High-Fidelity Gradient-Free Optimization of Low Pressure Turbine Cascades

Anthony Aubry

A Thesis

In the Department

of

Mechanical, Industrial, and Aerospace Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of

Master of Applied Sciences (Mechanical Engineering) at

Concordia University

Montreal, Quebec, Canada

August 2021

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to verify that thesis prepared

By: **Anthony Aubry**

Entitled: **High-Fidelity Gradient-Free Optimization of Low Pressure Turbine Cascades**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Mechanical Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final Examining Committee:

______________________________ Chair

Dr. Carole El Ayoubi

______________________________ Internal Examiner

Dr. Carole El Ayoubi

______________________________ External Examiner

Dr. Luis Rodrigues

______________________________ Supervisor

Dr. Brian Vermeire

Approved by: ____

Chair of Department or Graduate Program Director, Dr. Sivakumar Narayanswamy

________ 2021 ____

Dean of Faculty, Dr. Mourad Debbabi

Abstract

High-Fidelity Gradient-Free Optimization of Low Pressure Turbine Cascades

Anthony Aubry

This study explores aerodynamic shape optimization of Low Pressure Turbine (LPT) blades using Implicit Large Eddy Simulation (ILES) coupled with the Mesh Adaptive Direct Search (MADS) optimization algorithm. Aerodynamic shape optimization in the aerospace industry relies heavily on the adjoint method combined with steady-state Computational Fluid Dynamics (CFD) solvers, namely the Reynolds-Averaged Navier-Stokes (RANS) approach. While this methodology has proven quite useful and efficient for common optimization problem, its effectiveness is reduced when faced with separated turbulent flow. As LPT designs move increasingly towards high lift configurations to reduce engine weight, accurately predicting transition and turbulent separation for each optimization cycles becomes a necessity. The T106D blade with an exit Mach number of 0.4 and Reynolds number 80,000 is used as the initial conditions before optimization, because of the presence of turbulent transition and separation on the Suction Surface (SS). First, the ILES framework employing the Flux Reconstruction (FR) approach is presented along with the benefits of its implementation on modern many-core hardware, specifically Graphical Processing Units (GPUs). The formulation of the MADS algorithm is then presented and compared against gradient-based methods and other gradient-free methods. A Bézier curve is used to modify the original blade camber line, with the use of 4 control points, permitting a wide range of shapes to be obtained. Then, the results of the baseline case are presented with a comparison with experimental results to validate the methodology selected. The results of optimization cycles using two different objective functions are then presented and compared with the original blade. Results demonstrate that a total pressure loss coefficient reduction of about 16% was achieved in the first optimization, while a tangential force increase of more than 29% was achieved with the second optimization. Finally, recommendations are made as to how this methodology could be successfully applied in industrial applications, and what future research should focus on.

Acknowledgements

I would like to express my gratitude to my parents, for their constant love, support and guidance throughout my journey. Without their efforts and sacrifice, I would never have been able to pursue this degree and accomplish as much in all spheres of my life.

I would also like to thank my siblings and my girlfriend, who have always supported and encouraged me throughout my academic and athletic careers.

I am very thankful for the lab mates that have helped me along the way, especially Carlos and Mohsen for showing me the basics early on, and Hamid for guiding and helping me with everything regarding optimization.

Lastly, I would like to extend a very special thank you to my supervisor, Dr. Brian Vermeire, whose knowledge, guidance and continuous support have helped me tremendously throughout my research.

Contents

List of Figures

List of Tables

Acronyms

ACARE	Advisory Council for Aeronautics Research in Europe
CFD	Computational Fluid Dynamics
CFEM	Continuous Finite Element Method
CPR	Correction Procedure via Reconstruction
CPU	Core Processing Unit
DFEM	Discontinuous Finite Element Method
DG	Discontinuous Galerkin
DNS	Direct Numerical Simulation
FLOP	Floating-Point Operations
FR	Flux Reconstruction
FDM	Finite Difference Method
FEM	Finite Element Method
FVM	Finite Volume Method
GA	Genetic Algorithm
GPS	Generalized Pattern Search
GPU	Graphical Processing Unit
HP	High Pressure
HPC	High Pressure Compressor
HPT	High Pressure Turbine
ICAO	International Civil Aviation Organization
ILES	Implicit Large Eddy Simulation
LCP	Lifting Collocation Penalty
LDG	Local Discontinuous Galerkin
LE	Leading Edge

LES	Large Eddy Simulation
LP	Low Pressure
LPC	Low Pressure Compressor
LPT	Low Pressure Turbine
MADS	Mesh Adaptive Direct Search
NASA	National Aeronautics and Space Administration
NSE	Navier-Stokes Equations
PDE	Partial Differential Equation
PS	Pressure Surface
PSO	Particle Swarm Optimization
PyFR	Python Flux Reconstruction
RANS	Reynolds-Averaged Navier-Stokes
RK	Runge Kutta
SD	Spectral Difference
SGS	Sub-Grid Scale
SS	Suction Surface
SV	Spectral Volume
TE	Trailing Edge

Nomenclature

English Letters

$1D$	One-Dimensional
$2D$	Two-Dimensional
$3D$	Three-Dimensional
$a_{i,j,f,l}$	Scheme lifting coefficient at point *l* along boundary *f*
c	chord length
c_p	Specific heat coefficient at constant pressure
C_p	Pressure coefficient
c_v	Specific heat coefficient at constant volume
DS	Design Space
e	Internal energy per unit mass
E	Total energy
f_e	Body forces
f_t	Tangential force
F	Flux vector
$\mathcal{F}$	Physical boundary location function
$[\tilde{F}]$	Flux difference polynomial along boundaries
g	Flux correction function
h	Airfoil span
H	Total enthalpy
H^k	House-holder matrix at iteration *k*
I	Identity matrix
$\bar{\bar{I}}$	Unit tensor
$\mathcal{I}$	Objective function

k	Thermal conduction coefficient
l_i	Physical element size
L	Characteristic length
Ma	Mach number
M^k	Mesh points at iteration k
p	Static pressure
p_0	Stagnation pressure
$\mathbb{P}$	Polynomial degree
$\mathcal{P}^k$	Trial points at iteration k
Pr	Prandtl number
q_H	Heat sources
$\mathcal{R}$	Governing equation
R	Semi-discrete space operator
Ra	Radau polynomial
Re	Reynolds number
s_d	Sample standard deviation
S	Blade pitch
$S(u,t)$	Source term
S_f	Element surface
S^k	Optimal point at iteration k
t	Time
T	Static temperature
T_0	Stagnation temperature
T^k	Trial points evaluated before iteration k
u	Conserved variable
U	Vector of conserved variables
v	Velocity magnitude
V	Velocity vector
w	Random normalized vector
W_f	Work of external volume forces
x	Physical horizontal coordinate
X	Vector of design variables

y	Physical vertical coordinate
Y	Total pressure loss coefficient

Greek Letters

α	Penalty weight factor
β_1	Inlet flow angle
β_2	Outlet flow angle
β_3	Stagger angle
Δ	Grid spacing
δ	Increment
δ_{ij}	Kronecker delta
ϵ	Energy dissipation rate
η	Kolmogorov length scale
ℓ	Largest eddy scale
λ	Viscosity coefficient
ρ	Density
ϕ	Nodal basis function
ψ	Lagrange multiplier
Ψ	Flux correction field
$\overline{\overline{\sigma}}$	Total internal stress tensor
$\overline{\overline{\tau}}$	Viscous shear stress tensor
τ_{ij}	Reynolds stress
μ	Dynamic viscosity
ν	Kinematic viscosity
Ω	Computational domain
γ	Ratio of specific heats
Γ	Mapping function
ξ	Reference horizontal coordinate
ζ	Reference vertical coordinate

Operators

∇	Divergence operator $\nabla = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right]$

$\bigcup$	Union of sets
$\bigcap$	Intersection of sets
$\bigoplus$	Direct sum
$\sum$	Summation
$\prod$	Product
$\overline{\bullet}$	Average component
$\bullet'$	Fluctuating component
$\otimes$	Tensor product

Subscripts

ax	Axial
C	Convective
D	Diffusive
∞	Undisturbed
inv	Inviscid
L	Left interface
m	Mesh
p	Poll
R	Right interface
vis	Viscous

Superscripts

C	Continuous
h	Piece-wise continuous approximation
k	Iteration number
T	Transpose

Chapter 1

Introduction

The passenger aircraft industry has grown tremendously over the past several decades, and this trend is set to continue in the future. With this growing demand, and a global focus shifted towards reducing the environmental impacts of transportation, new technologies and tools are needed to design the next generation of aircraft. One of the main environmental impacts of air transportation is the generation of high quantities of CO_2 emitted from gas turbine engines. To tackle this issue, the Advisory Council for Aeronautics Research in Europe (ACARE) set a target reduction of 75% of CO_2 emitted per passenger per kilometer traveled to be achieved before 2030 [3]. In order to reach this goal, aircraft components must become lighter and the fuel consumption of engines must decrease.

Some of the most critical components of gas turbine engines, in terms of fuel consumption, are the compressor and turbine stages. Both categories of turbomachinery have benefited immensely from advances in Computational Fluid Dynamics (CFD) and computing power to numerically solve for the flow around complex shapes, instead of relying solely on wind tunnel testing. Although some experimental tests are still required to validate a design, CFD tools have significantly reduced the cost and time associated with the aerodynamic design of compressor and turbine cascades.

Traditionally, the aerodynamic design process for these components was focused on increasing their individual performance or efficiency. As CFD tools became increasingly efficient, achieving incremental gains in performance became much more costly. Following this, the industry's focus has shifted towards reducing the total impact of engine operation instead of being driven only by single component performance. An example of that is how the design of Low Pressure Turbine (LPT) stages are now focused on reducing the number of blades instead of increasing the isentropic efficiency of the stage. Reducing the number of components decreases the cost of engine maintenance and the overall weight of the engine. Weight reductions directly affect the overall performance of an aircraft by either alleviating the forces acting on it or by permitting more passengers to be flown by the same quantity of fuel. Hence, the next generation of CFD tools must allow designers to get accurate solutions for more complex flows around these few highly-loaded blades.

