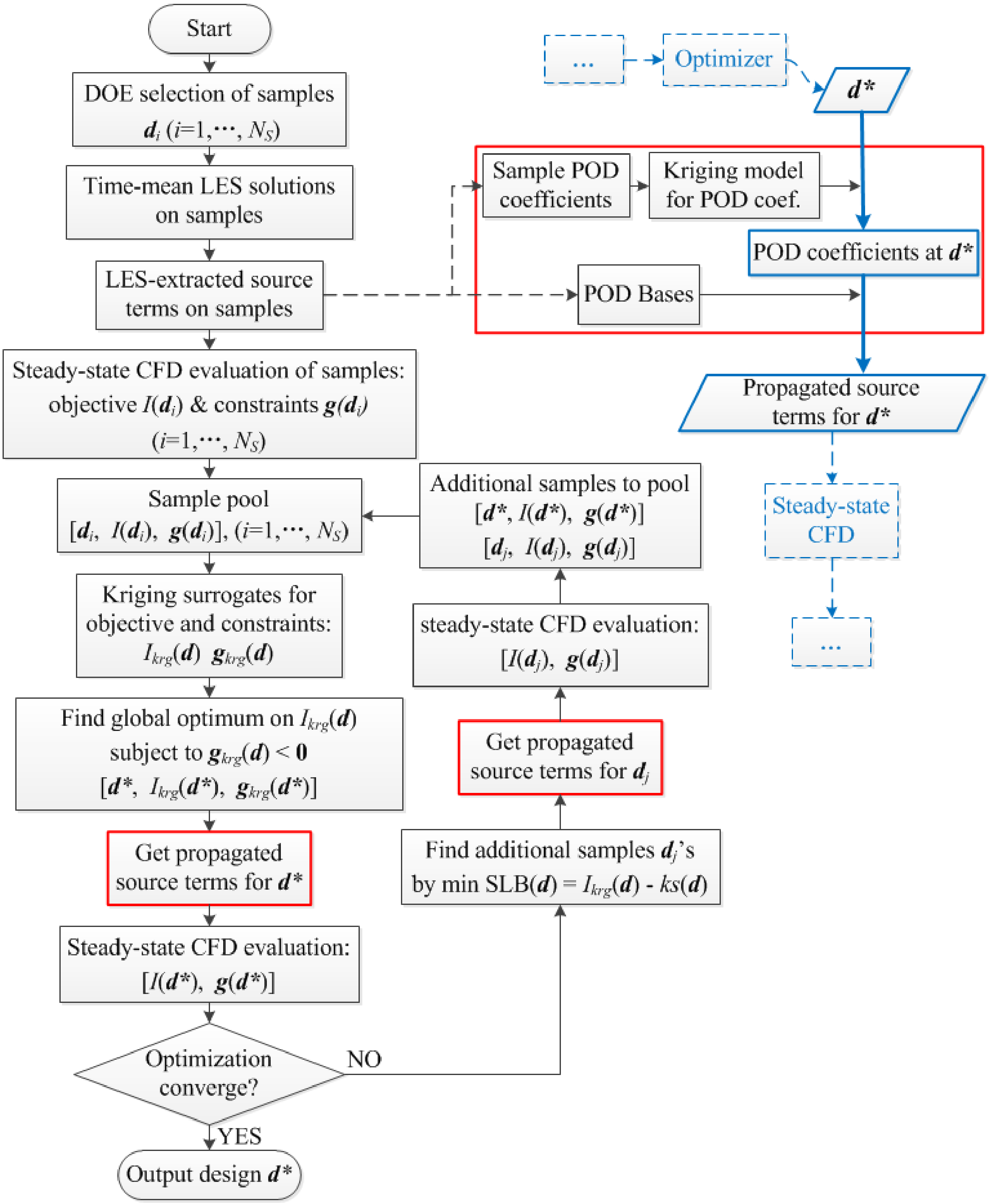


Figure 8-2 Implementation of the efficient LES method in the optimization procedure

In the above procedure, main modification to the original one of Chapter 3 is the CFD evaluation part, particularly with the added red-frame module for the source term calculation. First, after the DOE sample selection, LES are performed on these samples to compute and the source terms that will make the steady-state solution to reproduce the time-mean LES solution for each sample are computed using the method presented in Chapter 6.

Then, the sample source terms are used in two strands as follows.

On the side line, as indicated by the dash line to the red-frame module on the upper right hand side, the sample source terms are cast into the POD bases, and the coefficient for each POD basis at the all the samples are computed. One Kriging interpolant is then built respectively for each POD coefficient. The POD coefficient Kriging interpolant and the POD bases are then ready for use whenever source terms are requested by a new design. For example, in the subsequent optimization procedure, when the a new design represented by a set of design variables $\boldsymbol{d}$ is fed into the red-frame module, the POD coefficients are computed by with the POD Kriging interpolant at $\boldsymbol{d}$. The source terms for the new design $\boldsymbol{d}$ are then computed by the summation of the POD bases weighted by the interpolated POD coefficients.

On the mainline, the samples source terms are used by the steady-state simulations to compute the objective function and constraint functions of the samples. The Kriging models for the objective and constraint functions are then built to interpolate the functions across the design space. The global optimum on the surrogate objective function is then found. For the present design space of only two dimensions, it can be found by simply creating a uniform and fine grid say 201×201 points in the 2D design space, and evaluating the surrogate objective function at he grid points and then chose the maximum. The corresponding optimal design $\boldsymbol{d}^*$ is fed into the red-frame module to compute the propagated source terms for design $\boldsymbol{d}^*$. The steady-state solution with the propagated source terms are then obtained for the true objective function and constraints at $\boldsymbol{d}^*$. The values of the surrogate and true objective functions are compared, and combined with the other criteria (as described in Chapter 3) to check the convergence of the optimization. If convergence is not achieved, additional design can be chosen, and the source terms for these designs are computed in the red-frame module for the steady-state CFD evaluation of the additional designs. After that, the CFD evaluation of the current optimal design and the additional designs are added to the sample pool to update the Kriging models for the objective and constraint functions for the next optimization cycle.

It can be seen that during the entire optimization process, LES is only needed for the initial DOE samples at the beginning, while the remaining CFD evaluations are steady-state simulations.

8.2 Results and Discussions

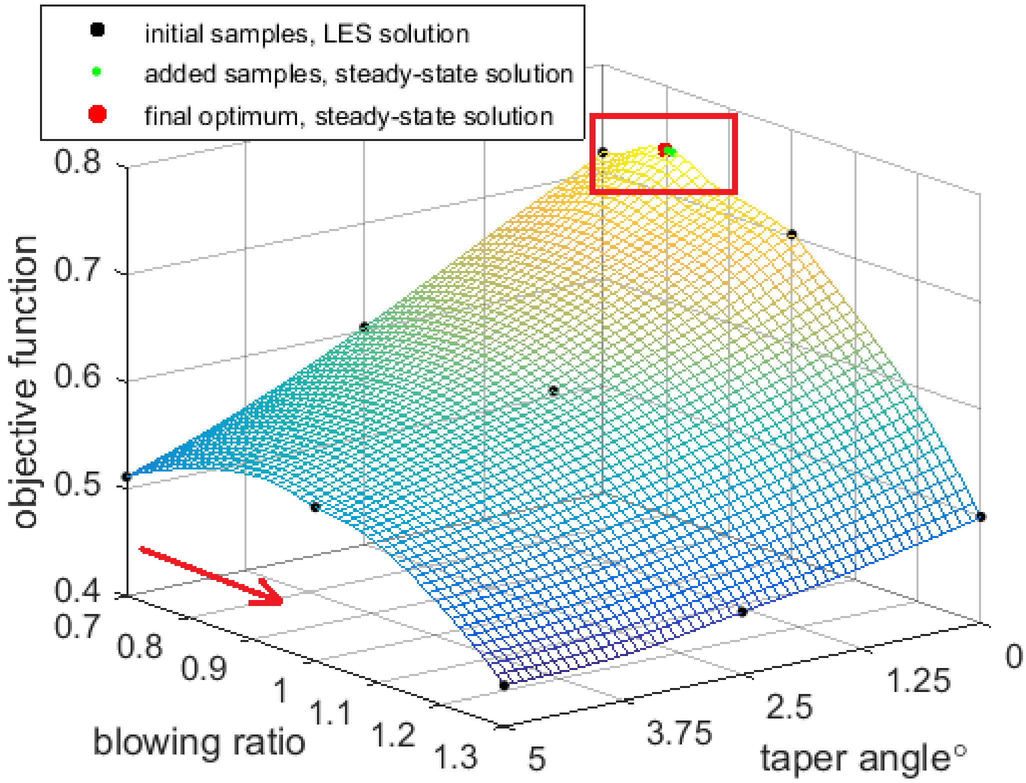
8.2.1 Optimal Design

The parameters of the global optimum are presented in Table 8-1 below.

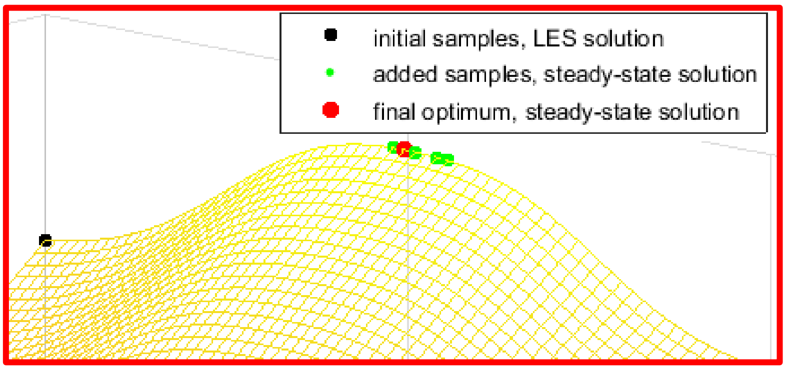
Table 8-1 Globally optimal design of the concurrent optimization of shape and flow control

Parameter	Value
Taper angle	0°
Blowing ratio	0.799
Objective function	0.7408
$\eta_{ad,brk}$	0.8669
$\eta_{ad,land}$	0.1484
ζ_{th}	0.1089

The global optimum and the Kriging model of the objective function from the final optimization cycle in the design space are shown in Figure 8-3 below, together with the initial samples, and the added sample designs - the global optima from the intermediate optimization cycles in this case. The added samples and the final optimum together show that the concurrent optimization convergences shortly after a few optimization cycles. It shows that for all the blowing ratios, as the taper angle increases, the objective function decreases monotonically. On the other hand, the objective function first increases and then decreases as the blowing ratio is increased, for all the taper angles. Thus, the optimal land shape is the straight land, and optimal blowing ratio lies somewhere in the middle of the flow control dimension.



(a) global view



(b) close-up view at the vincity of the global ptimum

Figure 8-3 Locations of samples, the final optimum and the final surrogate objective function for the objective function in the design space

Variations of the performances that make up the objective function are further analysed respectively. Figure 8-4 below shows the variation of the mean adiabatic effectiveness of the breakout surface in the design space. A Kriging surface is built through the initial samples to visualize the trend of variation. The performance of the optima of the intermediate optimization cycles (i.e. the added samples) and the final optimum evaluated with the steady-state simulations all fall closely onto this LES-predicted trend. At all the blowing ratio, the effectiveness of the breakout surface decreases as the taper angle is increased, probably due to the widened exposed area that dilutes the coolant. At all the taper angle, the effectiveness increases as the blowing ratio increases, which is expected as more coolant injection provides stronger shielding against the hot main flow.

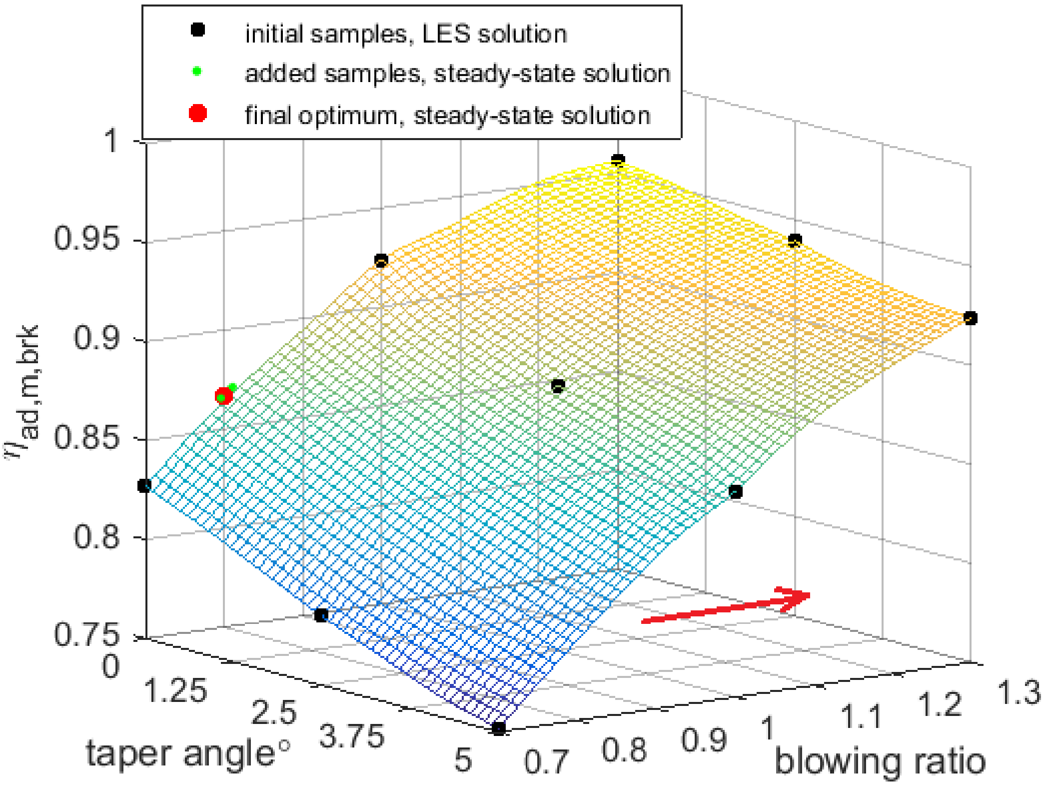


Figure 8-4 Average adiabatic effectiveness on rear 2/3 portion of the breakout surface, trend of variation, values of the samples and the final optimum

In Figure 8-5 below, variation of the mean adiabatic effectiveness on the land surface in the design space is shown. A Kriging surface is built through the initial samples to indicate the trend of variation. The added samples and the final optimum evaluated with steady-state simulations all fall closely onto this trend. The land effectiveness generally decreases as the blowing ratio is increased, which is opposite to the trend of the effectiveness on the breakout surface. At higher blowing ratios, the land effectiveness varies only slightly as the taper angle is increased. But at lower blowing ratios, the land effectiveness increases with the reduction of the taper angle, which is also opposite to the trend of the breakout effectiveness.