Currently, aerodynamic design using CFD relies heavily on low-order methods paired with Reynolds-Averaged Navier-Stokes (RANS) solvers to model the flow field in a steady-state manner. These industrial solvers offer satisfying results over a wide range of flow conditions at low computational cost, but often fail to accurately model complex turbulent flows with regions of separation. NASA's 2030 vision study [4] highlights this issue as one of the main factors limiting the use of RANS solvers in the aerospace design process, and urges the need to develop a new family of CFD software well-suited for scale-resolving simulations on modern many-core hardware architectures. Scale-resolving simulations, including Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS), capture the unsteady flow field, enabling them to resolve the true turbulent behavior.

With the need for LES growing, and higher levels of accuracy being desired, high-order schemes, defined as third-order accurate or more, are being developed. Some of the schemes currently used to perform high-order LES are the Discontinuous Galerkin (DG) [5, 6, 7], Spectral Volume (SV) [8] and Spectral Difference (SD) [9] methods. These schemes allow for much lower levels of numerical error, compared to the traditional low-order methods used with RANS solvers.

While high-order LES has distinct advantages over industry-standard solvers, its utility in aerodynamic shape optimization remains unexplored. Many roadblocks have prevented scale-resolving simulations to be used in the aerodynamic design process, but advances in computing and new CFD tools have solved some of them and now allow researchers to explore this application.

1.1 Aviation Trends

The International Civil Aviation Organization (ICAO) generates a yearly environmental report detailing aviation trends and their impact on the next generation of aircraft and the environment. The 2019 version [10] of this report highlights recent trends in aerodynamics and propulsion that have shown promising results while also showing growth opportunities. The three main trends in aircraft propulsion systems are the increase in thermal efficiency, increase in propulsive efficiency, and decrease of engine weight and drag.

The increase in thermal efficiency trend focuses on augmenting the peak temperature in the cycle, without increasing the amount of fuel burned. This is done by increasing the overall pressure ratio of the compressor, which is limited by structural weight and the availability of materials and cooling techniques able to sustain higher peak operating temperatures. This trend is mainly driven by research done on materials and cooling techniques.

Increasing the propulsive efficiency of a gas turbine engine has been recently done by augmenting the bypass ratio of turbofan engines. This is done by enlarging the fan diameter and finding new ways to drive it at the proper angular velocity, like the geared turbofan technology. This trend is mainly driven by structural advancements to integrate a slowly rotating large fan to the rest of the engine.

The trend on reducing the weight and drag produced by the engine and its nacelle is more general and can be achieved in a variety of ways. Reducing the size of the nacelle and engine can reduce the drag it generates, while reducing the number of components or their individual weight can help decrease the engine weight. This trend can be achieved by better aerodynamic design of components, new materials and a multitude of other engineering disciplines.

While aviation trends are driven by the environmental impact of airplanes, economical profitability remains a key aspect of aircraft design. As the fuel consumption of airplanes decreases and the aviation market grows, the total cost of engine ownership becomes dominated by factors other than fuel costs. From 2016 to 2017, even though the price of fuel rose and air traffic increased, overall fuel costs actually dropped [11]. The cost of fuel only accounted for around 20% of total airline expenses, which means that other costs such as labor and maintenance have become more prominent. Such a trend signifies that aircraft manufacturers must now take into account more than just the performance and initial cost of the aircraft, in order for it to remain profitable.

One of the ways to affect fuel consumption and profitability positively is by reducing the size or the number of components in a gas turbine engine, while keeping at least the same performance. Engine overhauls are costly operations in terms of money and time, so reducing the number of components would effectively reduce these costs, while also decreasing fuel consumption.

The functioning of a typical gas turbine engine and its components will now be explored.

1.2 Gas Turbine Engine

Currently, commercial aircraft are powered by gas turbine engines similar to the one shown in Figure 1.1 [1]. The five main components of such an engine are the intake, compressors, combustion chamber, turbines and nozzle. It should be noted that for commercial aircraft, the first compressor is a large fan, after which the bypass flow is introduced. The flow exiting the fan either goes in the core of the engine,

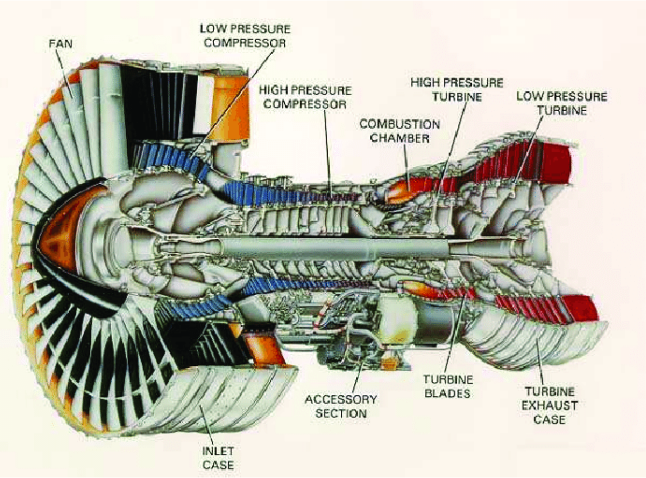


Figure 1.1. Turbofan engine layout [1]

where all of the other engine components are, or is bypassed directly to a nozzle to generate the majority of the engine's thrust. The following description focuses on the flow going through the core of the engine, where the energy is extracted to drive the fan and other components and systems.

The airflow enters the engine from the intake, which diffuses the flow before it reaches the first compressor stage. Multiple stages, where one stage is composed of one rotor and one stator, of compressors are then needed to bring the air pressure high enough without generating too much turbulence and energy loss. Then, the highly pressurized air is mixed with fuel and ignited to raise the energy level of the flow in the combustion chamber. Multiple turbine stages, each comprised of one stator and one rotor, are subsequently used to extract energy from the gas flow to drive the compressors and generate electricity for the rest of the aircraft. The last section is a nozzle which accelerates the flow to generate thrust to propel the aircraft forward.

Since the gas properties change drastically in the turbomachinery sections, it is usually separated into two distinct shafts which rotate at vastly different speeds. The High Pressure (HP) shaft includes the HP Compressor (HPC) and the HP Turbine (HPT) which are located in close proximity to the combustion chamber. The turbomachinery on the HP shaft deals with very high temperatures and densities, especially the HPT, which implies that their performance is heavily limited by materials and cooling techniques. The Low Pressure (LP) shaft includes the fan, LP Compressor (LPC) and the LPT, which are located closer to the intake and nozzle, respectively. With the temperatures and densities being relatively low in the LP sections, their performance is driven mainly by the aerodynamics of the gas flow. Whereas the HP shaft requires small components to act on a highly compressed flow, the components on the LP shaft are much bigger because they interact with large volumes of relatively low density gas.

For all stages of compressors and turbines, the flow is highly complex and unsteady. The complexity arises from a variety of factors, such as interactions between stators and rotors, different pressure gradients acting on the flow, turbulent separation, shocks, heat transfer and many more. Although some of these factors can be avoided or mitigated, the majority must be accounted for during the aerodynamic design process to ensure the validity of the designed airfoils and stages once placed in the complete engine. Each section of the engine is affected differently by these factors and design tools must be tailored for each section. For example, compressor flow is highly affected by shocks, but HPT flow is not and is instead much more affected by the presence of secondary flows caused by the high flow turning and low airfoil aspect ratios [12].

Hence, before beginning aerodynamic shape design, it is important to understand all the major fluid interactions that occur in the engine section in question and make sure that the proposed CFD tools are tailored to resolve these specific interactions efficiently.

The LPT section will now be expanded upon, since it is of particular interest for weight reduction purposes.

1.2.1 LPT Implications

A turbine stage is composed of one row of stationary vanes followed by a row of rotating blades. The role of the vanes is to guide the incoming flow towards the tangential direction of rotation of the blades following it. Then, the rotor blades are responsible to turn the flow to generated a net torque on the shaft.

The LPT is located after the HPT and before the nozzle and is composed of the last few stages of turbomachinery in a gas turbine engine. Since the LPT is the furthest set of turbines from the combustion chamber and compressors, the flow entering it is at relatively low temperatures and pressures. In order to extract a large amount of energy from these lower gas properties, the hub diameter of the stages and the vanes and blades must be much larger than for the HPT. For these reasons, it is estimated that the LPT alone is responsible for 20% to 33% of the total engine weight [13, 14]. Hence, the current environmental focus of aviation has pushed LPT designs heavily towards reducing weight while achieving similar or better levels of efficiency.

Since the majority of each flight is spent in higher altitude cruise conditions, the incoming air flow has a low density, which results in relatively low Reynolds number flow in the LPT. Fluid flows at low Reynolds number over blades are initially laminar and experience separation and transition to turbulence as they progress over a surface. Separation and turbulent transition are highly unsteady phenomena that are hard to predict and model, while also having a large impact on the efficiency of a LPT stage [15]. Because of their respective function in the stage, the rotor blades are much more likely to be affected by separation and transition than the vanes.

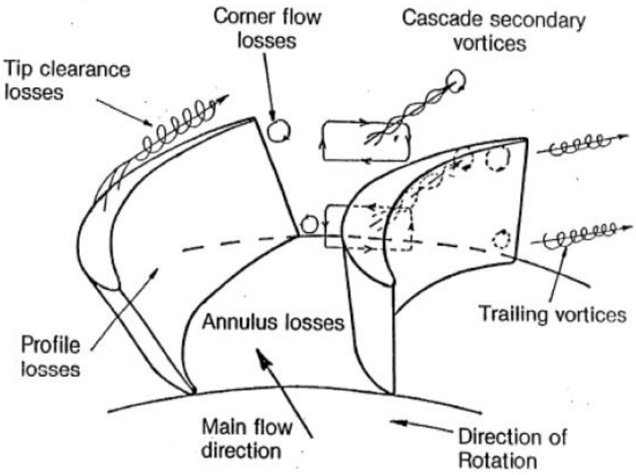


Figure 1.2. Turbine aerodynamic losses

To quantify the efficiency of a stage or a row of vanes or blades, multiple sources of losses are combined into a total pressure loss coefficient, which is then used to calculate how close the stage or airfoil is to isentropic operation. This coefficient is composed of different sources of losses, namely profile, secondary and tip clearance, which are shown in Figure 1.2 [16]. Profile losses include the effects of the boundary layer growth on the airfoil profile, the expansion and diffusion of the flow and the Trailing Edge (TE). Secondary losses include annulus losses caused by the casing and hub boundary layer growth effects, while also containing the effects of the pressure gradient created by adjacent airfoils. Tip clearance losses only occur for rotor blades, where a small portion of the total annulus height has no blade to guide the flow, leading to unwanted flow patterns [17]. Profile losses are mostly a two-dimensional (2D) phenomena stemming from the cross-sectional shape of the airfoil at different span, whereas secondary and tip losses are largely three-dimensional (3D) and caused by the geometry of the hub, full blade and casing. Depending on the stage geometry, one of these loss mechanisms might be dominant and the designer must select an optimization methodology suited to resolve this main mechanism.