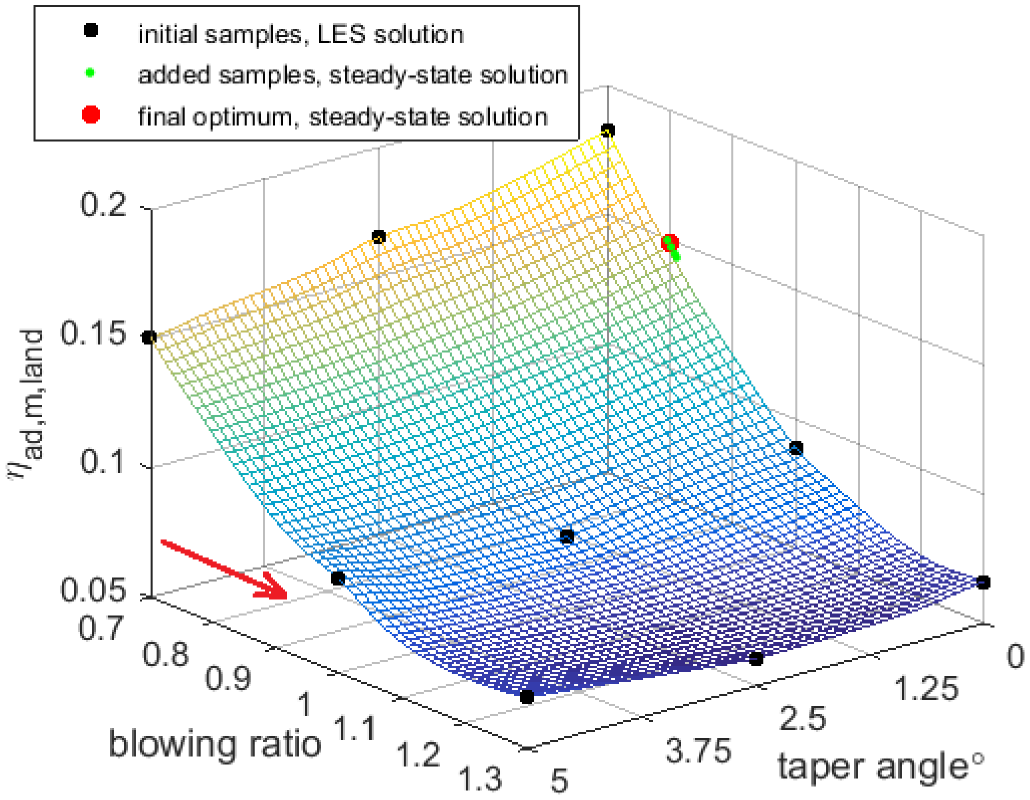


Figure 8-5 Average adiabatic effectiveness on rear 2/3 portion of the land surface, trend of variation, values of the samples and the final optimum

The increase of the land effectiveness at lower blowing ratios and lower taper angles may be attributed to the roll up of the coolant onto the land surface. A lower coolant injection speed results in a lower coolant velocity component normal to the land surface, thus the rolled-up coolant will be closer to the land surface to enhance the cooling. This is shown by the x-vorticity ω_x at different sample blowing ratios in Figure 8-6 below, at the cutplane at 2/3 breakout length downstream of the slot. It shows that the rolled-up x-vortices are closer to the land at $BR = 0.7$ than at $BR = 1.3$ for both taper angles. The corresponding effectiveness on the land is also compared in Figure 8-7. The effect of the taper angle on the effectiveness can also be seen in Figure 8-6. At $BR = 0.7$, both lands have coolant roll up, but the vortices indicated by x-vorticity contours at the straight land appear closer to the surface. At $BR = 1.3$, there is coolant roll-up at the straight land case, but on the tapered land the x-vorticity ω_x above the land is almost zero, which is also observed by Ling et al. (2015) at this blowing ratio. However, the land effectiveness of the two taper angles at this blowing ratio are almost the same, perhaps because the coolant roll up on the straight land is not close enough to the land, as also mentioned in the above experiment.

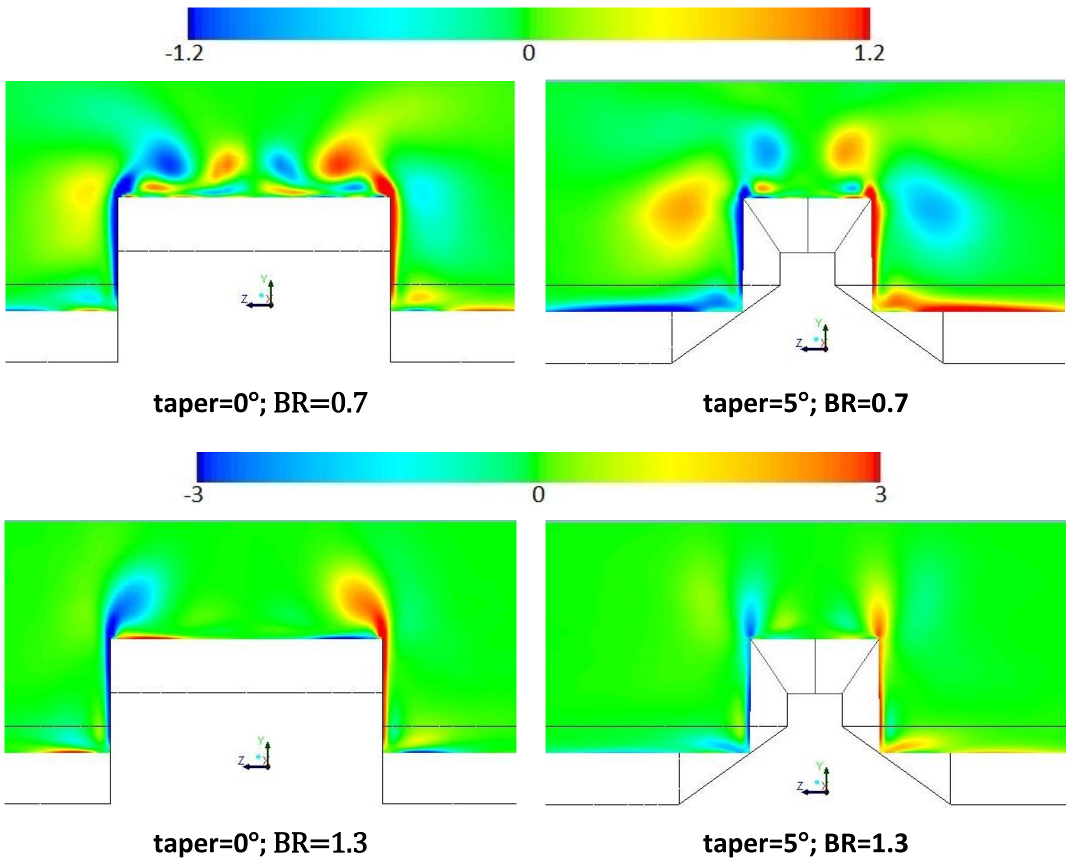


Figure 8-6 Streamwise vorticity ω_x normalized by slot height and mainstream velocity at slot, at 2/3 breakout length downstream of the slot

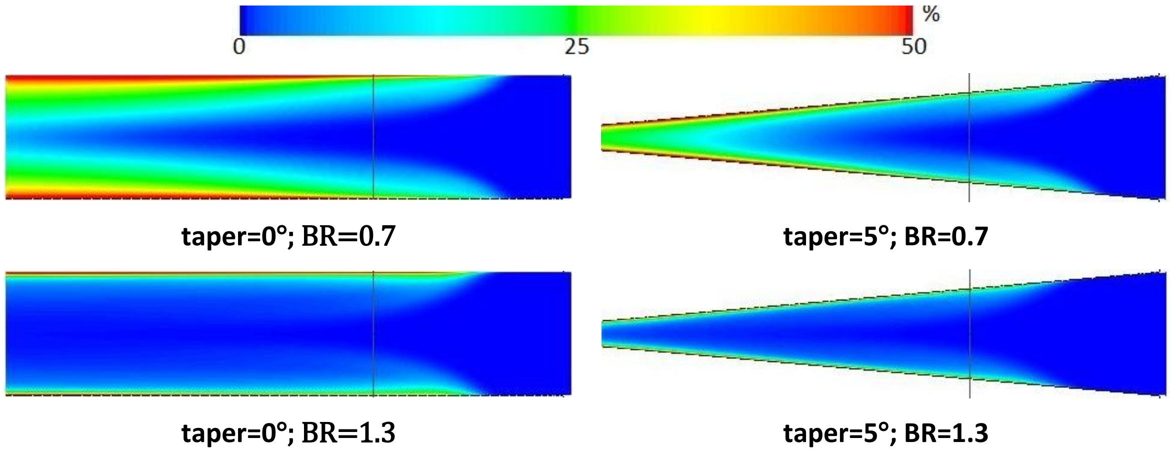


Figure 8-7 Adiabatic effectiveness on the land

The trend of variation in the thermodynamic loss is shown in Figure 8-8 below by the Kriging surface going through the LES-evaluated initial samples. The added samples and the final optimum evaluated by the steady-state solution all fit well onto the trend. The loss is almost insensitive to the taper angle, but sensitive to the blowing ratio, and increases as the blowing ratio is increased, which may well be due to the increased energy cost of the injection.

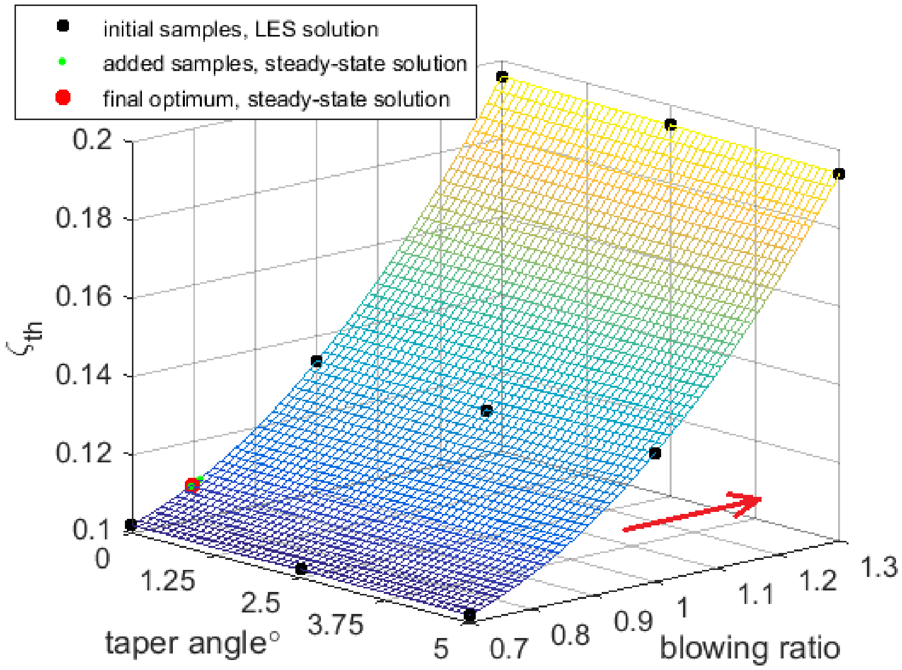


Figure 8-8 Thermodynamic loss coefficient at the outlet, trend of variation and values of the samples and the final optimum

8.2.2 Reduction of Computational Cost

The computational costs of the current demonstrative case is analysed, breaking down the optimization procedure into several stages, i.e. pre-LES RANS, LES, source term extraction, source term propagation and are listed in Table 8-2 below. The pre-LES RANS provides the initial solution for the LES. The total number of cells in the computational mesh is 3,585,800. The LES is run in parallel with 60 physical processors at 2.4 GHz on a computing cluster. All the steady-state simulations are performed in parallel with 10 physical processors at 2.4 GHz in a desktop workstation.

Table 8-2 Computational cost of each computational stage

Computation stage	Number of iteration/time-step	Number of processors	Wall clock time	CPU time
Pre-LES RANS	3000	10	7 h	70 h
LES time-integration	$1.4\times10^5\ \Delta t$	60	172 h (7.2 days)	10320 h
Source term extraction	2000	10	6.1 h	61 h
Source term propagation	---	1	< 1 min	< 1 min
Steady-state simulation	300	10	2 h	20 h

The above table shows cost of the LES is three orders of magnitude higher than the steady-state simulation and the source term extraction. Whereas the computational cost for the source term propagation is negligible.

Comparison of the computational cost between the current case with the efficient LES and the brute force LES (in which all CFD evaluations are done with LES) is summarized in Table 8-3. In the present case, 9 initial samples and 5 additional designs are evaluated, amounting to 14 CFD evaluations in total. For brute-force LES evaluation, 14 LES is required. This amounts to 145,460 CPU hours. While if the efficient LES method is used, 9 LES are performed, each of which requires a pre-LES RANS and a source term extraction. After that, 5 steady-state simulations are performed. This in total amounts to 94,159 CPU hours. Therefore, for the present optimization case, the efficient LES method shows a reduction of the total computation cost by 51,301 CPU hours, which is about 5 weeks if shared by 60 physical processors on the computing cluster.

Table 8-3 Reduction of CPU time by the efficient LES method for the present case

Computational stage	Brute-force LES (estimated)	Efficient LES method	Reduction of time
Pre-LES RANS	70×14 h	70×9 h	
LES	10320×14 h	10320×9 h	
Source term extraction	---	61 h	
Steady-state simulation	20×14 h	20×9 h	
Overall	145,460 h	94,159 h	51,301 h (i.e. over 1 month with 60 processors)

8.2.3 Some Comments on the Computational Cost and Solution Accuracy

In the above demonstration several aspects are simplified. In practice the use of more sophisticated methods can possibly lead to more savings of the computational cost. Firstly, more efficient DOE method can be used to reduce the number of initial samples, at which the LES is performed. Secondly, LES could possibly be performed on only some of the DOE samples. Thirdly, many more additional designs apart from the initial DOE sample could be required to update the Kriging model for the objective and constraint functions in optimization, where the efficient LES method can be used to evaluate these additional designs. With all these factors considered, the fraction of the LES runs in the total number of CFD runs can be much smaller. In this sense, a higher saving of the computational cost is possible.