As the LPT is composed of high aspect ratio airfoils, where aspect ratio is defined as the airfoil span h divided by its chord length c, the losses incurred by the annulus surfaces and the tip clearance only affect a small portion of the span. Hence, the profile losses and the pressure gradients between airfoils at mid-span account for almost 80% of all the losses occurring in the LPT [13]. It follows that the CFD tools used to simulate the flow around LPT blades must be capable of accurately capturing profile losses caused by separation and turbulent transition effects, in order to correctly predict the LPT efficiency.

A general turbine design process will now be detailed in the following section to expand on the role of CFD tools.

1.2.2 Turbine Design

The traditional aerodynamic design process of a gas turbine engine can be divided into four main steps, namely cycle calculations, preliminary design, through flow design and airfoil design [16, 18]. The cycle calculations take into account the geometry of the engine (bypass ratio, number of shafts, number of compressor and turbine stages, etc.) and the flight conditions in cruise (altitude, aircraft speed, etc.) to define gas properties at the different stages inside the engine and the overall predicted performance. Most of the properties are found using benchmarking and projecting the desired efficiency of different components prior to designing them. From the gas properties, each stage of the compressor and turbine go through a preliminary design that will estimate the required vane and blade geometries, flow and metal angles, rotational velocity and more. This design phase is based on previous experimental data and benchmarking and is focused on the mean line properties of each stage. Then, the through flow design phase will add considerations for the effects of the flow at the hub and tip of the stages to generate more realistic performance estimates. The last design phase is where CFD tools are used to simulate the flow around 2D and 3D airfoil geometries and full stages to find and optimize their precise performance.

Once the design reaches the numerical simulation stages, many parameters are already set and cannot be changed during the shape optimization of an airfoil geometry. Some of those parameters are inlet and outlet pressures, temperatures, velocities and flow angles. During preliminary design, these parameters are set by the neighboring airfoil rows and changing them would imply redesigning multiple components. For the LPT, the outflow angle constraint is less stringent, especially for the rotor of the last stage, but all constraints should preferably be respected.

In order to understand how CFD tools can help in the aerodynamic design process, the discretization process and its effects on solution accuracy are presented next.

1.3 Discretization

A general fluid flow is governed by the Navier-Stokes Equations (NSE), for which the complete mathematical formulation will be presented in Chapter 2. Since no general analytical solutions to the NSE exist, CFD must employ discretizations to find approximate ones. The discretization process consists of converting a continuous solution into a discrete approximation of the solution at a finite number of grid points or elements. This process is often used to approximate the solution of Partial Differential Equations (PDEs), which are the main components of the NSE. To generate an accurate solution, many discretization methods can be employed and the following sections will go over the spatial and temporal discretization processes and the methods typically used.

1.3.1 Spatial Discretization

There are three distinct spatial discretization methods used in CFD, which are the Finite Difference Method (FDM), Finite Volume Method (FVM), and Finite Element Method (FEM).

In the FDM, a finite number of points are used to represent the solution throughout the domain. No assumptions are made as to the behaviour of the solution between grid points and the points must be equally spaced. Hence, the FDM is difficult to implement on a domain containing non-rectangular boundaries, which would require an unstructured mesh. The solution accuracy can be controlled by refining the grid or by increasing the number of solution points taken into account to define each approximation.

On the other hand, the FVM separates the domain into volumes or cells, inside of which the solution is constant. It creates a piecewise constant approximation of the solution, with discontinuities at each cell interfaces. The use of volumes enables the implementation on unstructured grids, with cells of different sizes, to resolve the flow solution near round and non-rectangular shapes. The solution accuracy can be controlled by refinements of the grid, or extending the reconstruction stencil.

As for the FEM, it combines both approaches by representing the domain as volumes each containing points to represent the solution. The number of points inside each element creates a polynomial representation of the solution, using basis functions, which can more accurately fit the true curve of the solution without the need to reduce the size of each element. It follows that FEM can also be used on unstructured grids effectively. At the element interfaces, the solution can be forced to be continuous or be left discontinuous. The continuous form resembles the FDM with the addition of basis functions to describe the values between grid points. On the other hand, the discontinuous form resembles the FVM with the addition of a polynomial representation of the solution passing through the grid points inside each volume.

Figure 1.3 illustrates the solution approximation provided by each spatial discretization method using similar grid points for a simple sinusoidal curve. In aerospace applications, the ability of a method to resolve discontinuities, such as shocks, on unstructured grids plays a large role in determining whether it will be effective or not. The FVM and DFEM schemes are able to do both effectively, which makes them appealing for aerospace CFD.

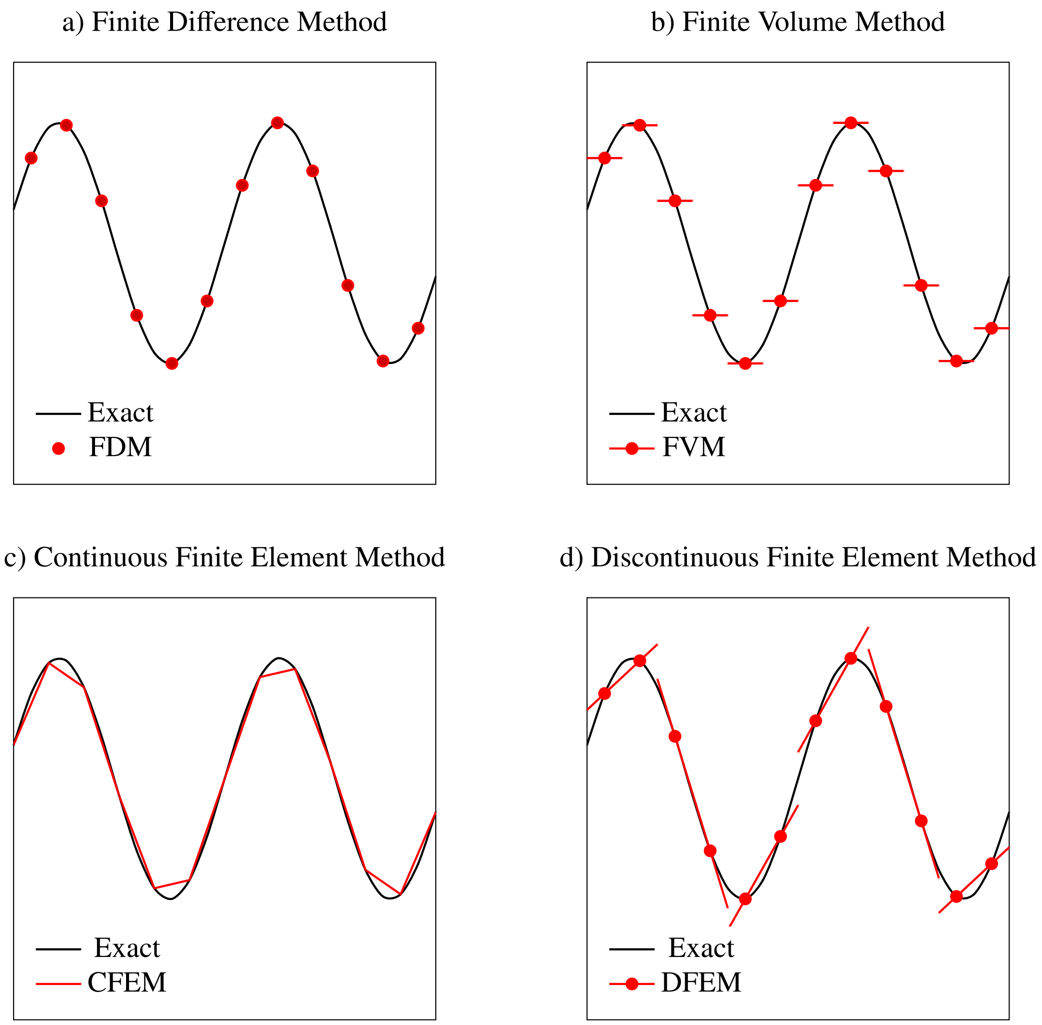


Figure 1.3. Second order spatial discretization comparison

Advantages of High-Order Methods

The FVM is widely used in industrial solvers due to its simplicity and ability to capture shocks efficiently. However, this method only provides low-order accuracy, due to the inability to refine the solution within each cell. Hence, increasing the resolution requires extremely fine grids, which prohibitively increases the time and memory needed to complete a simulation. On the other hand, a DFEM, like the DG method, can achieve the same level of accuracy with a much coarser mesh due to its ability to locally refine the resolution. This is why scale-resolving simulations, like LES and DNS, often use some form of DFEM to provide high-order accuracy while avoiding an extreme number of cells.

Along with differences in accuracy, spatial discretization methods use allocated memory differently. Every simulation typically incorporates three major types of operations, namely element-wise, point-wise direct, and point-wise indirect operations. Traditional FVM use mostly point-wise indirect operations, which require communication of information between mutliple cells. Historically, CFD solvers were run on Central Processing Units (CPUs) that shared large banks of memory, so this type of operation was not a limiting factor. Advances in accelerator hardware, such a Graphical Processing Units (GPUs), have brought high levels of parallelism and much higher peak performance compared to CPUs, for a similar monetary investment [19]. This surge in performance and parallelism signifies that element-wise and point-wise direct operations can be completed much more efficiently, but that operations requiring indirect memory did not improve as much.

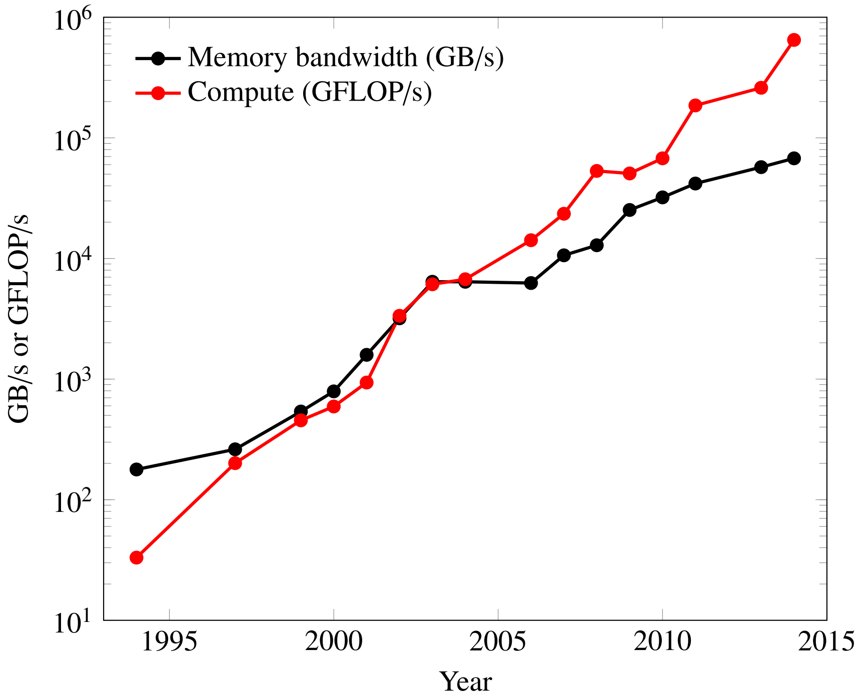


Figure 1.4. Evolution of peak processor capabilities

The trend of moving from performance limited processors towards memory bandwidth limited processors can be observed in Figure 1.4 [20]. As high performance computers evolve, this gap is only going to widen, making the FVM inefficient on modern hardware. Whereas modern families of DFEMs, like the Flux Reconstruction (FR) approach, can achieve 55% of the peak Floating-Point Operations per second (FLOP/s) of modern hardware, traditional FVM can only achieve about 3% [21]. For these reasons, recent attempts at scale-resolving simulations have stayed away from the FVM and focused on approaches like the FR or the DG method. The FR approach [22] is able to recover a wide variety of FEMs, such as the DG, SV, and SD methods depending on the flow encountered, which makes it appealing in many applications. In this study, the FR approach is employed and a detailed description of the method will be presented in Chapter 3.

1.3.2 Temporal Discretization

The temporal discretization methods used for PDEs can be categorized as implicit or explicit. To illustrate the main difference between the methods, let's consider the one-dimensional (1D) conservation law

$$\frac{\partial u}{\partial t} = R(u), \tag{1.1}$$

where $R(u)$ is the semi-discrete space operator, which is a function of the conserved variable u. For an explicit method, the spatial operator will be a function of the conserved variable at the current time step u^t, while an implicit formulation will be based on a future value of the conserved variable u^{t+1}.

Perhaps the most generalized and popular approach to discretize temporal derivatives is the family of Runge-Kutta (RK) schemes. These schemes can be written compactly in a Butcher's tableau, as shown in Table 1.1, where the vector c is composed of the positions of the stage values, the square matrix A is composed of the dependence of the stages on the derivatives found at other stages, and the vector b indicates the quadrature weights computed at the various stages [23].

c	A
	b

Table 1.1. General Butcher tableau

From the Butcher tableau of the selected scheme, the approximation of the temporal derivative at stage s is represented by

$$\frac{u_s - u}{\Delta t} = \sum_{j=1}^{s} a_{s,j} R_j, \tag{1.2}$$

and the solution at the next time-step is

$$u^{t+1} = u^t + \Delta t \sum_{i=1}^{s} b_i R_i, \tag{1.3}$$

where R in all equations is calculated using the spatial discretization technique selected. When A is a lower-diagonal matrix, the scheme is explicit, since the solution only depends on the values computed at previous time-steps and stages. Explicit time-stepping schemes efficiently utilize memory and are usually fast, but require a time-step size small enough to ensure convergence of the solution. This implies that stiff problems, characterized by regions of highly unstable flow and rapid variations, should not be solved using explicit schemes, since the time-step would need to be prohibitively small.

When the matrix A contains non-zero values on the diagonal or in the upper triangle, it is referred to as an implicit scheme. These schemes need to solve a system of linear equations every time-step, since they are based on unknown future values of the solution. Thus, using an implicit scheme will require a large memory allocation as soon as the grid becomes moderately refined. On the other hand, these methods are very robust and are not constrained by a time-step size in order to achieve convergence, explaining why they are very useful when simulating stiff systems. However, being free to select a larger step-size does not guarantee that the solution will be accurate when overly increasing the step-size.

Since a LPT cascade is usually not a stiff system, the present work will focus on explicit temporal discretization. In this study, an adaptive explicit time-stepping method is employed to advance the simulation in time and control the time-step size automatically. A detailed overview of the method, using RK approaches of fourth order, will be presented in Chapter 3.

1.4 Turbulence in Numerical Simulations

Turbulence as a phenomenon is defined as a fluid flow regime characterized by chaotic changes in pressure and velocity, creation of vortex structures and flow disturbances, boundary layer separation, etc. [15]. The Reynolds number is often used to determine the turbulence level present in a flow. This number is defined by

$$Re = \frac{Lv_\infty}{\nu}, \tag{1.4}$$

where L is the characteristic length, v_∞ is the undisturbed fluid's velocity, and ν is the kinematic viscosity of the fluid. A high Reynolds number flow is dominated by inertial forces, which are highly unstable and yield complex eddy generation. On the other hand, a low Reynolds number flow is dominated by viscous forces, which translate to rapid dissipation of eddies and stability. Stable flows are mostly linear and are referred to as laminar, whereas unstable flows are chaotic and yield turbulence. Flow fields of interest in engineering applications are mostly turbulent and their chaotic behaviour has proven difficult to capture or model accurately through numerical simulations. There are 3 common numerical approaches to simulate turbulent flows, namely DNS, the RANS equations, and LES.

1.4.1 DNS

The simpler and more general approach to simulate turbulent flows is through DNS. It involves the resolution of every time and length scale in the flow and no modeling. DNS is based on the integration of the NSE in time, which yields the instantaneous velocities at every point in space and time. The relationship between the size of the largest and smallest eddies in a flow can be related by [24]

$$\eta \approx \ell Re^{-3/4} = \left(\frac{\nu^3}{\epsilon}\right)^{1/4}, \tag{1.5}$$

where ℓ is the scale of the largest eddy, η is the scale of the smallest eddy or Kolmogorov scale, and ϵ is the energy dissipation rate. As the Reynolds number increases, the DNS grid size requirement must decrease rapidly to match the Kolmogorov scale's decrease. Hence, using DNS for highly turbulent flows requires a prohibitively expensive mesh and approaches using some form of modeling are needed for engineering applications.

1.4.2 RANS

On the other hand, RANS methods model the entire flow field in a steady-state manner instead of resolving the turbulent structures. These methods are based on a time-averaged formulation of the NSE. They make use of the Reynolds decomposition principle, which separates the instantaneous velocity of the fluid in an averaged velocity and fluctuating velocity

$$v(x,t) = \overline{v(x)} + v(x,t)'. \tag{1.6}$$

Applying this decomposition to the momentum equations for incompressible flow yields [25]

$$\rho\frac{\overline{D(v_j)}}{Dt} = \frac{\partial}{\partial x_i}\left[\mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) - \overline{p}\delta_{ij} - \rho\overline{v_i' v_j'}\right], \tag{1.7}$$

where the only unsteady term is $\rho\overline{v_i' v_j'}$ and is referred to as the Reynolds stress. In 3D flow, the RANS equations yield 6 Reynolds stresses that must be modeled adequately for each specific application of the RANS methods. Because of this, RANS methods lack generality and require deep knowledge of the flow field from its user, prior to conducting simulations. Hence, modeling inaccuracies are inherent to the RANS approach, which causes it to fail when applied to vortex-dominated regions, such as high-lift configurations [26]. This numerical approach can give accurate results at low computational costs, but its level of accuracy is limited by the type of flow encountered and the modeling technique selected.

1.4.3 LES

LES methods are based on the unsteady NSE and present a compromise between DNS and RANS equations. Whereas RANS equations are averaged in time, LES equations are averaged in space and

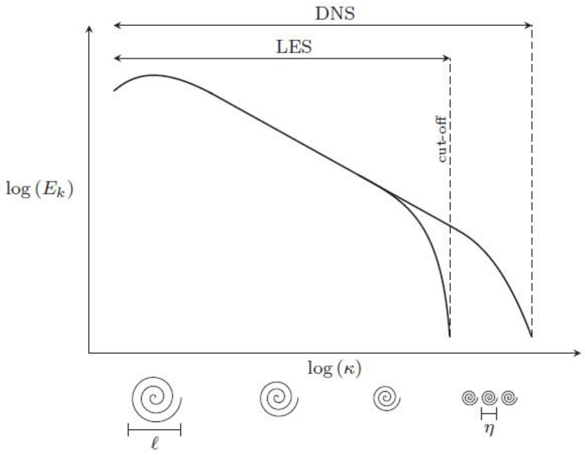


Figure 1.5. Turbulent kinetic energy cascade

capture the unsteady interactions between some turbulent structures. Instead of modeling the entire flow field, LES schemes simulate the larger turbulent structures and only model the smallest eddies. The velocity field is decomposed into a resolved and a residual component, whereas DNS has no residual component. A comparison of DNS and LES resolution is shown in Figure 1.5 [27]. The grid size usually determines the cut-off between what is resolved by the simulation and what needs to be modeled between the grid points. LES makes use of a Sub-Grid Scale (SGS) model to mimic the viscous physical dissipation that occurs naturally through the smallest eddies. The NSE are then modified to take into account this modeling through a SGS stress tensor in the momentum equations. Generally, the contribution of the SGS model is to add an artificial viscosity to achieve dissipation.

When employing a DFEM, the numerical error produced by the spatial discretization, which acts as a dissipation agent, can be used to replace the contribution of a SGS model, and create an Implicit LES (ILES) scheme. The ILES method has the advantage of being more general than RANS and LES, since it doesn't require changes in modeling techniques from one problem to another to remain accurate. Several studies have confirmed that using a DFEM, like the FR approach, to perform ILES can achieve good agreement with experimental and DNS data at a reasonable computational cost [28, 29, 30].

1.4.4 Validation of the selected methods

Before beginning the optimization of a LPT blade using the FR approach, adaptive time-stepping, and ILES, the validity of the methods must be proven. To do so, the combination of methods employed must be validated through comparison with experimental data and/or highly accurate numerical data, usually from DNS computations. This validation step ensures that the selected mathematical methodology can accurately capture important physical phenomenon such as separation, transition, and turbulence occurring in reality and affecting the performance of a high-lift LPT blade. Hence, the methodology will first be applied to the T106D LPT cascade case in Chapter 4 and validated through comparison with available experimental data from Stadtmuller et al. [31].