However, as both the propagated source terms and the initial surrogate objective and constraint functions are built from the same set of LES-evaluated initial samples, two possible situations need to be considered. On one hand, if the number of the initial samples evaluated by the LES is too large, the propagated source terms derived from these LES solutions and hence the efficient LES indeed would be accurate for evaluating the new designs. However, the large number of LES samples could also possibly make the initial surrogate objective and constraint functions accurate enough. This would probably means no further update of the surrogate models using the steady-state efficient LES solutions are needed, effectively leading to the brute-force LES evaluations for the optimization. On the other hand, if the number of initial LES-evaluated samples is too low, then the surrogate model for the objective and constraint function based on the LES-evaluated samples will probably not be accurate enough and thus needs to be updated. However, the small number of initial LES-evaluate samples could also possibly increase the error of the propagated source terms, making the efficient LES less accurate for the update of the surrogate models, thus making it more difficult to increase the accuracy of the surrogate models.

Therefore, a balanced situation needs to be found between the above two situations so that the efficient LES can be more effective. One such situation could be that a small number of LES evaluated samples are sufficient to make the propagated source terms accurate, but insufficient for the initial surrogate models. Thus, the efficient LES method can be used in place of the initial LES solutions to improve the accuracy of the surrogate models, leading to an effective reduction of computational cost. How well such a situation can be found so that how effective the efficient LES can be could well be case dependent and may need more study on different cases. But present results still show its worth future investigation.

Another benefit of using the efficient LES method is that despite the possible error in the propagated source terms, the flow field still needs to be solved. Thus the flow field will satisfy the boundary condition of the new designs, and also provides the flow detail without resorting to more time-consuming LES.

8.3 Summary

The present optimization case demonstrates the integration of the efficient LES method into an optimization case to concurrently optimize the geometry and flow control of a 3D realistic configuration. Instead of simply using the source terms to reproduce the known time-mean LES solutions, this work has shown the source terms can be propagated for the steady-state evaluation of the new designs without further LES for the optimization. This represents a step forward from previous research. The reduction of the computation cost also shows to be significant for the demonstrative case. Nevertheless, how effective the efficient LES method can be for the optimization to reduce the computational cost while preserving the accuracy of the flow solution still need more exploration on different cases. But the method still shows to be promising and is worthy for future study.

Chapter 9

Conclusions and Recommendations for Future Works

9.1 Conclusions

Shaping and flow control are two extensively studied disciplines in improving the various performances of aerodynamic surfaces. Research over the past decades has indicated that these two disciplines are very effective in improving the performance. However, there have been only a few papers that have considered the two disciplines at the same time. These few works have implied the potential for higher performance when combining the two. However, all of them stem from the purposes to solve the particular flow control problems. None of them have been spawned from or followed up to formally addressing the approach of combining shape optimization and flow control in a general sense. Moreover, they were either based on very low-fidelity flow predictions and simplified 2D geometries. Thus the results so far are scarce and inconclusive.

Two principal research questions are thus raised in this study: (I) is there any positive benefit of combining the shaping and flow control, and how could they be better combined? and (II) if so, what would be a required flow modelling fidelity, and what can we do to advance the modelling techniques towards providing such a modelling capability to evaluate the performance in the combined shaping and flow control optimization accurately and efficiently? The two objectives of the present work thus seek answers to these two questions.

9.1.1 Objective I

To address the first question, all the five possible optimization approaches are identified and systematically studied. The first two approaches represent conventional wisdoms of shaping and flow control confined within their own disciplines respectively. The third and fourth approaches represent the combination of the disciplines in a sequential manner. The fifth approach represents the combination in a concurrent manner. To compare the potentially best result from each approach, they need to be carried out consistently. The global optimum in their respective design space also needs to be found. An optimization system is developed to perform the optimization consistently for all the approaches, with the kriging surrogate model to facilitate the search for the global optimum.

The combination of the two disciplines is firstly investigated on 2D configurations with RANS flow modelling to quickly indicate the worthiness of future substantial investment on more realistic 3D cases with high-fidelity flow control modelling. Both internal flow and external flow are covered. In the internal flow case, a compressor blade with blowing and suction applied on the suction side is optimized. The results show that when optimizing the blade shape without applying the flow control, either the loss at the design inlet flow angle can be reduced or the operating range can be increased. But the two performance parameters cannot be improved at the same time. However when combining the shaping and flow control optimization, both performance parameters can be improved simultaneously. In the external flow case, the lift of a circulation control airfoil is maximized. The trailing edge shape and the injection location are optimized. Five optimization approaches are systematically studied. The results show that the combined optimization produces higher lift than a shaping alone or a flow control alone optimization, and the concurrent optimization produces higher lift than the two sequentially combined approaches. It also shows that in the combined shape-FC design space, the sequentially combined approaches can get stuck when repeated and failed to reach the globally optimal design which nevertheless can be found by the concurrent optimization.

Thus, both cases show further improvement of the performance when shaping and flow control optimization are combined. In addition, all the optimization approaches are systematically and consistently compared, so the present results are the first of the kind. The present study also suggests that if resource permitted, increasing concurrency in the interaction between the two disciplines is desirable for finding designs with higher performance.

9.1.2 Objective II

The inclusion of flow control in performance optimization has turned the attention to the required flow modelling fidelity in contrast to those commonly used in shaping only optimizations. To address the second research question, namely the modelling fidelity, past research and the present baseline LES validation show that for 3D realistic configurations, the complicated controlled flow field cannot be adequately predicted by RANS, but can be accurately predicted by the LES. Therefore, LES is considered necessary for the flow control prediction.

Thus, when combing the flow control optimization with shape optimization, conflicts arises between the large computational cost of LES and the required speed for a large number of performance evaluations. As the time-mean performance is of interest in the situation concerned, an efficient LES method is developed to use the fast steady-state simulation to predict the time-mean LES solution. The key is to model the turbulence correlation terms accurately, and put them as the source terms in the steady-state solver. These terms are derived from a few calibration LES solutions on the sample designs, and propagated to the remaining designs. So that except for a few LES solutions at the beginning, all the designs are evaluated with fast steady-state solutions while still achieving the prediction accuracy of the LES.

In the Part I development, the procedure to extract the source terms from the time-mean LES solutions is developed. The source terms can be conveniently computed from the steady-state residual evaluated at the time-mean LES solution, instead of directly computed from the instantaneous LES solutions. When explicit-Euler pseudo-time marching scheme is not available, the source terms should be computed iteratively. Also, some additional damping needs to be incorporated into the steady-state solver to get the steady-state solution converge. The source terms together with the additional damping terms accounts for the total unsteady effect on the time-mean flow. The results show that the steady-state solution with the extracted source terms successfully reproduces the LES solution on the LES-evaluated design itself. Validations are carried out in test cases for incompressible and compressible flow and for pure aerodynamic and aero-thermal flow problems.

In the Part II development, the extracted source terms are adapted and propagated from the LES-evaluated sample designs to the new designs. The assumptions that the source terms can be adapted and propagated are based on the similarity of the designs and their flow fields in the same design space. As a prerequisite, the geometry needs to be topologically similar. The source terms are shown to be as influential as the geometry and boundary conditions on the steady-state solutions, indicating the importance to develop the proper propagation methods.

The source term propagation can be carried out in the cell-to-cell (CTC) approach and the POD approach. In the CTC approach, the source terms on the new design are computed for each mesh cell one by one based on the values in the same-index cells of the sample designs. The interpolation or regression function needs to be built for each cell respectively. This can be inefficient for the large number of cells as required for the LES simulations. On the other hand, the POD propagation approach casts the source terms into a linear combination of POD bases and only needs to interpolate/approximate a few POD coefficients. Thus it is very efficient. For the case tested with Kriging interpolation for source terms, the POD method is shown to be 2 to 3 orders or magnitude faster than the CTC propagation approach. The source term propagation methods developed and carried out in both approaches are shown to be able to account for the difference in the shape and flow control properly, and get the steady-state solution on the new design fit well into the trend of variation set by the LES solutions of the sample designs.

Finally, the procedure to integrate the efficient LES method into the optimization is developed. The shape and flow control on 3D realistic cutback trailing edge blowing configuration is optimized. It demonstrates the source term can be propagated for the optimization, rather than simply reproducing the knwon LES solutions, which is a step forward from the previous research. Also, the efficient LES showed to reduce the overall computational cost for the demonstration case by more than one month if shared on 60 physical processors. For more practical implementation with more sophisticated optimization methods, further savings of computational cost is possible. However, how effective the efficient LES method is able to reduce computational cost while also preserving solution accuracy at the same time, especially comparing to the brute-force LES with surrogate models, can be case dependent. In sum, the efficient LES method is promising and is worth more exploration in the future.

9.2 Recommendations for Future Works

To the author's knowledge, the current study is the first piece of work in the open literature to recognize the long isolation of efforts in the shape optimization and flow control, and investigate the benefits of the combination of the two disciplines in a general sense supported by systematic and computational studies. The results are quite positive showing further performance improvement by the combination, and indicate that the concurrent optimization is desirable for higher performance.

This implies that future shape optimization and flow control designs should may pay more attention to the mutual interaction, making them mutually accommodating, particular for designs that needs flow control. For instance, in the design of the blade at off-design condition, airfoil/flap design at take-off landing condition, or the film cooling for turbine blade. The combination of the disciplines may produce significantly different and better designs than what has been put forward today by shaping only or flow control design alone.

The efficient LES method is the first attempt to provide efficient LES-based flow solution in the design optimization. Similar research for efficient unsteady flow analysis was all developed for URANS only. The performance of this method in this work shows basic success in resolving the conflict between high-fidelity and the speed in the flow modelling, and should make the combined numerical optimization of flow control and shaping in the future more affordable. A few suggestions are offered for future refinement of the methods.

First, it is worthwhile to investigate how the method would work with the shock-wave in high speed compressible flows. This is important in many practical applications, such as in the combined shape and cooling injection design of turbine blades in a realistic operating environment.

Second, one can seek to include the inlet turbulence modelling or hybrid-RANS (e.g. DES) issues into the method. The accuracy of the few calibration LES is the bases for the accuracy of the extracted and propagated source terms to produce the time-mean unsteady flow. Also, DES modelling helps to reduce the cost of the LES in high Reynolds number conditions. If the issues in the DES, such as the boundary between the RANS zone and

LES zone can be incorporated in the efficient LES method, the accuracy of the method can be further improved and the range of application widened.

Thirdly, further studies can be carried out in relation to the additional damping required to get the steady-state solution to converge. In the present study, the source terms are shown to provide a good amount of damping but still not enough. So it can be worthy to investigate the numerical mechanism of the instability of the steady-state solver at the time-mean LES solution with and without the additional damping, and see in what situation the additional damping will not be needed, or can be replaced by other measures.

Fourthly the method can be extended to less geometrically similar configuration to incorporate more radical changes. This may pose specific requirement and challenges for making consistent meshes and more sophisticated transformations of the computational domain for the source term propagation. The efforts will certainly be worthwhile for a much enhanced applicability of this kind of efficient high-fidelity methods as developed in the present work.

Finally, the situation where the efficient LES method can be more effective still needs further investigation. This includes applying the method to more optimization cases, using different number of initial samples for the LES evaluations, testing different ranges of the design variables, etc.

References

Abramson, J. (1977). *"Two-Dimensional Subsonic Wind Tunnel Evaluation of Two Related Cambered 15-Percent Thick Circulation Control Airfoils"*. DTIC Document.

Adamczyk, J. J. (1985). *"Model Equation for Simulating Flows in Multistage Turbomachinery"*. ASME Paper 85-GT-226.

Ahn, J., H. J. Kim, D. H. Lee, and O. H. Rho (2001). *"Response Surface Method for Airfoil Design in Transonic Flow"*. Journal of Aircraft, 38(2):231-238.

Anderson, W. K. and D. L. Bonhaus (1999). "*Airfoil Design on Unstructured Grids for Turbulent Flows"*. AIAA Journal, 37(2):185-191.

Andrews, C., J. Davies, and G. Schwarz (1967). *"Adaptive Data Compression"*. In Proceedings of the IEEE, volume 55. IEEE.

Armfield, S. and S. Street (1999). *"The Fractional-Step Method for the Navier-Stokes Equations on Staggered Grids: The Accuracy of Three Variations"*. Journal of Computational Physics ,153(2):660-665.

Balabanov, V., M. Kaufman, D. Knill, D. Haim, O. Golovidov, A. Giunta, R. T., Haftka, B. Grossman, W. Mason, and L. Watson (1996). *"Dependence of Optimal Structural Weight on Aerodynamic Shape for A High Speed Civil Transport"*. In 6th AIAA/NASA/USAF Multidisciplinary Analysis & Optimization Symposium.

Barigozzi, G., A. Armellini, C. Mucignat, and L. Casarsa (2012). *"Experimental Investigation of the Effects of Blowing Conditions and Mach Number on the Unsteady Behavior of Coolant Ejection through a Trailing Edge Cutback"*. International Journal of Heat and Fluid Flow, 37:37-50.

Beaudan, P. and P. Moin (1994). *"Numerical Experiments on the Flow past A Circular at Sub-Critical Reynolds Number"*. Report No. TF-62, Stanford University.

Benson, M. J. (2011). *"3D Velocity and Scalar Field Diagnostics using Magnetic Resonance Imaging with Applications in Film Cooling"*. PhD thesis, Stanford University.

Benson, M. J., C. J. Elkins, S. D. Yapa, J. B. Ling, and J. K. Eaton (2012). *"Effects of Varying Reynolds Number, Blowing Ratio, and internal Geometry on Trailing Edge Cutback Film Cooling"*. Experiments in Fluids, 52(6):1415-1430.

Breuer, M. (1998). *"Large Eddy Simulation of the Subcritical Flow past a Circular Cylinder: Numerical and Modeling Aspects"*. International Journal for Numerical Methods in Fluids, 28(9):1281-1302.

Buckley, H. P., B. Y. Zhou, and D. W. Zingg (2010). *"Airfoil Optimization using Practical Aerodynamic Design Requirements"*. Journal of Aircraft, 47(5):1707-1719.

Burguburu, S., C. Toussaint, C. Bonhomme, and G. Leroy (2004). *"Numerical Optimization of Turbomachinery Bladings"*. Journal of Turbomachinery, 126(1):91-100.

Busby, J., D. Sondak, B. Staubach, and R. Davis (1999). *"Deterministic Stress Modeling of Hot Gas Segregation in a Turbine"*. Journal of Turbomachinery, 122(1):62-67.

Castonguay, P. and S. K. Nadarajah (2007). *"Effect of Shape Parameterization on Aerodynamic Shape Optimization"*. In 45th AIAA Aerospace Sciences Meeting and Exhibit. AIAA.

Chang, P. A., J. Slomski, T. Marino, and M. P. Ebert (2005). *"Numerical Simulation of Two-And Three-Dimensional Circulation Control Problems"*. In 43rd AIAA Aerospace Sciences Meeting and Exhibit. AIAA.

Charbonnier, D. and F. Leboeuf (2004). *"Steady Flow Simulation of Rotor-Stator Interactions with a New Unsteady Flow Model"*. In 40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit. AIAA.