With the CFD framework now set, the next section will focus on the optimization process and the different possible approaches.

1.5 Previous Work on Optimization

As the role of CFD rose to prominence in engineering applications over the years, so did its use in design practices. Instead of only using numerical simulations to confirm whether or not a design was satisfactory, it became increasingly used in the process of optimizing the design directly. Hence, optimization methods to converge towards an optimal design, using CFD results as inputs, were required.

The optimal design is usually quantified using an objective function, which combines one or more performance parameters of interest into a single value, of which the minimum is desired. For all existing optimization methods, the goal of the process is to use the results of each CFD simulation to calculate an objective function and then modify the shape of the designed object to move towards the optimal value of this function. The objective function can include aerodynamic properties, thermodynamic properties, structural properties, etc. In the current study, blade design refers to aerodynamic shape optimization, so the objective functions will be focused on aerodynamic properties generated by the blade.

There are two major categories of optimization techniques, namely the gradient-based and gradient-free methods. Choosing the right optimization strategy is highly problem dependent, and has a sizable impact on the computational cost and effectiveness of the process [32]. The next sections will expand upon the concept of each category, some of the most relevant methods, and review the previous work achieved with each.

1.5.1 Gradient-Based Methods

Pironneau [33] introduced an optimal control mathematical framework to obtain a minimum drag design in low Reynolds number flows. This paper served as the first step in the evolution of gradient-based methods. These methods use the sensitivities of the objective function with respect to the design variables to help guide each design iteration towards an optimum. Perhaps the most common gradient-based method used in aerodynamic shape optimization is the adjoint method, advocated by Jameson [34, 35]. To help understand the underlying concepts of the method, an abstract description is provided, by considering a simple objective function of the form

$$\mathcal{I} = \mathcal{I}(U, \mathcal{F}), \tag{1.8}$$

where U is the vector of flow field variables, and $\mathcal{F}$ is a function representing the physical boundary location. When changing the boundary shape, the change in objective function can be written as

$$\delta \mathcal{I} = \frac{\partial \mathcal{I}^T}{\partial U} \delta U + \frac{\partial \mathcal{I}^T}{\partial \mathcal{F}} \delta \mathcal{F}. \tag{1.9}$$

Through the implementation of the governing equation $\mathcal{R}$ and the application of the adjoint equation, the change in objective function can be rewritten as

$$\delta \mathcal{I} = \left[\frac{\partial \mathcal{I}^T}{\partial \mathcal{F}} - \psi^T \frac{\partial \mathcal{R}}{\partial \mathcal{F}} \right] \delta \mathcal{F}, \tag{1.10}$$

where ψ is a Lagrange multiplier. The absence of any term related to δU in Equation 1.10 illustrates how the method permits a gradient determination without needing to re-evaluate the flow field. The evaluation of the gradient has a computational cost similar to a single flow field evaluation. Hence, the computational cost of the adjoint for problems with many design variables remains relatively low, which is its most attractive feature.

The adjoint has been employed to solve countless optimization problems in recent years, including simple airfoils [36], full wing body configurations [37, 38], steam turbine blades [39], and a full HPT stage with cooling flow [40]. This method is usually used in conjunction with RANS flow solutions, since they are steady-state, which helps to compute stable sensitivities. Unfortunately, using LES or DNS solutions of the NSE with the adjoint has proven unfeasible, because it cannot compute usable sensitivities for chaotic systems [41]. An attempt to generate sensitivities for chaotic systems using non-intrusive least-squares shadowing by Blonigan et al. [42] showed that it can be achieved for a few simple, low Reynolds number, and 2D flows, but that the expansion to more complex flows is still extremely impractical. Hence, the adjoint is mainly used to solve optimization problems where the assumption of fully-turbulent flow [43] or other simplifications can be made to render RANS flow solutions satisfactory.

1.5.2 Gradient-Free Methods

On the other hand, gradient-free methods do not require any information about the derivative of the objective function to converge towards the optimal design. The general working principle is that each design iteration requires multiple flow solutions to determine a new optimum, with the number of flow solutions determined from the number of design variables. Hence, the computational cost of gradient-free optimization can be very high if a large number of design variables are employed, compared to gradient-based optimization. Nonetheless, gradient-free methods offer appealing properties for a variety of problems, such as problems requiring great accuracy and problems with discontinuous or noisy objective functions. Since the sensitivities of the objective function are not needed, these methods can successfully conduct optimizations with scale-resolving flow solutions. These methods are more robust than their gradient-based counterparts, and there is also a higher likelihood that a gradient-free method will find the true global optimum and not get trapped in a local optimum [38, 44, 45].

Most gradient-free methods can be categorized as either a population-based method or a search method. Population-based methods, such as the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) [46], begin from a vast and randomly generated array of design points covering the entire design space. From there, evolutionary principles observed in nature are employed to converge the points towards the best possible design. When applied to aerospace optimization problems, the GA has shown that it can quickly determine the region of optimal design, but is then very slow to converge towards the true optimum [32]. This is mainly due to the high number of flow solutions necessary at each design iteration, which renders the method very computationally expensive. In addition, these methods, especially the PSO, are prone to poor performance if the initial population distribution is improper [47].

Whereas population-based methods start from a wide array of possible designs, search methods start from a single design point. A grid is constructed around the initial design point, objective function evaluations are conducted at the grid points, and the grid is then translated to a new optimum or shrank around the previous optimum. These methods enable a more computationally efficient optimization, due to the lower number of flow solutions needed at each design iteration.

The selected optimization method in this study is the Mesh Adaptive Direct Search (MADS) algorithm, introduced by Audet and Dennis [48]. MADS lies somewhere between the Generalized Pattern Search (GPS) class of algorithms [49] and the Coope and Price frame-based methods [50]. The main advantage of MADS over GPS is that the local exploration of the design space is not limited to a finite number of directions, called poll directions [51]. The MADS code that is employed in this study comes from Karbasian [52].

MADS has previously been used for optimization using LES of the SD7003 airfoil at low Reynolds numbers by Karbasian and Vermeire [53]. MADS has also been used successfully with RANS simulations to design hydraulic turbine runner blades by Bahrami [54] and Bahrami et al. [55]. It was also used to optimize the Fontan procedure by simulating the flow field within vessels by Yang [56]. The full mathematical description of the MADS algorithm will be presented in Chapter 5.

1.6 Thesis Objectives and Contribution

The objective of this thesis is to conduct multiple aerodynamic shape optimizations of a LPT cascade blade using ILES and show that this methodology is viable. To the author's knowledge, producing LPT cascade blade designs using ILES has not been attempted before, so this will be a major contribution of this study. Since the efficiency of LPT blades is mainly determined from 2D flow phenomena, as discussed in Section 1.2.1, the focus will remain on the mid-span section of a blade.

First, a grid convergence study is presented using the Python Flux Reconstruction (PyFR) [20] solver for ILES of the T106D cascade. Validation of the selected methodology and polynomial approximation is then conducted using readily available experimental data of the T106D cascade. This portion of the study aims to prove that accurate measurement of separation and transition can be made using the aforementioned methodology.

Then, shape optimization is conducted initially using 2D simulations in order to set the design space and objective functions, all while providing a first set of results to analyze. For all optimizations, the initial blade is the T106D, for which the solver and mesh were previously validated. Two blade designs are then produced using 3D simulations and distinct objective functions. The first objective function aims to provide a classical optimization, where increasing the efficiency of the blade is the main goal. The second optimization aims to increase the tangential force produced by the blade without decreasing its efficiency. This last optimization cycle will also show the ability of the method to optimize near a design space where separation and transition is constantly present, since high-lift blades are the objective.

Finally, conclusions are drawn as to the effectiveness of the method and its potential utility in research and industrial applications.

1.7 Thesis Outline

In Chapter 2, the governing equations of fluid flow are presented in details to introduce the theoretical concepts behind CFD. In Chapter 3, the numerical schemes used throughout this study are detailed. In Chapter 4, the validation case is described, the selected grid is introduced, and the validation results are illustrated. In Chapter 5, the MADS algorithms is explained, along with the presentation of the shape deformation technique and the design parameters employed during the study. In Chapter 6, descriptions and results are provided for every optimization conducted. It includes an initial study using

2D simulations, followed by two separate optimizations using 3D simulations. In Chapter 7, a summary of the thesis is presented, along with future work recommendations.

Chapter 2

Governing Equations

A fluid flow can be described by a set of conservation laws, known as the NSE. In this chapter, the conservation laws are presented in their integral and divergence form. It is important to note that the FR approach, employed in this study, makes use of these laws in their divergence formulation. A more detailed description of these laws is provided by Hirsch [57]. The following sections detail the mass, momentum and energy conservation laws, as well as the full NSE.

2.1 General Conservation Law

The definition of a general conservation law begins with the identification of a conserved fluid variable u. Considering a control volume Ω fixed in space, the rate of change of the conserved variable within the control volume is given by

$$\frac{d}{dt}\int_{\Omega} u d\Omega. \tag{2.1}$$

Considering the control volume's surface $\partial\Omega$ and the flux vector F through it, the rate of change of the conserved variable entering and exiting the domain is given by

$$-\oint_{\partial\Omega} F \cdot ds, \tag{2.2}$$

where ds is the surface element vector, pointing in the outward direction. Combining the previous equations and adding the effects of sources $S(u,t)$, the general conservation law is written as

$$\frac{d}{dt}\int_{\Omega} u d\Omega + \oint_{\partial\Omega} F \cdot ds = \int_{\Omega} S(u,t) d\Omega. \tag{2.3}$$

This is the integral form of the general conservation law. To get the divergence formulation, the surface integral is converted to a volume integral using Gauss' theorem

$$\oint_{\partial\Omega} F \cdot ds = \int_{\Omega} \nabla \cdot F d\Omega, \tag{2.4}$$

where ∇ is the divergence operator. Following this change to the flux term, all terms can be combined under the same volume integral. For the integral form to be valid, conservation must also be observed locally at any point, leading to the integral to vanish and the divergence form of the conservation law to appear as

$$\frac{\partial u}{\partial t} + \nabla \cdot F = S(u, t). \tag{2.5}$$

2.2 Mass Conservation Equation

From the mass conservation law, mass cannot be created or destroyed. For this conservation equation, the density ρ of the fluid is used as the conserved variable. Since no mass can be created, there is no source term included and the flux vector is linearly proportional to the velocity v. The integral form of the mass conservation law is described by

$$\frac{d}{dt} \int_{\Omega} \rho d\Omega + \oint_{\partial\Omega} \rho(V \cdot ds) = 0, \tag{2.6}$$

where V is the velocity vector. By using Gauss' theorem, the divergence form is obtained as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho V) = 0. \tag{2.7}$$