Chernukhin, O. and D. W. Zingg (2013). *"Multimodality and Global Optimization in Aerodynamic Design"*. AIAA Journal, 51(6):1342-1354.

Cox, D. D. and S. John (1997). *"SDO: A Statistical Method for Global Optimization"*. In Multidisciplinary Design Optimization: State-of-the-Art. SIAM.

Duvigneau, R. and P. Chandrashekar (2012). *"Kriging-Based Optimization Applied To Flow Control"*. International Journal for Numerical Methods in Fluids, 69(11):1701-1714.

Duvigneau, R. and M. Visonneau (2004). *"Hydrodynamic Design using A Derivative-Free Method"*. Structural and Multidisciplinary Optimization, 28(2-3):195-205.

Egorov, Y., F. Menter, R. Lechner, and D. Cokljat (2010). *"The Scale-Adaptive Simulation Method for Unsteady Turbulent Flow Predictions, Part 2: Application to Complex Flows"*. Flow, Turbulence and Combustion, 85(1):139-165.

Elazar, Y. (1988). *"A Mapping of the Viscous Flow Behavior in A Controlled Diffusion Compressor Cascade using Laser Doppler Velocimetry and Preliminary Evaluation of Codes for the Prediction of Stall"*. PhD thesis, Naval Postgraduate School.

Elliott, J. and J. Peraire (1997). *"Practical Three-Dimensional Aerodynamic Design and Optimization using Unstructured Meshes"*. AIAA Journal, 35(9):1479-1485.

Englar, R. J. (1979). *"Development of the A-6/Circulation Control Wing Flight Demonstrator Configuration"*. DTIC Document.

Ephraim, Y. and H. L. Van Trees (1995). *"A Signal Subspace Approach for Speech Enhancement"*. IEEE Transactions on Speech and Audio Processing, 3(4):251-266.

Epstein, B. and S. Peigin (2005). *"Constrained Aerodynamic Optimization of Three-Dimensional Wings Driven by Navier-Stokes Computations"*. AIAA Journal, 43(9):1946-1957.

Fang, K. T., D. K. Lin, P. Winker, and Y. Zhang (2000). *"Uniform Design: theory and Application"*. Technometrics, 42(3):237-248.

Fernandez, E., R. Kumar, and F. Alvi (2013). *"Separation Control on A Low-Pressure*

Turbine Blade using Microjets". Journal of Propulsion and Power, 29(4):867-881.

Ferziger, J. H. and M. Peric (2002). *"Computational Methods for Fluid Dynamics"*. 3rd ed. Springer.

Forrester, A. I. and A. J. Keane (2009). *"Recent Advances in Surrogate-Based Optimization"*. Progress in Aerospace Sciences, 45(1):50-79.

Franke, J. and W. Frank (2002). *"Large Eddy Simulation of the Flow past A Circular Cylinder at Re_D=3900"*. Journal of Wind Engineering and Industrial Aerodynamics, 90(10):1191-1206.

Gad-el-Hak, M. (2001). *"Flow Control: the Future"*. Journal of Aircraft, 38(3):402-418.

Galbraith, M. C. and P. D. Orkwis (2008). *"Transport Equation for Deterministic Stresses"*. In 46th AIAA Aerospace Sciences Meeting and Exhibit. AIAA.

Gano, S. E., H. Kim, and D. E. Brown (2006). *"Comparison of Three Surrogate Modeling Techniques: Datascape, Kriging, and Second Order Regression"*. In 11th AIAA/ISSMO Multidisciplinary analysis and Optimization Conference. AIAA.

Garnier, E., N. Adams, and P. Sagaut (2009). *"Large Eddy Simulation for Compressible Flows"*. Springer Science & Business Media.

Gbadebo, S. A., N. A. Cumpsty, and T. P. Hynes (2008). *"Control of Three-Dimensional Separations in Axial Compressors by Tailored Boundary Layer Suction"*. Journal of Turbomachinery, 130(1):011004.

Georgiadis, N. J., D. P. Rizzetta, and C. Fureby (2010). *"Large-Eddy Simulation: Current Capabilities, Recommended Practices, and Future Research"*. AIAA Journal, 48(8):1772-1784.

Giunta, a. and L. Watson (1998). *"A comparison of Approximation Modeling Techniques - Polynomial versus Interpolating Models"*. In 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary analysis and Optimization. AIAA.

Giunta, A. A., S. F. Wojtkiewicz, M. S. Eldred, et al. (2003). *"Overview of Modern Design of Experiments Methods for Computational Simulations"*. In Proceedings of the 41st AIAA Aerospace Sciences Meeting and Exhibit.

Gmelin, C., V. Zander, M. Hecklau, F. Thiele, W. Nitsche, a. Huppertz, and M. Swoboda (2012). *"Active Flow Control Concepts on a Highly Loaded Subsonic Compressor Cascade: Resume of Experimental and Numerical Results"*. Journal of Turbomachinery, 134(6):061021.

Hall, K. C., J. P. Thomas, and W. S. Clark (2002). *"Computation of Unsteady Nonlinear Flows in Cascades using A Harmonic Balance Technique"*. AIAA Journal, 40(5):879-886.

Han, S.-P. (1977). *"A Globally Convergent Method for Nonlinear Programming. Journal of Optimization theory and Applications"*, 22(3):297-309.

Han, Z. H., K. S. Zhang, W. P. Song, and Z. D. Qiao (2010). *"Optimization of Active Flow*

Control over an Airfoil using a Surrogate-Management Framework". Journal of Aircraft, 47(2):603-612.

He, L. (2008). *"Harmonic Solution of Unsteady Flow Around Blades with Separation"*. AIAA Journal, 46(6):1299-1307.

He, L. and W. Ning (1998). *"Efficient Approach for Analysis of Unsteady Viscous Flows in Turbomachines"*. AIAA Journal, 36(11):2005-2012.

He, L. and D. Wang (2011). *"Concurrent Blade Aerodynamic-Aero-Elastic Design Optimization using Adjoint Method"*. Journal of Turbomachinery, 133(1):011021.

Hicks, R. M. and P. A. Henne (1978). *"Wing Design by Numerical Optimization"*. Journal of Aircraft, 15(7):407-412.

Hicks, R. M., E. M. Murman, and G. N. Vanderplaats (1974). *"An Assessment of Airfoil Design by Numerical Optimization"*. NASA TM X-3092.

Holloway, D. S., J. H. Leylek, and F. A. Buck (2002a). *"Pressure-Side Bleed Film Cooling: Part I: Steady Framework for Experimental and Computational Results"*. In Turbo Expo 2002. ASME.

Holloway, D. S., J. H. Leylek, and F. A. Buck (2002b). *"Pressure-Side Bleed Film Cooling: Part II - Unsteady Framework for Experimental and Computational Results"*. In Turbo Expo 2002. ASME.

Holmes, P., J. L. Lumley, G. Berkooz, and C. W. Rowley (2012). *"Turbulence, Coherent Structures, Dynamical Systems and Symmetry"*. Cambridge University Press.

Huang, L., G. Huang, R. P. LeBeau, and T. Hauser (2007). *"Optimization of Aifoil Flow Control using a Genetic Algorithm with Diversity Control"*. Journal of Aircraft, 44(4):1337-1349.

Jasak, H., H. G. Well, and A. D. Gosman (1999). *"High Resolution NVD Differencing Scheme for Arbitrarily Unstructured Meshes"*. International Journal for Numerical Methods in Fluids, 31(2):431-449.

Jameson, A. (1988). *"Aerodynamic Design via Control Theory"*. Journal of Scientific Computing, 3(3):233-260.

Jeong, S., M. Murayama, and K. Yamamoto (2005). *"Efficient Optimization Design Method using Kriging Model"*. Journal of Aircraft, 42(2):413-420.

Jones, D. R. (2001). *"A Taxonomy of Global Optimization Methods Based on Response Surfaces"*. Journal of Global Optimization, 21(4):345-383.

Jones, G. S., S. A. Viken, A. E. Washburn, L. N. Jenkins, and C. M. Cagle (2002). *"An Active Flow Circulation Controlled Flap Concept for General Aviation Aircraft Applications"*. In 1st Flow Control Conference. AIAA.

Joo, J. and P. Durbin (2009). *"Simulation of Turbine Blade Trailing Edge Cooling"*. Journal of Fluids Engineering, 131(2):021102.

Kravchenko, A. G. and P. Moin (2000). *"Numerical Studies of Flow over A Circular*

Cylinder at Re_D=3900". Physics of Fluid, 12(2):403-417.

Koch, P. N., T. W. Simpson, J. K. Allen, and F. Mistree (1999). *"Statistical Approximations for Multidisciplinary Design Optimization: the Problem of Size"*. Journal of Aircraft, 36(1):275-286.

Lee, K. and S. Eyi (1993). *"Transonic Airfoil Design by Constrained Optimization"*. Journal of Aircraft, 30(6):805-806.

Leonard, B. (1991). *"The ULTIMATE Conservative Difference Scheme Applied To Unsteady One-Dimensional Advection"*. Computer Methods in Applied Mechanics and Engineering, 88(1):17-74.

Lepine, F. Guibault, J. Y. Trepanier, and F. Pepin (2001). *"Optimized Non-Uniform Rational B-Spline Geometrical Representation for Aerodynamic Design of Wings"*. AIAA Journal, 39(11):2033-2041.

Lin, J. C., S. K. Robinson, R. J. McGhee, W. O. Valarezo (1994). *"Separation Control on High-Lift Airfoils via Micro-Vortex Generators"*. Journal of Aircraft, 31(6):1317-23.

Lin, J. C. (2002). *"Review of Research on Low-Profile Vortex* Generators to *Control Boundary-Layer Separation"*. Progress in Aerospace Sciences, 38(4):389-420.

Lin, J. C., S. K. Robinson, R. J. McGhee, and W. O. Valarezo (1994). *"Separation Control on High-Lift Airfoils via Micro-Vortex Generators"*. Journal of Aircraft, 31(6):1317-1323.

Ling, J., S. D. Yapa, M. J. Benson, C. J. Elkins, and J. K. Eaton (2013). *"Three-Dimensional Velocity and Scalar Field Measurements of an Airfoil Trailing Edge with Slot Film Cooling: the Efect of an Internal Structure in the Slot"*. Journal of Turbomachinery, 135(3):031018.

Ling, J., C. J. Elkins, and J. K. Eaton (2014). *"Optimal Turbulent Schmidt Number for RANS Modeling of Trailing Edge Slot Film Cooling"*. In Turbo Expo 2014. ASME.

Ling, J., C. J. Elkins, and J. K. Eaton (2015). *"The Effect of Land Taper Angle on Trailing Edge Slot Film Cooling"*. Journal of Turbomachinery, 137(7):071003.

Lint, J. H. and R. M. Wilson (2001). *"A Course in Combinatorics"*. 2nd-ed. Cambridge University Press.

Lophaven, S., H. Nielsen, and J. Sondergaard (2002). *"DACE: A MATLAB Kriging Toolbox"*, Version 2.0. IMM-TR-2002-12, Technical University of Denmark.

Lukovic, B., P. D. Orkwis, M. G. Turner, and B. Sekar (2006). *"Neural Network Modeling of Deterministic Unsteadiness Source Terms"*. AIAA Journal, 44(7):1609-1616.

Lumley, J. L. (1967). *"The Structure of inhomogeneous Turbulent Flows"*. Atmospheric Turbulence and Wave Propagation, Pp. 166-178.

Luo, J., J. Xiong, F. Liu, and I. McBean (2011). *"Three-Dimensional Aerodynamic Design Optimization of a Turbine Blade by using an Adjoint Method"*. Journal of Turbomachinery, 133(1):011026.

Lyu, Z. and J. Martins (2014). *"Aerodynamic Design Optimization Studies of a Blended-Wing-Body Aircraft"*. Journal of Aircraft, 51(5):1604-1617.

Martin, J. D. and T. W. Simpson (2005). *"Use of Kriging Models to Approximate Deterministic Computer Models"*. AIAA Journal, 43(4):853-863.

Martini, P., A. Schulz, and H.-J. Bauer (2005a). *"Film Cooling Effectiveness and Heat Transfer on the Trailing Edge Cutback of Gas Turbine Airfoils with Various internal Cooling Designs"*. Journal of Turbomachinery, 128(1):196-205.

Martini, P., A. Schulz, H.-J. Bauer, and C. Whitney (2005b). *"Detached Eddy Simulation of Film Cooling Performance on the Trailing Edge Cutback of Gas Turbine Airfoils"*. Journal of Turbomachinery, 128(2):292-299.

McGhee, R. and W. Beasley (1973). *"Low Speed Aerodynamic Characteristics of a 17-Percent Thick Airfoil Section Designed for General Aviation Applications"*. NASA TN D-7428

McKay, M. D., R. J. Beckman, and W. J. Conover (2000). *"A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code"*. Technometrics, 42(1):55-61.

McMullen, M. S. (2003). *"The Application of Non-Linear Frequency Domain Methods To the Euler and Navier-Stokes Equations"*. PhD thesis, Stanford University.

Medic, G. and P. A. Durbin (2005). *"Unsteady Effects on Trailing Edge Cooling"*. Journal of Heat Transfer, 127(4):388-392.

Menter, F. R. (1994). *"Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications"*. AIAA Journal, 32(8):1598-1605.

Meunier, M. (2009). *"Simulation and Optimization of Flow Control Strategies for Novel High-Lift Configurations"*. AIAA Journal, 47(5):1145-1157.

Min, b.-Y., W. Lee, R. Englar, and L. N. Sankar (2009). *"Numerical investigation of Circulation Control Airfoils"*. Journal of Aircraft, 46(4):1403-1410.

Murata, A., S. Nishida, H. Saito, K. Iwamoto, Y. Okita, and C. Nakamata (2012). *"Effects of Surface Geometry on Film Cooling Performance at Airfoil Trailing Edge"*. Journal of Turbomachinery, 134(5):051033.

Myers, R. H., D. C. Montgomery, and C. M. Anderson Cook (2016). *"Response Surface Methodology: Process and Product Optimization using Designed Experiments"*. John Wiley & Sons.

Myers, R. H., D. C. Montgomery, G. G. Vining, C. M. Borror, and S. M. Kowalski (2004). *"Response Surface Methodology: A Retrospective and Literature Survey"*. Journal of Quality Technology, 36(1):53.

Narducci, R., B. Grossman, M. Valorani, A. Dadone, and R. Haftka (1995). *"Optimization Methods for Non-Smooth or Noisy Objective Functions in Fluid Design Problems"*. In 12th Computational Fluid Dynamics Conference. AIAA.

Nielsen, E. and W. K. Anderson (1999). "*Aerodynamic Design Optimization on Unstructured Meshes using the Navier-Stokes Equations"*. AIAA Journal, 37(11):1411-1419.