2.3 Momentum Conservation Equation

Momentum is the product of mass and velocity, which can also be represented by the product of density and velocity. Hence, the conserved variable becomes the vector quantity ρV in the momentum conservation equation. The flux of momentum in a system can then be defined by

$$F = \rho V \otimes V, \tag{2.8}$$

where ⊗ denotes the tensor product of the vectors. The integral form of the momentum conservation law can then be represented by

$$\frac{d}{dt}\int_{\Omega}\rho V d\Omega + \oint_{\partial\Omega}\rho V(V\cdot ds) = \int_{\Omega}\rho f_e d\Omega + \oint_{\partial\Omega}\overline{\overline{\sigma}}\cdot ds, \tag{2.9}$$

where the source term in this instance is represented by the combination of the body force f_e and stress tensor $\overline{\overline{\sigma}}$ acting on the fluid. Assuming the fluid is Newtonian, the total internal stress tensor is represented as

$$\overline{\overline{\sigma}} = -p\overline{\overline{I}} + \overline{\overline{\tau}}, \tag{2.10}$$

where p is the pressure, $\overline{\overline{I}}$ is the unit tensor, and $\overline{\overline{\tau}}$ is the viscous shear stress tensor represented by

$$\overline{\overline{\tau}} = \mu\left[\left(\frac{\partial V_j}{\partial x_i} + \frac{\partial V_i}{\partial x_j}\right) + \lambda(\nabla\cdot V)\delta_{ij}\right], \tag{2.11}$$

where μ is the dynamic viscosity of the fluid, λ is a viscosity coefficient, and δ_{ij} is the Kronecker delta defined as

$$\delta_{ij} = \begin{cases} 0 & i \neq j, \\ 1 & i = j. \end{cases} \tag{2.12}$$

For a Newtonian fluid in equilibrium, $\lambda = -2/3$. By using Gauss' theorem and some rearranging, the divergence form is obtained as

$$\frac{\partial(\rho V)}{\partial t} + \nabla\cdot(\rho V\otimes V + p\overline{\overline{I}} - \overline{\overline{\tau}}) = \rho f_e. \tag{2.13}$$

2.4 Energy Conservation Equation

The energy content in a continuous closed system is given by its internal energy per unit mass e. In a fluid, the kinetic energy per unit mass of the flow must also be considered, leading to the conserved variable being the total energy defined as

$$E = e + \frac{v^2}{2}, \tag{2.14}$$

where v denotes the magnitude of the velocity vector. From the first law of thermodynamics, the flux of total energy can be defined by adding the convective and diffusive fluxes

$$F = F_C + F_D = \rho E V - k\nabla T, \tag{2.15}$$

where T is the absolute static temperature and k is the thermal conduction coefficient which is formulated as

$$k = \frac{c_p \mu}{Pr}, \tag{2.16}$$

where c_p is the specific heat at constant pressure, and Pr is the Prandtl number which quantifies the ratio of momentum diffusivity to thermal diffusivity. Replacing these formulations and the source terms in the general conservation law yields the integral form of the energy conservation law

$$\frac{d}{dt}\int_{\Omega} \rho E d\Omega + \oint_{\partial\Omega} (\rho E V - k\nabla T) \cdot ds = \int_{\Omega} (\rho f_e \cdot V + q_H) d\Omega + \oint_{\partial\Omega} \left(\overline{\overline{\sigma}} \cdot V\right) \cdot ds, \tag{2.17}$$

where q_H represents the heat sources other than conduction and the other energy sources come from the work from the internal forces. In its divergence form, the equation is defined as

$$\frac{\partial(\rho E)}{\partial t} + \nabla \cdot \left(\rho V E - k\nabla T - \overline{\overline{\sigma}} \cdot V\right) = W_f + q_H, \tag{2.18}$$

or in alternative form

$$\frac{\partial(\rho E)}{\partial t} + \nabla \cdot \left(\rho V H - k\nabla T - \overline{\overline{\tau}} \cdot V\right) = W_f + q_H, \tag{2.19}$$

where W_f is the work of the external volume forces

$$W_f = \rho f_e \cdot V, \tag{2.20}$$

and H is the total enthalpy defined as

$$H = e + \frac{p}{\rho} + \frac{v^2}{2} = E + \frac{p}{\rho}. \tag{2.21}$$

2.5 Navier-Stokes Equations

The current section groups the conservation laws presented previously as the NSE. These equations describe the compressible and viscous fluid flow encountered in reality. The 3D NSE in divergence form can be written in vector notation as

$$\frac{\partial U}{\partial t} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} = S(U, t), \tag{2.22}$$

where each flux vector can be decomposed into its inviscid and viscous components such that

$$\frac{\partial F_i}{\partial i} = \frac{\partial}{\partial i}\left(F_{i,inv} - F_{i,vis}\right), \tag{2.23}$$

where i represents any of the three physical directions, namely x, y and z. In the NSE, the vector of conserved variables U is defined as

$$U = \begin{bmatrix} \rho \\ \rho V_x \\ \rho V_y \\ \rho V_z \\ \rho E \end{bmatrix}, \tag{2.24}$$

each vector of inviscid fluxes is defined as

$$F_{i,inv} = \begin{bmatrix} \rho V_i \\ \rho V_i V_x + \delta_{ix} p \\ \rho V_i V_y + \delta_{iy} p \\ \rho V_i V_z + \delta_{iz} p \\ \rho V_i H \end{bmatrix}, \tag{2.25}$$

each vector of viscous fluxes is defined as

$$F_{i,vis} = \begin{bmatrix} 0 \\ \tau_{xi} \\ \tau_{yi} \\ \tau_{zi} \\ k\partial_i T + \tau_i \cdot V \end{bmatrix}, \tag{2.26}$$

and the vector of source terms is defined as

$$S(U,t) = \begin{bmatrix} 0 \\ \rho f_{e,x} \\ \rho f_{e,y} \\ \rho f_{e,z} \\ W_f + q_H \end{bmatrix}. \tag{2.27}$$

In these equations, τ is a component of the viscous shear stress, $f_{e,i}$ is a component of the body forces acting on the fluid, and $\partial_i T$ is the derivative of the absolute temperature in a specific direction.

Chapter 3

Numerical Schemes

Now that the equations of fluid motion have been described, a suitable set of discretization schemes must be introduced to solve these spatial and temporal PDEs. To illustrate how these methods operate, a general conservation law in divergence form will be discretized in the following sections. The conservation law is expressed as

$$\frac{\partial U}{\partial t} + \nabla \cdot F = 0, \tag{3.1}$$

where U is the vector of conserved variables, t is time, and F is the flux vector. For simplicity, the schemes will be explained with a single conserved variable u instead of the full vector of conserved variable U. The following sections detail the FR approach in 1D and multiple dimensions, as well as the selected temporal discretization scheme employed in this work.

3.1 Flux Reconstruction Method

The FR approach is the spatial discretization technique employed in this study. First introduced by Huynh [22, 58] in 1D, the FR approach is a compact scheme able to recover multiple high-order FEM including the DG, SD, and SV schemes on mixed-element unstructured grids for linear flux functions [29]. This method was later extended to 2D and 3D via the Lifting Collocation Penalty (LCP) formulation [59], and the Correction Procedure via Reconstruction (CPR) scheme [60], respectively. This approach makes use of an element-wise polynomial representation of the solution, of which the degree can be increased to obtain high-order accuracy. With the memory layout leveraging the internal data locality of the polynomial solution representation, the majority of computationally expensive operations involved in the FR approach can be cast as matrix multiplications to effectively utilize the capabilities of modern many-core hardware architectures, such as GPUs [20, 30, 61, 62, 63]. A more detailed mathematical description of the FR approach is provided by Huynh [22].

3.1.1 1D Formulation

In 1D, the general conservation law can be explicitly written as

$$\frac{\partial u}{\partial t} + \frac{\partial F_x}{\partial x} = 0, \tag{3.2}$$

where $F_x = F_x(u)$ is the flux component of $u = u(x,t)$ in the physical x direction. The FR approach is used to discretize the spatial PDE included in this conservation law. The process begins with the separation of the domain Ω into a grid of N_e non-overlapping elements such that

$$\Omega = \bigcup_{i=1}^{N_e} \Omega_i, \qquad \bigcap_{i=1}^{N_e} \Omega_i = 0, \tag{3.3}$$

where each element Ω_i is defined by $x \in [x_{i,L}, x_{i,R}]$. The solution within each element is discretely defined by a set of N_p solution points. Throughout the element, the solution is defined by a polynomial of degree $\mathbb{P} = N_p - 1$ and passes through these N_p points, forming a continuous solution approximation in each element. Similarly, the flux in each element is approximated by a flux polynomial of degree $\mathbb{P} + 1$. The approximations of the solution and flux can be discontinuous across element interfaces. It follows that the approximation of the solution can be summarized by

$$u \approx u^h = \bigoplus_{i=1}^{N_e} u_i^h, \qquad F_x \approx F_x^h = \bigoplus_{i=1}^{N_e} F_{x,i}^h, \tag{3.4}$$

where u^h and F_x^h are the piece-wise continuous approximations over the entire domain, and u_i^h and $F_{x,i}^h$ are the continuous approximations within each element.