Nielsen, E. and W. Jones (2011). *"Integrated Design of an Active Flow Control System using A Time-Dependent Adjoint Method"*. Mathematical Modelling of Natural Phenomena, 6(03):141-165.

Ning, W. and L. He (2001). *"Some Modeling Issues on Trailing-Edge Vortex Shedding"*. AIAA Journal, 39(5):787-793.

Norberg, C. (2003). *"Fluctuating Lift on a Circular Cylinder: Review and New Measurements"*. Journal of Fluids and Structures, 17(1):57-96.

Okui, H., T. Verstraete, R. Van den Braembussche, and Z. Alsalihi (2013). *"Three-Dimensional Design and Optimization of a Transonic Rotor in Axial Flow Compressors"*. Journal of Turbomachinery, 135(3):031009.

Okui, H., T. Verstraete, R. Van den Braembussche, and Z. Alsalihi (2013). *"Three-Dimensional Design and Optimization of a Transonic Rotor in Axial Flow Compressors"*. Journal of Turbomachinery, 135(3):031009.

Osnaghi, C., A. Perdichizzi, M. Savini, P. Harasgama and E. Lutum (1997). *"The Influence of Film-Cooling on the Aerodynamic Performance of a Turbine Nozzle Guide Vane"*. In ASME 1997 International Gas Turbine and Aeroengine Congress and Exhibition.

Owen, A. B. (1992). *"Orthogonal Arrays for Computer Experiments, Integration and Visualization"*. Statistica Sinica, 2(2):439-452.

Painchaud-Ouellet, S., C. Tribes, J.-Y. Trepanier, and D. Pelletier (2006). *"Airfoil Shape Optimization using a Non-Uniform Rational B-Splines Parametrization under Thickness Constraint"*. AIAA Journal, 44(10):2170-2178.

Paiva, R. M., a. R. D. Carvalho, C. Crawford, and A. Suleman (2010). *"Comparison of Surrogate Models in A Multidisciplinary Optimization Framework for Wing Design"*. AIAA Journal, 48(5):995-1006.

Parnaudeau, P., J. Carlier, D. Heitz, and E. Lamballais (2008). *"Experimental and Numerical Studies of the Flow over a Circular Cylinder at Reynolds Number 3900"*. Physics of Fluids, 20(8):085101.

Pehlivanoglu, Y. V. and B. Yagiz (2011). *"Optimization of Active/Passive Flow Control Parameters on Airfoils at Transonic Speeds"*. Journal of Aircraft, 48(1):212-219.

Peri, D. and F. Tinti (2012). *"A Multistart Gradient-Based Algorithm with Surrogate Model for Global Optimization"*. Communications in Applied and Industrial Mathematics, 3(1).

Queipo, N. V., R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker (2005). *"Surrogate-Based analysis and Optimization"*. Progress in Aerospace Sciences, 41(1):1-28.

Rallabhandi, S. K. and D. N. Mavris (2007). *“New Computational Procedure for Incorporating Computational Fluid Dynamics into Sonic Boom Prediction”*. Journal of Aircraft, 44(6):1964-1971.

Ravelli, S. and G. Barigozzi (2013). *“Evaluation of RANS Predictions on a Linear Nozzle Vane Cascade with Trailing Edge Cutback Film Cooling”*. In Turbo Expo 2013. ASME.

Ravelli, S. and G. Barigozzi (2014). *“Application of Unsteady CFD Methods to Trailing Edge Cutback Film Cooling”*. In Turbo Expo 2014. ASME.

Reijnen, D. P. (1997). *“Experimental Study of Boundary Layer Suction in a Transonic Compressor”*. PhD thesis, Massachusetts institute of Technology.

Reuther, J., J. J. Alonso, M. J. Rimlinger, and a. Jameson (1999). *“Aerodynamic Shape Optimization of Supersonic Aircraft Configurations via an Adjoint Formulation on Distributed Memory Parallel Computers”*. Computers & Fluids, 28(4):675-700

Rizzetta, D. P. and M. R. Visbal (2003). *“Large-Eddy Simulation of Supersonic Cavity Flow fields Including Flow Control”*. AIAA Journal, 41(8):1452-1462.

Rizzetta, D. P., M. R. Visbal, and P. E. Morgan (2008). *“A High-Order Compact Finite-Difference Scheme for Large-Eddy Simulation of Active Flow Control”*. Progress in Aerospace Sciences, 44(6):397-426.

Rogers, D. F. (2000). *“An introduction To NURBS: with Historical Perspective”*. Elsevier.

Rooij, M. P. C., T. Q. Dang, and L. M. Larosiliere (2005). *“Improving Aerodynamic Matching of Axial Compressor Blading using a Three-Dimensional Multistage Inverse Design Method”*. Journal of Turbomachinery, 129(1):108-118.

Rosenfeld, A. and A. C. Kak (2014). *“Digital Picture Processing”*. Elsevier.

Rumsey, C. L. and T. Nishino (2011). *“Numerical Study Comparing RANS and LES Approaches on A Circulation Control Airfoil”*. International Journal of Heat and Fluid Flow, 32(5):847-864.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn (1989). *“Design and analysis of Computer Experiments”*. Statistical Science, Pp. 409-423.

Sagaut, P. (2006). *“Large Eddy Simulation for Incompressible Flows: an Introduction”*. Springer Science & Business Media.

Sanger, N. (1983). *“The Use of Optimization Techniques to Design-Controlled Diffusion Compressor Blading”*. Journal of Engineering for Power, 105(2):257.

Schabowski, Z., H. Hodson, D. Giacche, B. Power, and M. R. Stokes (2014). *“Aeromechanical Optimization of A Winglet-Squealer Tip for an Axial Turbine”*. Journal of Turbomachinery, 136(7):071004.

Schlecht, R. and S. Anders (2007). *“Parametric Evaluation of Thin, Transonic Circulation Control Airfoils”*. In 45th AIAA Aerospace Sciences Meeting and Exhibit.

Schneider, H., D. von Terzi, and H.-J. Bauer (2012). *"Turbulent Heat Transfer and Large Coherent Structures in Trailing-Edge Cutback Film Cooling"*. Flow, Turbulence and Combustion, 88(1-2):101-120.

Sieverding, C. and H. Heinemann (1990). *"The Influence of Boundary Layer State on Vortex Shedding from Flat Plates and Turbine Cascades"*. Journal of Turbomachinery, 112(2):181-187.

Simpson, T. W., T. M. Mauery, J. J. Korte, and F. Mistree (2001). *"Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization"*. AIAA Journal, 39(12):2233-2241.

Slomski, J., P. Chang, and S. Arunajatesan (2006). *"Large Eddy Simulation of a Circulation Control Airfoil"*. In 3rd AIAA Flow Control Conference. AIAA.

Slomski, J., J. Gorski, R. Miller, and T. Marino (2002). *"Numerical Simulation of Circulation Control Airfoils as Affected by Different Turbulence Models"*. In 40th AIAA Aerospace Sciences Meeting and Exhibit. AIAA.

Smargorinsky, J. (1963). *"General Circulation Experiments with the Primitive Equation I"*. Monthly Weather Review, 91(3):99-164.

Sondak, D. L., D. J. Dorney, and R. L. Davis (1996). *"Modeling Turbomachinery Unsteadiness with Lumped Deterministic Stresses"*. In 32nd Joint Propulsion Conference and Exhibit. AIAA.

Song, W. and a. J. Keane (2007). *"Surrogate-Based Aerodynamic Shape Optimization of a Civil Aircraft Engine Nacelle"*. AIAA Journal, 45(10):2565-2574.

Sonoda, T., Y. Yamaguchi, T. Arima, M. Olhofer, B. Sendhoff, and H. A. Schreiber (2004). *"Advanced High Turning Compressor Airfoils for Low Reynolds Number Condition. Part I: Design and Optimization"*. Journal of Turbomachinery, 126(3):350-359.

Spalart, P. and S. Allmaras (1992). *"A One-Equation Turbulence Model for Aerodynamic Flows"*. Technical Report AIAA-92-0439.

Spalart, P. R. (2000). *"Strategies for Turbulence Modelling and Simulations"*. International Journal of Heat and Fluid Flow, 21(3):252-263.

Stollenwerk, S. and E. Kugeler (2013). *"Deterministic Stress Modeling for Multistage Compressor Flow Fields"*. In Turbo Expo 2013. ASME.

Stollenwerk, S. and D. Nuernberger (2008). *"A Model for Potential Deterministic Effects in Turbomachinery Flows"*. In Proceedings of the World Congress on Engineering, London, 2008.

Swanson, R. and C. Rumsey (2009). *"Computation of Circulation Control Airfoil Flows"*. Computers & Fluids, 38(10):1925-1942.

Swanson, R., C. L. Rumsey, and S. Anders (2005). *"Progress Towards Computational Method for Circulation Control Airfoils"*. In 43rd AIAA Aerospace Sciences Meeting and Exhibit. AIAA.

Tai, T. C. and G. Kidwell (1985). *“Numerical Optimization of Circulation Control Airfoil at High Subsonic Speed”*. Journal of Aircraft, 22(10):869-874.

Tai, T. C., G. H. Kidwell Jr, and G. N. Vanderpiaats (1982). *“Numerical Optimization of Circulation Control Airfoils”*. Journal of Aircraft, 19(2):145-150.

Taslim, M., S. Spring, and B. Mehlman (1992). *“Experimental Investigation of Film Cooling Effectiveness for Slots of Various Exit Geometries”*. Journal of Thermophysics and Heat Transfer, 6(2):302-307.

Ugray, Z., L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Marti (2007). *“Scatter Search and Local NLP Solvers: A Multistart Framework for Global Optimization”*. INFORMS Journal on Computing, 19(3):328-340.

Vavalle, A. and N. Qin (2007). *“Iterative Response Surface Based Optimization Scheme for Transonic Airfoil Design”*. Journal of Aircraft, 44(2):365-376.

Viana, F. A., T. W. Simpson, V. Balabanov, and V. Toropov (2014). *“Special Section on Multidisciplinary Design Optimization: Metamodeling in Multidisciplinary Design Optimization: How Far Have We Really Come?”*. AIAA Journal, 52(4):670-690.

Vicini, A. and D. Quagliarella (1999). *“Airfoil and Wing Design through Hybrid Optimization Strategies”*. AIAA Journal, 37(5):634-641.

Vreman, A. W. (1995). *“Direct and Large Eddy Simulation of the Compressible Turbulent Mixing Layer”*. PhD Dissertation, University of Twente, Twente.

Wall, A. G., J. R. Kadambi, and J. J. Adamczyk (2000). *“A Transport Model for the Deterministic Stresses Associated with Turbomachinery Blade Row Interactions”*. In Turbo Expo 2000. ASME.

Walther, B. and S. Nadarajah (2015). *“Adjoint-Based Constrained Aerodynamic Shape Optimization for Multistage Turbomachines”*. Journal of Propulsion and Power, 31(5):1298-1319.

Wang, D. and L. He (2010). *“Adjoint Aerodynamic Design Optimization for Blades in Multistage Turbomachines Part I: Methodology and Verification”*. Journal of Turbomachinery, 132(2):021011.

Wang, D., L. He, Y. Li, and R. Wells (2010). *“Adjoint Aerodynamic Design Optimization for Blades in Multistage Turbomachines Part II: Validation and Application”*. Journal of Turbomachinery, 132(2):021012.

Wendt, B. J. and J. C. Dudek (1998). *“Development of Vortex Generator Use for a Transitioning High-Speed Inlet”*. Journal of Aircraft, 35(4):536-543.

White, F. M. (2003). *“Fluid Mechanics”*. 5th-ed. Boston: McGraw-Hill Book Company.

Wu, H. Y., S. Yang, F. Liu, and H.-M. Tsai (2003). *“Comparison of Three Geometric Representations of Airfoils for Aerodynamic Optimization”*. In 16th AIAA Computational Fluid Dynamics Conference. AIAA.

Yagiz, B., O. Kandil, and Y. V. Pehlivanoglu (2012). *“Drag Minimization using Active* *and Passive Flow Control Techniques"*. Aerospace Science and Technology, 17(1):21-31.

Yang, J., D. Zhang, A. F. Frangi, and J.-y. Yang (2004). *"Two-Dimensional PCA: a New Approach to Appearance-Based Face Representation and Recognition"*. IEEE Transactions on Pattern analysis and Machine intelligence, 26(1):131-137.

You, D. and P. Moin (2008). *"Active Control of Flow Separation over an Airfoil using Synthetic Jets"*. Journal of Fluids and Structures, 24(8):1349-1357.

You, D., M. Wang, and P. Moin (2006). *"Large-Eddy Simulation of Flow over a Wall-Mounted Hump with Separation Control"*. AIAA Journal, 44(11):2571-2577.

Zhang, M. and L. He (2014). *"Combining Shaping and Flow Control for Aerodynamic Optimization"*. AIAA Journal, 53(4):888-901.

Zingg, D. W. and S. Elias (2006). *"Aerodynamic Optimization under a Range of Operating Conditions"*. AIAA Journal, 44(11):2787-2792.