For simplicity and compactness, each element Ω_i is mapped from the physical space $x \in \mathbb{R}$ to a reference space $\xi \in [-1, 1]$, generating reference element Ω_i'. The mapping function to convert from the physical to the reference space must be invertible, such that

$$x(\xi) = \frac{1}{2}(x_{i,L} + x_{i,R} + \xi l_i) \quad \Leftrightarrow \quad \xi(x) = \frac{2}{l_i}\left(x - \frac{x_{i,L} + x_{i,R}}{2}\right), \tag{3.5}$$

where l_i is the size of the physical element. From there, the spatial PDE can be converted using

$$\frac{\partial x}{\partial \xi} = \frac{l_i}{2} \quad \Leftrightarrow \quad \frac{\partial \xi}{\partial x} = \frac{2}{l_i}. \tag{3.6}$$

The conservation law in the reference element then becomes

$$\frac{\partial u_i^h}{\partial t} + \frac{2}{l_i}\frac{\partial F_{x,i}^h}{\partial \xi} = 0, \tag{3.7}$$

where $u_i^h = u_i^h(\xi, t)$, and $F_{x,i}^h = F_{x,i}^h(u_i^h)$. In order to get this flux approximation within each element, an approximation of the solution must be generated in the reference space. To approximate the solution continuously within the reference element, a polynomial nodal basis representation is used such that

$$u_i^h(\xi, t) = \sum_{m=1}^{N_p} u_{i,m}(t)\phi_m(\xi), \tag{3.8}$$

where $u_{i,m}(t)$ is the value of the solution at one of the N_p points, and $\phi_m(\xi)$ is its corresponding nodal basis function expressed as

$$\phi_m(\xi) = \prod_{q=1,q\neq m}^{N_p} \frac{\xi - \xi_q}{\xi_m - \xi_q}. \tag{3.9}$$

Figure 3.1 shows examples of nodal basis functions for an element with up to $\mathbb{P} = 5$, also noted $\mathbb{P}_5$, using Gauss points. These basis functions take a value of 1 at their respective points and a value of 0 at every other point in the reference element. Since the solution points are at constant positions for all elements in the reference space, the nodal basis functions are also identical for every mapped element. Hence, computations to define the nodal basis functions only need to be conducted once, given that all elements use the same polynomial degree representation.

Now that the solution is fully represented in each element, the flux can be represented continuously in each element using the same nodal basis function notation such that

$$F_{x,i}^h(\xi, t) = \sum_{m=1}^{N_p} F_{x,i,m}(t)\phi_m(\xi), \tag{3.10}$$

where $F_{x,i,m}(t)$ is the value of the flux at one of the N_p points, and $\phi_m(\xi)$ is its corresponding nodal basis function. These nodal basis functions are taken at the same N_p solution points and are then the same functions as for the solution representation. Since these flux functions are discontinuous at each element interface, they must be reconstructed to ensure conservation over the entire domain Ω and that the derivative of the flux functions lie in the same polynomial space as the solution. This continuous flux will then be defined by $N_p + 1$ points and a polynomial of order $\mathbb{P} + 1$. For the remainder of this chapter, the continuous flux will be represented by $F_{x,i}^C$ and the discontinuous flux will be kept as $F_{x,i}^h$.

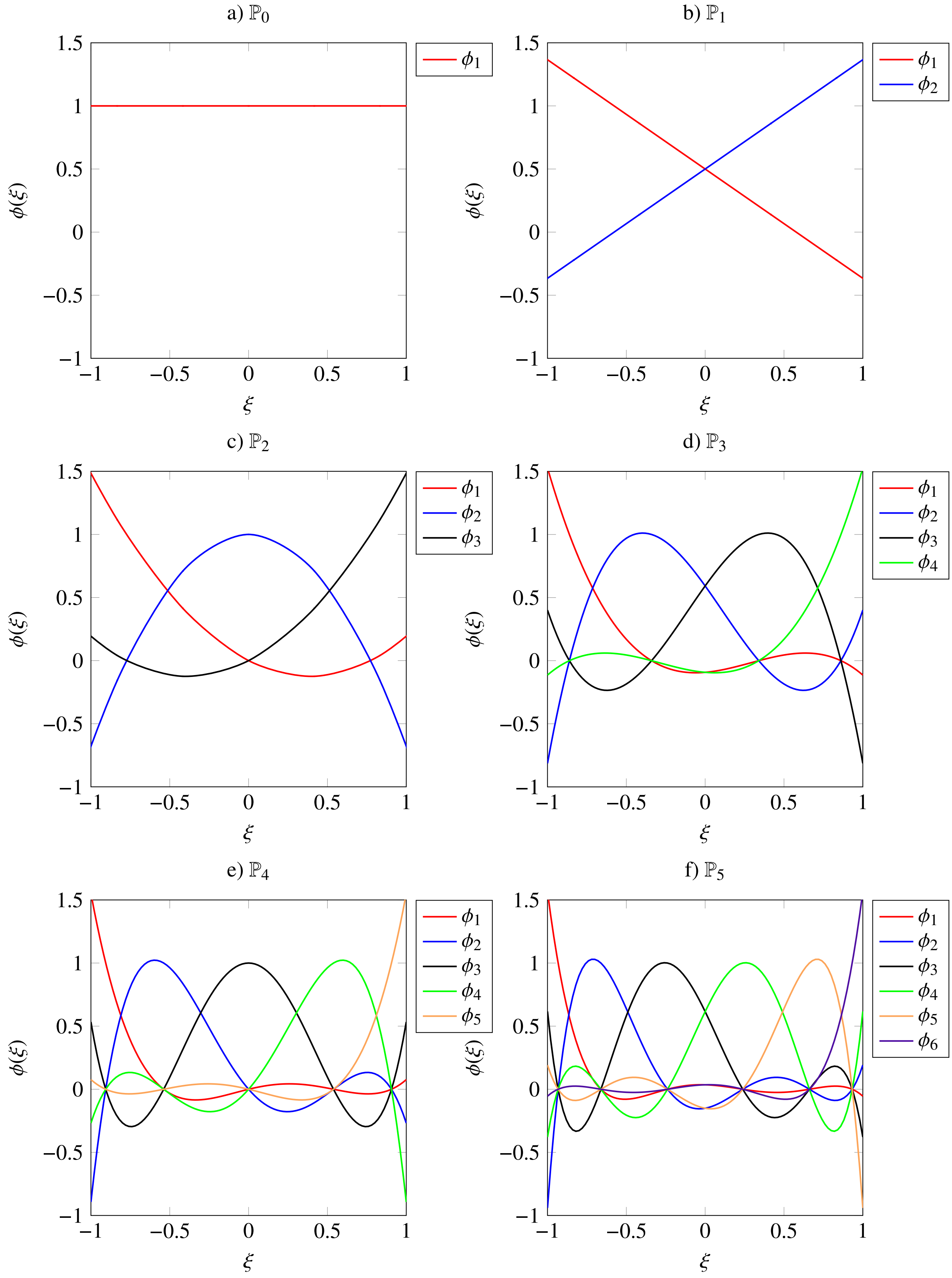


Figure 3.1. Nodal basis functions for 1D elements using $\mathbb{P}_0$ to $\mathbb{P}_5$

The relationship between the fluxes is given by

$$F^C_{x,i} = F^h_{x,i} + \Psi_i, \tag{3.11}$$

where Ψ_i is a correction field for fluxes in the reference space, which can be defined as

$$\Psi_i = \left[F^*_{x,i,L} - F^h_{x,i}(-1)\right] g_L(\xi) + \left[F^*_{x,i,R} - F^h_{x,i}(1)\right] g_R(\xi), \tag{3.12}$$

where $F^*_{x,i}$ is the common interface flux at the left boundary L and right boundary R, and $g(\xi)$ is a correction function. The common interface fluxes are determined from a selected Riemann solver, which determines the interface values to use for the continuous flux function. Examples of Riemann solvers include the upwind, central, Roe, and Rusanov schemes, among many others. The Riemann solver must be chosen adequately, since it can greatly affect the magnitude and type of numerical error encountered [64]. The correction function must be of polynomial degree $\mathbb{P} + 1$ and satisfy

$$\begin{aligned} g_L(-1) &= 1, \qquad & g_L(1) &= 0, \\ g_R(-1) &= 0, \qquad & g_R(1) &= 1. \end{aligned} \tag{3.13}$$

Employing different correction functions enables the recovery of certain specific schemes, such as SD, SV and DG. For example, using the Radau polynomials as correction functions effectively recovers the DG scheme, which is described in details in the work of Huynh [22]. Since the current study employs the Radau polynomials to recover the DG scheme, Figure 3.2 illustrates some of the Radau polynomials. The left Radau polynomials are used to define g_R, whereas the right Radau polynomials are used to define g_L. Hence, a solution employing a $\mathbb{P}_2$ representation in each element would need $Ra_{R,3}$ and $Ra_{L,3}$ to represent g_L and g_R, respectively.

Finally, replacing the discontinuous flux by the continuous flux in the conservation law yields

$$\frac{\partial u^h_i}{\partial t} + \frac{2}{l_i}\frac{\partial}{\partial \xi}\left(F^h_{x,i} + \Psi_i\right) = 0, \tag{3.14}$$

$$\frac{\partial u^h_i}{\partial t} + \frac{2}{l_i}\sum_{m=0}^{N_p}\left[F^h_{x,i,m}\frac{\partial \phi_m}{\partial \xi} + \left(F^*_{x,i,L} - F^h_{x,i}(-1)\right)\frac{\partial g_L}{\partial \xi} + \left(F^*_{x,i,R} - F^h_{x,i}(1)\right)\frac{\partial g_R}{\partial \xi}\right] = 0, \tag{3.15}$$

in which all PDEs are in the same polynomial space of degree $\mathbb{P}$.

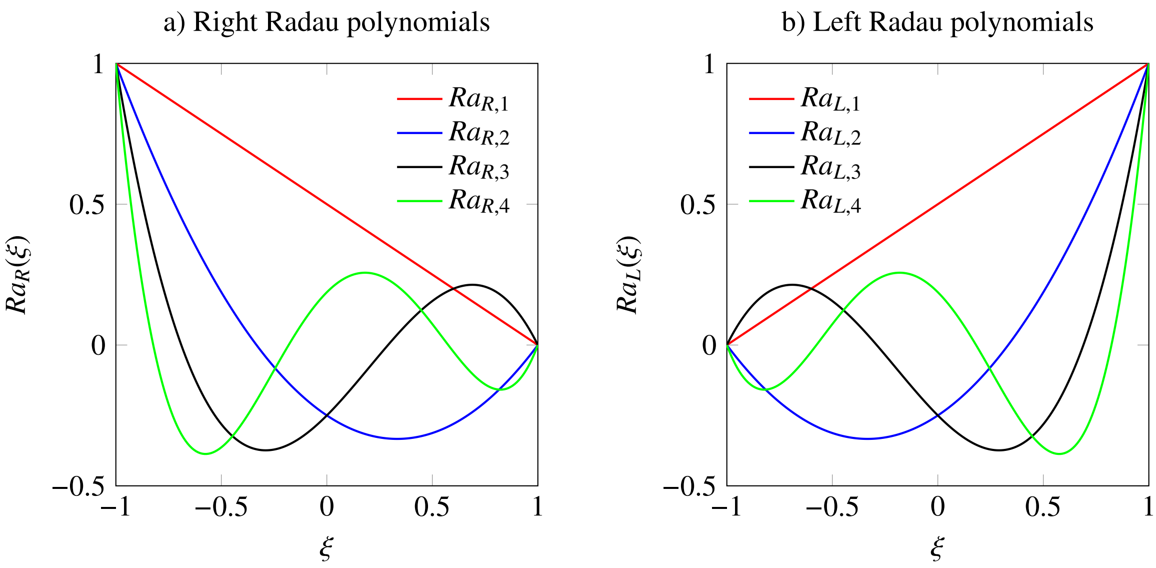


Figure 3.2. Radau polynomials up to fourth-order

3.1.2 Multi-Dimensional Formulation

In this section, the 1D procedure is extended to 2D. The extension to 3D is not presented here, as the required steps are the same as for the 2D extension. For reference, the complete extension to 2D can be found in the work of Wang and Gao [59], and the extension to 3D in the work of Haga et al. [60].