Appendix A

Uniform Design Tables

In the uniform design table, the DOE samples in each dimension are chosen based on the number of design variables N_V and the desire number of samples N_S. For each number of design variables, multiple choices for the number of samples may exist which is subject to the users' choice. Some examples of the table used in the present study are listed below. According to the tables, each dimension of the design space is scale to range between 1 and 6. Each row represents the values for all the design variables of a sample, whereas each column represents the values of a design variable at the sample design sites. A complete set of the tables can be found in the following website:

http://uic.edu.hk/isci/UniformDesign/table/CD2.htm

A.1 $N_V = 4, N_S = 30$

d_1	d_2	d_3	d_4
4	1	1	1
5	4	2	2
6	1	4	2
4	1	6	5
4	5	4	4
3	2	3	2
2	4	3	1
2	2	1	4
5	2	3	4
3	5	2	3
6	6	1	5
6	2	5	6
2	6	6	6
5	5	3	6
4	3	1	6
3	3	4	5
1	2	6	1
2	1	4	3
2	3	5	3
1	6	3	4
3	6	5	2
5	3	6	4
6	5	6	3

d_1	d_2	d_3	d_4
1	4	5	5
5	6	4	1
1	1	2	6
6	3	2	1
3	4	2	5
4	4	5	2
1	5	1	2

A.2 $N_V = 6, N_S = 48$

d_1	d_2	d_3	d_4	d_5	d_6
6	1	5	3	6	5
2	5	3	2	1	4
1	5	2	1	6	3
4	1	1	5	2	3
3	5	2	6	3	4
2	2	3	4	2	6
1	2	2	4	6	2
2	2	1	2	4	1
6	4	3	4	5	1
3	6	4	2	6	1
4	6	4	1	3	5
6	3	1	5	3	2
5	6	1	3	5	2
1	3	5	4	1	3
5	5	3	5	6	5
5	1	6	4	3	1
1	3	4	5	4	5
1	5	3	6	2	1
4	6	2	5	4	3
6	2	2	1	1	4
5	2	4	6	1	2
1	4	1	3	3	4
5	4	4	1	2	1
4	3	1	2	6	6
6	6	5	6	2	4
5	1	2	6	4	6
3	6	1	4	1	5
2	3	4	3	2	3
6	5	4	4	4	6
1	6	5	3	5	6
2	6	6	5	3	2
3	1	2	3	1	1
2	4	5	1	4	2
3	4	6	4	6	4
2	4	1	6	5	5
4	2	3	3	4	5

d_1	d_2	d_3	d_4	d_5	d_6
4	4	6	5	1	6
6	3	3	2	3	3
4	3	5	6	6	1
4	5	5	3	1	2
3	1	3	1	5	2
6	5	6	2	4	3
5	4	2	2	2	6
3	2	5	1	3	6
2	1	4	5	5	4
3	2	6	6	5	3
5	3	6	1	5	4
1	1	6	2	2	5

A.3 $N_V = 10, N_S = 60$

d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
1	6	5	6	4	5	1	4	2	3
4	2	6	1	6	2	3	1	5	3
1	3	1	1	5	4	2	2	1	2
4	1	2	5	4	4	5	1	1	6
5	1	3	1	1	3	3	5	2	5
4	4	2	6	1	4	2	5	3	1
2	5	1	5	2	6	3	3	1	5
6	2	2	3	5	1	3	4	6	1
3	2	1	5	4	6	3	5	6	2
2	1	4	2	6	4	6	4	3	2
3	5	3	4	3	1	1	6	1	2
1	1	3	3	3	4	2	6	6	4
1	5	6	1	2	4	4	5	3	3
1	2	5	3	2	2	5	4	2	6
5	4	4	1	5	5	2	6	6	3
6	5	5	1	4	5	3	3	3	1
2	4	4	2	1	2	3	4	1	1
1	1	4	5	2	1	2	3	4	1
5	6	4	5	4	1	2	2	6	5
6	1	5	5	5	4	6	5	5	4
5	2	5	4	3	3	5	6	1	1
2	3	1	4	1	5	6	6	2	3
5	4	1	2	4	1	5	5	3	6
3	3	5	6	1	1	4	6	5	5
1	4	5	3	4	6	5	1	5	4
4	6	6	3	5	6	4	6	2	5
3	2	5	2	1	6	1	2	4	4
3	1	6	4	5	3	3	3	1	5
6	2	2	6	2	3	6	2	3	3
4	1	1	3	1	1	4	1	4	3

d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
2	6	1	1	2	3	5	4	6	4
2	5	4	6	5	3	3	1	4	6
6	3	3	2	2	6	5	4	5	2
1	3	3	4	6	2	6	3	6	5
1	5	2	5	6	3	4	6	5	3
5	3	6	3	2	5	2	1	1	4
4	6	5	3	6	3	1	5	4	2
5	2	1	6	5	2	1	4	2	4
3	3	6	2	4	2	6	6	4	1
6	4	1	3	6	4	1	2	5	5
1	6	2	2	3	2	2	1	3	5
6	6	4	4	1	4	4	1	2	2
3	4	3	4	2	3	1	1	6	1
2	4	6	5	5	1	5	2	2	2
5	3	3	5	6	6	4	1	3	1
3	2	2	1	3	5	4	2	5	6
2	2	4	4	6	6	2	5	3	6
4	5	5	2	1	4	6	3	6	6
5	4	4	6	3	6	6	4	4	5
6	6	2	4	2	5	3	6	4	6
2	3	3	1	4	1	1	5	4	6
6	3	6	5	3	3	1	3	3	6
3	6	2	3	5	5	6	2	4	1
6	5	3	2	6	2	4	3	1	4
5	5	6	4	1	2	2	4	5	3
3	4	3	6	6	5	5	5	1	4
4	5	4	1	3	1	6	2	2	4
2	1	6	6	3	5	4	2	6	2
4	6	1	6	3	2	5	3	5	2
4	1	2	2	4	6	1	3	2	3

Appendix B

Programming Codes

This chapter presents the Matlab code used for the source term propagation, and the C programing code for some Fluent user-defined functions (UDF) used in the efficient LES method.

B.1 Matlab Code for Source Term Propagation

B.1.1 Quadratic Regression in Cell to Cell Approach

```
% build quadadratic interpolation/approximiation of 2D design space
% suing leatp000 squared fitting
% in cell by cell propagation approach
clear all
close all
% clc
Ncell=3583800;
% ==== load sample data
% cid_uns in the loaded data are assumed ascending as exported from
serial Fluent.
% == taper=0 i.e. tp000raight land
%BR0.7
load('...\source term data\tp000\BR0.7\12e4\source_fluent.mat');
Sx_tp000_BR070=Sx_uns;
Sy_tp000_BR070=Sy_uns;
Sz_tp000_BR070=Sz_uns;
Se_tp000_BR070=Se_uns;
Sm_tp000_BR070=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
%BR1
load('...\source term data\tp000\BR1\12e4\source_fluent.mat');
Sx_tp000_BR100=Sx_uns;
Sy_tp000_BR100=Sy_uns;
Sz_tp000_BR100=Sz_uns;
Se_tp000_BR100=Se_uns;
Sm_tp000_BR100=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
%BR1.3
load('...\source term data\tp000\BR1.3\12e4\source_fluent.mat');
Sx_tp000_BR130=Sx_uns;
Sy_tp000_BR130=Sy_uns;
Sz_tp000_BR130=Sz_uns;
Se_tp000_BR130=Se_uns;
Sm_tp000_BR130=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
% == tp250
%BR0.7
load('...\source term data\tp250\BR0.7\12e4\source_fluent.mat');
Sx_tp250_BR070=Sx_uns;
Sy_tp250_BR070=Sy_uns;
```

```
Sz_tp250_BR070=Sz_uns;
Se_tp250_BR070=Se_uns;
Sm_tp250_BR070=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
%BR1
load('...\source term data\tp250\BR1\12e4\source_fluent.mat');
Sx_tp250_BR100=Sx_uns;
Sy_tp250_BR100=Sy_uns;
Sz_tp250_BR100=Sz_uns;
Se_tp250_BR100=Se_uns;
Sm_tp250_BR100=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
%BR1.3
load('...\source term data\tp250\BR1.3\12e4\source_fluent.mat');
Sx_tp250_BR130=Sx_uns;
Sy_tp250_BR130=Sy_uns;
Sz_tp250_BR130=Sz_uns;
Se_tp250_BR130=Se_uns;
Sm_tp250_BR130=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
% == tp500
%BR0.7
load('...\source term data\tp500\BR0.7\12e4\source_fluent.mat');
Sx_tp500_BR070=Sx_uns;
Sy_tp500_BR070=Sy_uns;
Sz_tp500_BR070=Sz_uns;
Se_tp500_BR070=Se_uns;
Sm_tp500_BR070=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
%BR1
load('...\source term data\tp500\BR1\12e4\source_fluent.mat');
Sx_tp500_BR100=Sx_uns;
Sy_tp500_BR100=Sy_uns;
Sz_tp500_BR100=Sz_uns;
Se_tp500_BR100=Se_uns;
Sm_tp500_BR100=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
%BR1.3
load('...\source term data\tp500\BR1.3\12e4\source_fluent.mat');
Sx_tp500_BR130=Sx_uns;
Sy_tp500_BR130=Sy_uns;
Sz_tp500_BR130=Sz_uns;
Se_tp500_BR130=Se_uns;
Sm_tp500_BR130=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
% ==== sample ensemble matrix
% == 9 samples in total
Sx_s=[Sx_tp000_BR070 Sx_tp000_BR100 Sx_tp000_BR130 ...
       Sx_tp250_BR070 Sx_tp250_BR100 Sx_tp250_BR130 ...
       Sx_tp500_BR070 Sx_tp500_BR100 Sx_tp500_BR130];
Sy_s=[Sy_tp000_BR070 Sy_tp000_BR100 Sy_tp000_BR130 ...
       Sy_tp250_BR070 Sy_tp250_BR100 Sy_tp250_BR130 ...
       Sy_tp500_BR070 Sy_tp500_BR100 Sy_tp500_BR130];
Sz_s=[Sz_tp000_BR070 Sz_tp000_BR100 Sz_tp000_BR130 ...
       Sz_tp250_BR070 Sz_tp250_BR100 Sz_tp250_BR130 ...
       Sz_tp500_BR070 Sz_tp500_BR100 Sz_tp500_BR130];
Se_s=[Se_tp000_BR070 Se_tp000_BR100 Se_tp000_BR130 ...
       Se_tp250_BR070 Se_tp250_BR100 Se_tp250_BR130 ...
       Se_tp500_BR070 Se_tp500_BR100 Se_tp500_BR130];
Sm_s=[Sm_tp000_BR070 Sm_tp000_BR100 Sm_tp000_BR130 ...
       Sm_tp250_BR070 Sm_tp250_BR100 Sm_tp250_BR130 ...
```

```
       Sm_tp500_BR070 Sm_tp500_BR100 Sm_tp500_BR130];
% ==== build quadratic regression coefficients beta for each cell
% regression function f=[d1^2 d2^2 d1d2 d1 d2 1]; d1:shape d2:FC
% == design matrix
d1_s=[-1 -1 -1 0 0 0 1 1 1];
d2_s=[-1 0 1 -1 0 1 -1 0 1];
for i=1:size(Sx_s,2);
   F(i,:)=[d1_s(i)^2 d2_s(i)^2 d1_s(i)*d2_s(i) d1_s(i) d2_s(i) 1];
end
% coefficients for the cells
disp('computing coefficients beta...')
bx=F\Sx_s';
by=F\Sy_s';
bz=F\Sz_s';
be=F\Se_s';
bm=F\Sm_s';
bx=bx';
by=by';
bz=bz';
be=be';
bm=bm';
disp('beta computed')
% ==== build new source
d1=0;
d2=-0.5;
f=[d1^2 d2^2 d1*d2 d1 d2 1]';
Sx_new=bx*f;
Sy_new=by*f;
Sz_new=bz*f;
Se_new=be*f;
Sm_new=bm*f;
disp('new source computed')
clearvars -except Sx_new Sy_new Sz_new Se_new Sm_new
%%  ===== wrtie to files
disp('start writting...')
Sx_uns=Sx_new;
Sy_uns=Sy_new;
Sz_uns=Sz_new;
Se_uns=Se_new;
Sm_uns=Sm_new;
source_w=[Sx_uns Sy_uns Sz_uns Se_uns Sm_uns];
fp_write=['...\source term data\source_fluent.txt'];
fid=fopen(fp_write,'wt');
fprintf(fid,'%.9e\t%.9e\t%.9e\t%.9e\t%.9e\n',source_w');
fclose('all');
disp('source terms written')
save('...\source term data\source_fluent.mat',...
    'Sx_uns','Sy_uns','Sz_uns','Se_uns','Sm_uns',...
    '-v6');
disp('source terms saved');
```

B.1.2 Kriging Interpolation in POD Approach

A. Compute POD Bases

```
% =================
% load sample source term data from 3x3 samples and compute the pod bases
clear all
close all
% clc
```

```
% Ncell=3583800; % dont need
% ==== load sample data
% cid_uns in the loaded data are assumed ascending as exported from
serial Fluent.
% == st
% construct st by reordering into st cid order
% cid_match(:,1) is cid sequence of st already  accending order
% cid_match(:,2) is cid sequence of st, from the perspective of st
% convert to correct cid sequence
%BR0.7
load('...\tp000\BR0.7\12e4\source_fluent.mat');
Sx_tp000_BR070=Sx_uns;
Sy_tp000_BR070=Sy_uns;
Sz_tp000_BR070=Sz_uns;
Se_tp000_BR070=Se_uns;
Sm_tp000_BR070=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
%BR1
load('...\tp000\BR1\12e4\source_fluent.mat');
Sx_tp000_BR100=Sx_uns;
Sy_tp000_BR100=Sy_uns;
Sz_tp000_BR100=Sz_uns;
Se_tp000_BR100=Se_uns;
Sm_tp000_BR100=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
%BR1.3
load('...\tp000\BR1.3\12e4\source_fluent.mat');
Sx_tp000_BR130=Sx_uns;
Sy_tp000_BR130=Sy_uns;
Sz_tp000_BR130=Sz_uns;
Se_tp000_BR130=Se_uns;
Sm_tp000_BR130=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
% == tp250
%BR0.7
load('...\tp250\BR0.7\12e4\source_fluent.mat');
Sx_tp250_BR070=Sx_uns;
Sy_tp250_BR070=Sy_uns;
Sz_tp250_BR070=Sz_uns;
Se_tp250_BR070=Se_uns;
Sm_tp250_BR070=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
%BR1
load('...\tp250\BR1\12e4\source_fluent.mat');
Sx_tp250_BR100=Sx_uns;
Sy_tp250_BR100=Sy_uns;
Sz_tp250_BR100=Sz_uns;
Se_tp250_BR100=Se_uns;
Sm_tp250_BR100=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
%BR1.3
load('...\tp250\BR1.3\12e4\source_fluent.mat');
Sx_tp250_BR130=Sx_uns;
Sy_tp250_BR130=Sy_uns;
Sz_tp250_BR130=Sz_uns;
Se_tp250_BR130=Se_uns;
Sm_tp250_BR130=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
% == tp500
%BR0.7
```

```
load('...\tp500\BR0.7\12e4\source_fluent.mat');
Sx_tp500_BR070=Sx_uns;
Sy_tp500_BR070=Sy_uns;
Sz_tp500_BR070=Sz_uns;
Se_tp500_BR070=Se_uns;
Sm_tp500_BR070=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
%BR1
load('...\tp500\BR1\12e4\source_fluent.mat');
Sx_tp500_BR100=Sx_uns;
Sy_tp500_BR100=Sy_uns;
Sz_tp500_BR100=Sz_uns;
Se_tp500_BR100=Se_uns;
Sm_tp500_BR100=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
%BR1.3
load('...\tp500\BR1.3\12e4\source_fluent.mat');
Sx_tp500_BR130=Sx_uns;
Sy_tp500_BR130=Sy_uns;
Sz_tp500_BR130=Sz_uns;
Se_tp500_BR130=Se_uns;
Sm_tp500_BR130=Sm_uns;
clearvars Sx_uns Sy_uns Sz_uns Se_uns Sm_uns
% ==== build POD bases by thin SVD
% == ensemble matrix with 9 snapshots
Snapx=[Sx_tp000_BR070 Sx_tp000_BR100 Sx_tp000_BR130 ...
       Sx_tp250_BR070 Sx_tp250_BR100 Sx_tp250_BR130 ...
       Sx_tp500_BR070 Sx_tp500_BR100 Sx_tp500_BR130];
Snapy=[Sy_tp000_BR070 Sy_tp000_BR100 Sy_tp000_BR130 ...
       Sy_tp250_BR070 Sy_tp250_BR100 Sy_tp250_BR130 ...
       Sy_tp500_BR070 Sy_tp500_BR100 Sy_tp500_BR130];
Snapz=[Sz_tp000_BR070 Sz_tp000_BR100 Sz_tp000_BR130 ...
       Sz_tp250_BR070 Sz_tp250_BR100 Sz_tp250_BR130 ...
       Sz_tp500_BR070 Sz_tp500_BR100 Sz_tp500_BR130];
Snape=[Se_tp000_BR070 Se_tp000_BR100 Se_tp000_BR130 ...
       Se_tp250_BR070 Se_tp250_BR100 Se_tp250_BR130 ...
       Se_tp500_BR070 Se_tp500_BR100 Se_tp500_BR130];
Snapm=[Sm_tp000_BR070 Sm_tp000_BR100 Sm_tp000_BR130 ...
       Sm_tp250_BR070 Sm_tp250_BR100 Sm_tp250_BR130 ...
       Sm_tp500_BR070 Sm_tp500_BR100 Sm_tp500_BR130];
tic
disp('SVD for POD basis...');
[Ux, Sigx, Vx]=svd(Snapx, 0);   %U is Ncell*9 matrix
[Uy, Sigy, Vy]=svd(Snapy, 0);
[Uz, Sigz, Vz]=svd(Snapz, 0);
[Ue, Sige, Ve]=svd(Snape, 0);
[Um, Sigm, Vm]=svd(Snapm, 0);
disp('SVD done');
toc
% === compute POD coefficients matrix W for each source term, with m
snapshots and m POD basis
% at d_i instance, format as:
% W=[w1(d_1) w1(d_2) ... w1(d_m)] -- w1(d) transposed, 1st POD coef
%   [w2(d_1) w2(d_2) ... w2(d_m)]
%   [        ...               ]
%   [wk(d_1) ...         wk(d_m)] -- wk(d) transposed, kth POD coef
%   ...
%   [wm(d_1) ...         wm(d_m)]
Wx=Sigx*Vx';     % 3*3 or m*m matrix
Wy=Sigy*Vy';
Wz=Sigz*Vz';
```

```
We=Sige*Ve';
Wm=Sigm*Vm';
%  ===== save the workspace
tic
disp('saving workspace ...')
save('...\Code\Optimization\pod_workspace_shape_FC_3x3.mat','-v6');
disp('workspace saved');
toc
```

B. Build Kriging Model for POD Coefficients

```
% Build the kriging model for the POD coefficients, shape + FC
% Need to run pod_shape_FC.m to obtain the POD bases first!
% works for 3x3 samples so there will be 9 POD coefficients
% use DACE Matlab Kriging toolbox
clear all
close all
% clc
% ==== load POD workspace
load('...\Code\Optimization\pod_workspace_shape_FC_3x3.mat',...
     'Ux','Uy','Uz','Ue','Um',...
     'Wx','Wy','Wz','We','Wm');
disp('pod bases');
% ==== preprocess data for Kriging
% correlation parameter
lb=[0.01 0.01]; theta0=[0.02 0.02]; ub=[10 10];
DV=[-1 -1; -1 0; -1 1; % DV_S=[shape FC];
    0 -1; 0 0; 0 1;
    1 -1; 1 0; 1 1];
OF_Sx_coef=Wx';
OF_Sy_coef=Wy';
OF_Sz_coef=Wz';
OF_Se_coef=We';
OF_Sm_coef=Wm';
% == Sx
[dmodel_Sx_coef1,perf]=dacefit(DV,OF_Sx_coef(:,1),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sx_coef2,perf]=dacefit(DV,OF_Sx_coef(:,2),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sx_coef3,perf]=dacefit(DV,OF_Sx_coef(:,3),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sx_coef4,perf]=dacefit(DV,OF_Sx_coef(:,4),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sx_coef5,perf]=dacefit(DV,OF_Sx_coef(:,5),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sx_coef6,perf]=dacefit(DV,OF_Sx_coef(:,6),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sx_coef7,perf]=dacefit(DV,OF_Sx_coef(:,7),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sx_coef8,perf]=dacefit(DV,OF_Sx_coef(:,8),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sx_coef9,perf]=dacefit(DV,OF_Sx_coef(:,9),@regpoly2,@corrgauss,th
eta0,lb,ub);

% == Sy
[dmodel_Sy_coef1,perf]=dacefit(DV,OF_Sy_coef(:,1),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sy_coef2,perf]=dacefit(DV,OF_Sy_coef(:,2),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sy_coef3,perf]=dacefit(DV,OF_Sy_coef(:,3),@regpoly2,@corrgauss,th
eta0,lb,ub);
```

```
[dmodel_Sy_coef4,perf]=dacefit(DV,OF_Sy_coef(:,4),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sy_coef5,perf]=dacefit(DV,OF_Sy_coef(:,5),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sy_coef6,perf]=dacefit(DV,OF_Sy_coef(:,6),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sy_coef7,perf]=dacefit(DV,OF_Sy_coef(:,7),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sy_coef8,perf]=dacefit(DV,OF_Sy_coef(:,8),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sy_coef9,perf]=dacefit(DV,OF_Sy_coef(:,9),@regpoly2,@corrgauss,th
eta0,lb,ub);
% == Sz
[dmodel_Sz_coef1,perf]=dacefit(DV,OF_Sz_coef(:,1),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sz_coef2,perf]=dacefit(DV,OF_Sz_coef(:,2),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sz_coef3,perf]=dacefit(DV,OF_Sz_coef(:,3),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sz_coef4,perf]=dacefit(DV,OF_Sz_coef(:,4),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sz_coef5,perf]=dacefit(DV,OF_Sz_coef(:,5),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sz_coef6,perf]=dacefit(DV,OF_Sz_coef(:,6),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sz_coef7,perf]=dacefit(DV,OF_Sz_coef(:,7),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sz_coef8,perf]=dacefit(DV,OF_Sz_coef(:,8),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sz_coef9,perf]=dacefit(DV,OF_Sz_coef(:,9),@regpoly2,@corrgauss,th
eta0,lb,ub);
% == Se
[dmodel_Se_coef1,perf]=dacefit(DV,OF_Se_coef(:,1),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Se_coef2,perf]=dacefit(DV,OF_Se_coef(:,2),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Se_coef3,perf]=dacefit(DV,OF_Se_coef(:,3),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Se_coef4,perf]=dacefit(DV,OF_Se_coef(:,4),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Se_coef5,perf]=dacefit(DV,OF_Se_coef(:,5),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Se_coef6,perf]=dacefit(DV,OF_Se_coef(:,6),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Se_coef7,perf]=dacefit(DV,OF_Se_coef(:,7),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Se_coef8,perf]=dacefit(DV,OF_Se_coef(:,8),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Se_coef9,perf]=dacefit(DV,OF_Se_coef(:,9),@regpoly2,@corrgauss,th
eta0,lb,ub);
% == Sm
[dmodel_Sm_coef1,perf]=dacefit(DV,OF_Sm_coef(:,1),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sm_coef2,perf]=dacefit(DV,OF_Sm_coef(:,2),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sm_coef3,perf]=dacefit(DV,OF_Sm_coef(:,3),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sm_coef4,perf]=dacefit(DV,OF_Sm_coef(:,4),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sm_coef5,perf]=dacefit(DV,OF_Sm_coef(:,5),@regpoly2,@corrgauss,th
eta0,lb,ub);
```

```
[dmodel_Sm_coef6,perf]=dacefit(DV,OF_Sm_coef(:,6),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sm_coef7,perf]=dacefit(DV,OF_Sm_coef(:,7),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sm_coef8,perf]=dacefit(DV,OF_Sm_coef(:,8),@regpoly2,@corrgauss,th
eta0,lb,ub);
[dmodel_Sm_coef9,perf]=dacefit(DV,OF_Sm_coef(:,9),@regpoly2,@corrgauss,th
eta0,lb,ub);
disp('kriging constructed');
% ==== save kriging model
clearvars perf Ux Uy Uz Ue Um Wx Wy Wz We Wm ...
    DV OF_Sx_coef OF_Sy_coef OF_Sz_coef OF_Se_coef OF_Sm_coef
save('D:\projects\Land
Optimization\Code\Optimization\krg_pod_coef_workspace.mat')
disp('kriging saved')
```

C. Compute Source Terms for New Design

```
% ============
% Constructe new source terms from kriging of POD coefs
clear all
close all
% == new design variables
taper_new=0;
BR_new=0.799;
% ==== load data: POD bases and POD-coef kriging model
% == POD bases
load('D:\projects\Land
Optimization\Code\Optimization\pod_workspace_shape_FC_3x3.mat',...
     'Ux','Uy','Uz','Ue','Um');
% == kriging
load('D:\projects\Land
Optimization\Code\Optimization\krg_pod_coef_workspace.mat',...
     '-regexp','dmodel...');
DV_new=[(taper_new-0)*2/(5-0)-1 (BR_new-0.7)*2/(1.3-0.7)-1];
% == POD coefficients
% Sx
OF_Sx_coef_new(1,1)=predictor(DV_new,dmodel_Sx_coef1);
OF_Sx_coef_new(1,2)=predictor(DV_new,dmodel_Sx_coef2);
OF_Sx_coef_new(1,3)=predictor(DV_new,dmodel_Sx_coef3);
OF_Sx_coef_new(1,4)=predictor(DV_new,dmodel_Sx_coef4);
OF_Sx_coef_new(1,5)=predictor(DV_new,dmodel_Sx_coef5);
OF_Sx_coef_new(1,6)=predictor(DV_new,dmodel_Sx_coef6);
OF_Sx_coef_new(1,7)=predictor(DV_new,dmodel_Sx_coef7);
OF_Sx_coef_new(1,8)=predictor(DV_new,dmodel_Sx_coef8);
OF_Sx_coef_new(1,9)=predictor(DV_new,dmodel_Sx_coef9);
% Sy
OF_Sy_coef_new(1,1)=predictor(DV_new,dmodel_Sy_coef1);
OF_Sy_coef_new(1,2)=predictor(DV_new,dmodel_Sy_coef2);
OF_Sy_coef_new(1,3)=predictor(DV_new,dmodel_Sy_coef3);
OF_Sy_coef_new(1,4)=predictor(DV_new,dmodel_Sy_coef4);
OF_Sy_coef_new(1,5)=predictor(DV_new,dmodel_Sy_coef5);
OF_Sy_coef_new(1,6)=predictor(DV_new,dmodel_Sy_coef6);
OF_Sy_coef_new(1,7)=predictor(DV_new,dmodel_Sy_coef7);
OF_Sy_coef_new(1,8)=predictor(DV_new,dmodel_Sy_coef8);
OF_Sy_coef_new(1,9)=predictor(DV_new,dmodel_Sy_coef9);
% Sz
OF_Sz_coef_new(1,1)=predictor(DV_new,dmodel_Sz_coef1);
OF_Sz_coef_new(1,2)=predictor(DV_new,dmodel_Sz_coef2);
OF_Sz_coef_new(1,3)=predictor(DV_new,dmodel_Sz_coef3);
OF_Sz_coef_new(1,4)=predictor(DV_new,dmodel_Sz_coef4);
OF_Sz_coef_new(1,5)=predictor(DV_new,dmodel_Sz_coef5);
```

```
OF_Sz_coef_new(1,6)=predictor(DV_new,dmodel_Sz_coef6);
OF_Sz_coef_new(1,7)=predictor(DV_new,dmodel_Sz_coef7);
OF_Sz_coef_new(1,8)=predictor(DV_new,dmodel_Sz_coef8);
OF_Sz_coef_new(1,9)=predictor(DV_new,dmodel_Sz_coef9);
% Se
OF_Se_coef_new(1,1)=predictor(DV_new,dmodel_Se_coef1);
OF_Se_coef_new(1,2)=predictor(DV_new,dmodel_Se_coef2);
OF_Se_coef_new(1,3)=predictor(DV_new,dmodel_Se_coef3);
OF_Se_coef_new(1,4)=predictor(DV_new,dmodel_Se_coef4);
OF_Se_coef_new(1,5)=predictor(DV_new,dmodel_Se_coef5);
OF_Se_coef_new(1,6)=predictor(DV_new,dmodel_Se_coef6);
OF_Se_coef_new(1,7)=predictor(DV_new,dmodel_Se_coef7);
OF_Se_coef_new(1,8)=predictor(DV_new,dmodel_Se_coef8);
OF_Se_coef_new(1,9)=predictor(DV_new,dmodel_Se_coef9);
% Sm
OF_Sm_coef_new(1,1)=predictor(DV_new,dmodel_Sm_coef1);
OF_Sm_coef_new(1,2)=predictor(DV_new,dmodel_Sm_coef2);
OF_Sm_coef_new(1,3)=predictor(DV_new,dmodel_Sm_coef3);
OF_Sm_coef_new(1,4)=predictor(DV_new,dmodel_Sm_coef4);
OF_Sm_coef_new(1,5)=predictor(DV_new,dmodel_Sm_coef5);
OF_Sm_coef_new(1,6)=predictor(DV_new,dmodel_Sm_coef6);
OF_Sm_coef_new(1,7)=predictor(DV_new,dmodel_Sm_coef7);
OF_Sm_coef_new(1,8)=predictor(DV_new,dmodel_Sm_coef8);
OF_Sm_coef_new(1,9)=predictor(DV_new,dmodel_Sm_coef9);
% == source terms;
Sx_new=Ux*OF_Sx_coef_new';
Sy_new=Uy*OF_Sy_coef_new';
Sz_new=Uz*OF_Sz_coef_new';
Se_new=Ue*OF_Se_coef_new';
Sm_new=Um*OF_Sm_coef_new';
disp('new source terms constructed')
% ==== write mew source terms to file (need pod workspace)
% == rearrange to cell index of stmsh6b if taper=0
Sx_uns=Sx_new;
Sy_uns=Sy_new;
Sz_uns=Sz_new;
Se_uns=Se_new;
Sm_uns=Sm_new;
source_w=[Sx_uns Sy_uns Sz_uns Se_uns Sm_uns];
disp('writing source terms...')
fp_write=['...\source term data\source_fluent.txt'];
fid=fopen(fp_write,'wt');
fprintf(fid,'%.9e\t%.9e\t%.9e\t%.9e\t%.9e\n',source_w');
fclose('all');
disp('source terms written')
```

B.2 Some Fluent UDF Codes

B.2.1 Time Averaging Instantaneous LES Solution

```
#include "udf.h"
/*time-average the instantaneous LES solution on the fly and store in
udm*/
DEFINE_ON_DEMAND(fluent_tave)
{
#if !RP_HOST
      Domain *d=Get_Domain(1);
      Thread *t;
      cell_t c;
```

```
	int i, cid;
	real tp;
	tp=N_TIME;
	thread_loop_c(t,d)
	{
		begin_c_loop(c,t)
		{
			C_UDMI(c,t,1)=((C_UDMI(c,t,1)*(tp-1))+C_U(c,t))/tp;
			C_UDMI(c,t,2)=((C_UDMI(c,t,2)*(tp-1))+C_V(c,t))/tp;
			C_UDMI(c,t,3)=((C_UDMI(c,t,3)*(tp-1))+C_W(c,t))/tp;
			C_UDMI(c,t,4)=((C_UDMI(c,t,4)*(tp-1))+C_P(c,t))/tp;
			C_UDMI(c,t,5)=((C_UDMI(c,t,5)*(tp-1))+C_T(c,t))/tp;
		C_UDMI(c,t,6)=((C_UDMI(c,t,6)*(tp-
		1))+C_P(c,t)+0.5*(SQR(C_U(c,t)+SQR(C_V(c,t))+SQR(C_W(c,t))))/t
		p;
		}
		end_c_loop(c,t)
	}
#endif
}
```

B.2.2 Source Term Correction Iteration

A. Source Term Correction

```
// suitable for both serial and parallel Fluent
#define Cp 1006.43
#define Tref 0
#define Ur 12
/*For source term correction iteration*/
/*correct the source term based on previous source term iteration*/
DEFINE_ON_DEMAND(correct_source)
{
#if !RP_HOST
	Domain *d=Get_Domain(1);
	Thread *t;
	cell_t c;
	int i, cid;
	real E;
	real phi_x=0.8,phi_y=0.8,phi_z=0.8,phi_m=0.01,phi_e=1;	/*
relaxation factor for each source correction */
	thread_loop_c(t,d)
	{
		begin_c_loop(c,t)
		{
	 E=Cp*(C_T(c,t)-
	 Tref)+0.5*(SQR(C_U(c,t))+SQR(C_V(c,t))+SQR(C_W(c,t)));
	 C_UDMI(c,t,9)+=(C_R(c,t)*C_UDMI(c,t,1)-
	 C_R(c,t)*C_U(c,t))/C_UDMI(c,t,7)*phi_x;
	 C_UDMI(c,t,10)+=(C_R(c,t)*C_UDMI(c,t,2)-
	 C_R(c,t)*C_V(c,t))/C_UDMI(c,t,7)*phi_y;
	 C_UDMI(c,t,11)+=(C_R(c,t)*C_UDMI(c,t,3)-
	 C_R(c,t)*C_W(c,t))/C_UDMI(c,t,7)*phi_z;
	 C_UDMI(c,t,12)+=(C_R(c,t)*C_UDMI(c,t,6)-
	 C_R(c,t)*E)/C_UDMI(c,t,7)*phi_e;
	 C_UDMI(c,t,13)+=(C_UDMI(c,t,4)-
	 C_P(c,t))/(C_UDMI(c,t,7)*SQR(Ur))*phi_m;
		}
		end_c_loop(c,t)
	}
```

```
      #if RP_NODE
    if (I_AM_NODE_ZERO_P)
    {
            Message("\nsource corrected\n");
    }
      #endif
      #if !PARALLEL
      Message("\nsource corrected\n");
      #endif
#endif
}
```

B. Compute RMS Solution Change

```
// RMS solution change to decide convergence of source term iteraion
#include "udf.h"
#define Ncell 3583800
#define Cp 1006.43
#define Tref 0
#define Ur 12
DEFINE_EXECUTE_AT_END(timestep_udm_custom_residual)
{
      real Rx,Ry,Rz,Rm,Re;
      real Rx_rms=0,Ry_rms=0,Rz_rms=0,Rm_rms=0,Re_rms=0;
      real E;
#if RP_NODE
      Domain *d=Get_Domain(1);
      Thread *t=NULL;
      cell_t c=-1;
#endif
#if RP_HOST
      Domain *d=Get_Domain(1);
      FILE *fp1=NULL,*fp2=NULL,*fp3=NULL,*fp4=NULL,*fp5=NULL;
      char fn1[200],fn2[200],fn3[200],fn4[200],fn5[200];
#endif
#if !PARALLEL
      Domain *d=Get_Domain(1);
      Thread *t=NULL;
      cell_t c=-1;
      FILE *fp1=NULL,*fp2=NULL,*fp3=NULL,*fp4=NULL,*fp5=NULL;
      char fn1[200],fn2[200],fn3[200],fn4[200],fn5[200];
#endif
      /* get local pseudo time-step */
      Cell_Function_Values(d,"time-step");
#if !RP_HOST
      thread_loop_c(t,d)
      {
            begin_c_loop(c,t)
            {
                  C_UDMI(c,t,7)=C_TMP(c,t);
            }
            end_c_loop(c,t)
      }

      #if RP_NODE
    if (I_AM_NODE_ZERO_P)
    {
        Message("time-step passed to udm\n");
    }
      #endif
      #if !PARALLEL
      Message("time-step passed to udm\n");
```

```
      #endif
#endif
#if !RP_HOST
      thread_loop_c(t,d)
      {
            begin_c_loop_int(c,t)
            {
       E=Cp*(C_T(c,t)-
       Tref)+0.5*(SQR(C_U(c,t))+SQR(C_V(c,t))+SQR(C_W(c,t)));
       Rx=(C_R(c,t)*C_UDMI(c,t,1)-C_R(c,t)*C_U(c,t))/C_UDMI(c,t,7);
            Ry=(C_R(c,t)*C_UDMI(c,t,2)-C_R(c,t)*C_V(c,t))/C_UDMI(c,t,7);
            Rz=(C_R(c,t)*C_UDMI(c,t,3)-C_R(c,t)*C_W(c,t))/C_UDMI(c,t,7);
            Re=(C_R(c,t)*C_UDMI(c,t,6)-C_R(c,t)*E)/C_UDMI(c,t,7);
            Rm=(C_UDMI(c,t,4)-C_P(c,t))/(C_UDMI(c,t,7)*SQR(Ur));
            Rx_rms+=SQR(Rx);
            Ry_rms+=SQR(Ry);
            Rz_rms+=SQR(Rz);
            Re_rms+=SQR(Re);
            Rm_rms+=SQR(Rm);
            }
            end_c_loop_int(c,t)
      }
#endif
#if RP_NODE
      Rx_rms=PRF_GRSUM1(Rx_rms);
      Ry_rms=PRF_GRSUM1(Ry_rms);
      Rz_rms=PRF_GRSUM1(Rz_rms);
      Re_rms=PRF_GRSUM1(Re_rms);
      Rm_rms=PRF_GRSUM1(Rm_rms);
#endif
      node_to_host_real_1(Rx_rms);
      node_to_host_real_1(Ry_rms);
      node_to_host_real_1(Rz_rms);
      node_to_host_real_1(Re_rms);
      node_to_host_real_1(Rm_rms);
#if !RP_NODE
      Rx_rms=sqrt(Rx_rms/Ncell);
      Ry_rms=sqrt(Ry_rms/Ncell);
      Rz_rms=sqrt(Rz_rms/Ncell);
      Re_rms=sqrt(Re_rms/Ncell);
      Rm_rms=sqrt(Rm_rms/Ncell);
      strcpy(fn1,"...\\custom_residuals\\Rx.txt");
      fp1=fopen(fn1,"a");
      fprintf(fp1,"%e\n",Rx_rms);
      fclose(fp1);
      strcpy(fn2," ...\\custom_residuals\\Ry.txt");
      fp2=fopen(fn2,"a");
      fprintf(fp2,"%e\n",Ry_rms);
      fclose(fp2);
      strcpy(fn3," ...\\custom_residuals\\Rz.txt");
      fp3=fopen(fn3,"a");
      fprintf(fp3,"%e\n",Rz_rms);
      fclose(fp3);
      strcpy(fn4," ...\\custom_residuals\\Rm.txt");
      fp4=fopen(fn4,"a");
      fprintf(fp4,"%e\n",Rm_rms);
      fclose(fp4);
      strcpy(fn5," ...\\custom_residuals\\Re.txt");
      fp5=fopen(fn5,"a");
      fprintf(fp5,"%e\n",Re_rms);
      fclose(fp5);
```

```
      Message("Rx_rms\t\tRy_rms\t\tRz_rms\t\tRe_rms\t\tRm_rms\n");
      Message("%e\t%e\t%e\t%e\t%e\n",Rx_rms,Ry_rms,Rz_rms,Re_rms,Rm_rms);
#endif
}
```

B.2.3 Load Source Term Data to the Solver

```
#include "udf.h"
#define Ncell 3583800
#define Ur 12
#define Cp 1006.43
#define Tref 0
#define rho 1.225
DEFINE_ON_DEMAND(load_sources)
{
 #if !RP_HOST
      FILE *fid;
      Domain *d=Get_Domain(1);
      Thread *t;
      cell_t c;
      int i, cid;
      float *Sx,*Sy,*Sz,*Se,*Sm;
      char filename[200], t_str[10];
      strcpy(filename,"...\\source term data \\source_fluent.txt");
      fid=fopen(filename, "r");
      /*allocate memory to store the data*/
      Sx = (float *)malloc(Ncell * sizeof(float));
      Sy = (float *)malloc(Ncell * sizeof(float));
      Sz = (float *)malloc(Ncell * sizeof(float));
      Se = (float *)malloc(Ncell * sizeof(float));
      Sm = (float *)malloc(Ncell * sizeof(float));
      for (i=0; i<Ncell; i++)
      {
            fscanf(fid,"%e%e%e%e%e",&Sx[i],&Sy[i],&Sz[i],&Se[i],&Sm[i]);
      }
      fclose(fid);
      /* store the gradients to UDM 4-12 */
      thread_loop_c(t,d)
      {
            begin_c_loop(c,t)
            {
                  cid=C_UDMI(c,t,0);
                  C_UDMI(c,t,9)=Sx[cid];
                  C_UDMI(c,t,10)=Sy[cid];
                  C_UDMI(c,t,11)=Sz[cid];
                  C_UDMI(c,t,12)=Se[cid];
                  C_UDMI(c,t,13)=Sm[cid];
            }
            end_c_loop(c,t)
      }
      free(Sx);
      free(Sy);
      free(Sz);
      free(Se);
      free(Sm);
      #if RP_NODE
    if (I_AM_NODE_ZERO_P)
    {
            Message("sources loaded\n");
    }
      #endif
```

```
      #if !PARALLEL
      Message("sources loaded\n");
      #endif
 #endif
}
```

B.2.4 Link Source Term Data to Flow Equations

```
#include "udf.h"
DEFINE_SOURCE(mass_source, c, t, dS, eqn)
{
      real Sm;
      dS[eqn]=0.0;
      Sm=C_UDMI(c,t,13);
      return Sm;
}
DEFINE_SOURCE(xmom_source, c, t, dS, eqn)
{
      real Sx;
      dS[eqn]=0.0;
      Sx=C_UDMI(c,t,9);
      return Sx;
}
DEFINE_SOURCE(ymom_source, c, t, dS, eqn)
{
      real Sy;
      dS[eqn]=0.0;
      Sy=C_UDMI(c,t,10);
      return Sy;
}
DEFINE_SOURCE(zmom_source, c, t, dS, eqn)
{
      real Sz;
      dS[eqn]=0.0;
      Sz=C_UDMI(c,t,11);
      return Sz;
}
DEFINE_SOURCE(energy_source, c, t, dS, eqn)
{
      real Se;
      dS[eqn]=0.0;
      Se=C_UDMI(c,t,12);
      return Se;
}
```

B.2.5 Load and Initialize Flow with Time-Mean LES Solution Q^*

A. Load Time-Mean LES Solution

```
#include "udf.h"
#define Ncell 3583800
#define Ur 12
#define Cp 1006.43
#define Tref 0
#define rho 1.225
DEFINE_ON_DEMAND(load_tave_solution)
{
#if !RP_HOST
      FILE *fid;
      Domain *d=Get_Domain(1);
```

```
      Thread *t;
      cell_t c;
      int i, cid;
    float *u_t=NULL,*v_t=NULL,*w_t=NULL,*p_t=NULL,*T_t=NULL,*pt_t=NULL;
      char filename[200], t_str[10];
      real E;
      strcpy(filename,"...\\tave solution\\tave_solution.txt");
      fid=fopen(filename, "r");
      /* allocate memory to store the data */
      u_t = (float *)malloc(Ncell * sizeof(float));
      v_t = (float *)malloc(Ncell * sizeof(float));
      w_t = (float *)malloc(Ncell * sizeof(float));
      p_t = (float *)malloc(Ncell * sizeof(float));
      T_t = (float *)malloc(Ncell * sizeof(float));
      pt_t = (float *)malloc(Ncell * sizeof(float));
      for (i=0; i<Ncell; i++)
      {
      fscanf(fid,"%e%e%e%e%e%e",&u_t[i],&v_t[i],&w_t[i],&p_t[i],&T_t[i],&
pt_t[i]);
      }
      fclose(fid);
      thread_loop_c(t,d)
      {
            begin_c_loop(c,t)
            {
                  cid=C_UDMI(c,t,0);
                  C_UDMI(c,t,1)=u_t[cid];
                  C_UDMI(c,t,2)=v_t[cid];
                  C_UDMI(c,t,3)=w_t[cid];
                  C_UDMI(c,t,4)=p_t[cid];
                  C_UDMI(c,t,5)=T_t[cid];
          E=Cp*(T_t[cid]-
          Tref)+0.5*(SQR(u_t[cid])+SQR(v_t[cid])+SQR(w_t[cid]));
                  C_UDMI(c,t,6)=E;
                  C_UDMI(c,t,7)=1.0;
                  C_UDMI(c,t,22)=pt_t[cid];
            }
            end_c_loop(c,t)
      }
      #if RP_NODE
    if (I_AM_NODE_ZERO_P)
    {
            Message("tave solution loaded\n");
    }
      #endif
      #if !PARALLEL
      Message("tave solution loaded\n");
      #endif
      free(u_t);
      free(v_t);
      free(w_t);
      free(p_t);
      free(T_t);
      free(pt_t);
#endif
}
```

B. Initialize Flow with Q^* (Fluent Scheme Code)

```
(ti-menu-load-string (format #f "/solve/patch fluid () x-velocity yes yes
udm1"))
```

```
(ti-menu-load-string (format #f "/solve/patch fluid () y-velocity yes yes
udm2"))
(ti-menu-load-string (format #f "/solve/patch fluid () z-velocity yes yes
udm3"))
(ti-menu-load-string (format #f "/solve/patch fluid () pressure yes
udm4"))
(ti-menu-load-string (format #f "/solve/patch fluid () temperature yes
udm5"))
```