In 2D, the general conservation law can be explicitly written as

$$\frac{\partial u}{\partial t} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} = 0, \tag{3.16}$$

where $F_y = F_y(u)$ is the flux component of $u = u(x, y, t)$ in the physical y direction. Similarly to 1D, the domain is divided into N_e non-overlapping elements $\Omega_{i,j}$. For simplicity in this example, the elements are considered to be rectangular. Each element is then mapped onto a square reference element $\Omega'_{i,j}$ using an invertible mapping function such that

$$(x, y) = \Gamma_{i,j}(\xi, \zeta) \quad \Leftrightarrow \quad (\xi, \zeta) = \Gamma^{-1}_{i,j}(x, y), \tag{3.17}$$

where $\Gamma_{i,j}$ is the mapping function, and ζ is the direction analogous to y in the reference space. The reference element is such that $\xi \in [-1, 1]$ and $\zeta \in [-1, 1]$. The solution is then approximated using a polynomial nodal basis representation of the true solution such that

$$u^h_{i,j}(\xi, \zeta, t) = \sum_{m,n=1}^{N_p} u_{i,j,m,n}(t)\phi_{m,n}(\xi, \zeta), \tag{3.18}$$

where $u_{i,j,m,n}(t)$ is the value of the solution at one of the $N_p \times N_p$ points, and $\phi_m, n(\xi, \zeta)$ is its corresponding nodal basis function, which can be split into

$$\phi_{m,n}(\xi, \zeta) = \phi_m(\xi)\phi_n(\zeta), \tag{3.19}$$

where each function ϕ can constructed from Equation 3.9 previously shown. It should be noted that the number of solution points in each direction is the same N_p, since the element is rectangular.

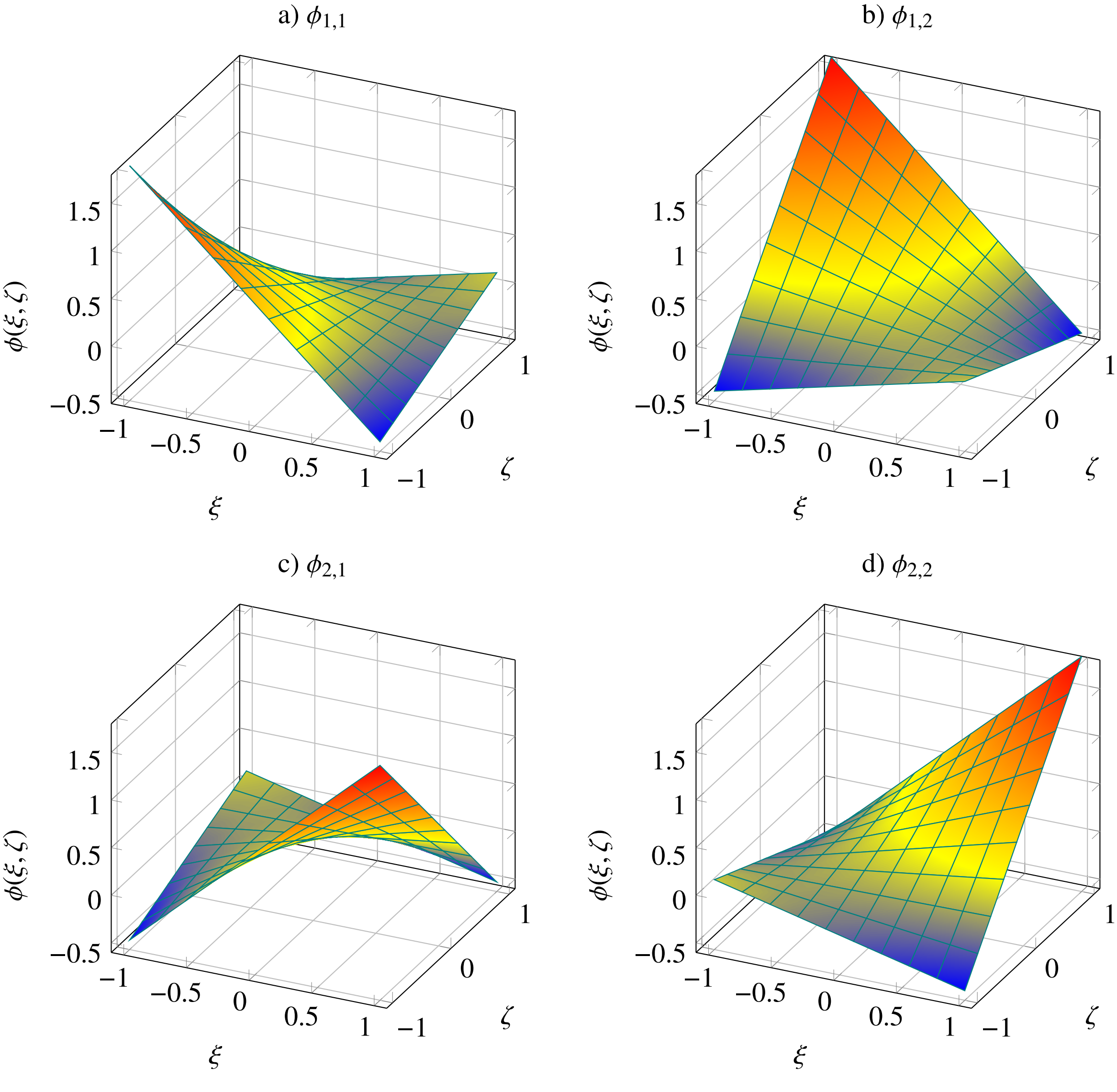


Figure 3.3. Nodal basis functions for a 2D reference element with $\mathbb{P}_1$

Figure 3.3 illustrates the four nodal basis functions for an element with $\mathbb{P}_1$ using Gauss points. Again, each function $\phi_{m,n}$ takes a value of 1 at the point (ξ_m, η_n) and a value of 0 at all other solution points. From the solution polynomial approximation, the discontinuous flux can also be represented using the same nodal basis functions such that

$$F^h_{x,i,j}(\xi, \zeta, t) = \sum_{m,n=1}^{N_p} F_{x,i,j,m,n}(t)\phi_{m,n}(\xi, \zeta), \tag{3.20}$$

$$F^h_{y,i,j}(\xi, \zeta, t) = \sum_{m,n=1}^{N_p} F_{y,i,j,m,n}(t)\phi_{m,n}(\xi, \zeta), \tag{3.21}$$

where $F_{x,i,j,m,n}(t)$ and $F_{y,i,j,m,n}(t)$ are the value of the fluxes at one of the N_p points, and $\phi_{m,n}(\xi, \zeta)$ is the corresponding nodal basis function. Similarly to 1D, the discontinuous fluxes must now be reconstructed to ensure conservation over the entire domain Ω and that the derivative of the flux functions lie in the same polynomial space as the solution. Hence, the discontinuous flux can be corrected in the conservation law as

$$\frac{\partial u^h_{i,j}}{\partial t} + \frac{\partial F^h_{x,i,j}}{\partial \xi} + \frac{\partial F^h_{y,i,j}}{\partial \zeta} + \Psi_{i,j} = 0, \tag{3.22}$$

where $\Psi_{i,j}$ is a scalar correction field for fluxes in the reference space, defined as

$$\Psi_{i,j} = \frac{1}{|\Omega_{i,j}|} \sum_{f \in S} \sum_{l} a_{i,j,f,l} [\tilde{F}]_{f,l} S_f, \tag{3.23}$$

where $|\Omega_{i,j}|$ is the element area in 2D, or its volume in 3D, $a_{i,j,f,l}$ are constant lifting coefficients independent of the solution and geometry, $[\tilde{F}]_{f,l}$ is the flux difference polynomial at flux point l along element boundary f, and S_f is the length in 2D, or area in 3D, of boundary f. Depending on the value of the lifting coefficient $a_{i,j,f,l}$, a wide range of energy stable schemes can be recovered, such as the SD, SV and DG methods.

When applying this spatial discretization scheme to the NSE, it is important to divide the flux into its inviscid and viscous components. The disparity observed between these two components implies that the ideal Riemann solver used for the common fluxes of each component should be different in order to obtain a more accurate flux reconstruction. Hence, the FR approach splits the flux components, when applied to the NSE. In this study, the Rusanov Riemann solver is employed to reconstruct the inviscid fluxes, while the Local Discontinuous Galerkin (LDG) scheme is used for the viscous fluxes [58]. Gauss-Legendre points are employed on the faces and in the volumes of each element for solution and flux representation.

3.2 Time-Stepping Schemes

With the spatial PDEs discretized, the focus can be put on advancing the solution in time. For simplicity, the resulting expression from the spatial discretization is replaced by the space operator $R(u)$ such that the conservation law can be written as

$$\frac{\partial u}{\partial t} + R(u) = 0. \tag{3.24}$$

In this study, a two-register five-stage fourth-order adaptive RK (RK45[2R+]) scheme [65] is employed to discretize the temporal PDE. Table 3.1 shows the Butcher tableau of the employed scheme.

0					
0.225022459	0.225022459				
0.595272620	0.051229307	0.544043313			
0.576752376	0.051229307	0.380954826	0.144568244		
0.845495878	0.051229307	0.380954826	-0.373352596	0.786664342	
	0.051229307	0.380954826	-0.373352596	0.592501285	0.348667179
	0.137217322	0.191880762	-0.229206721	0.624294677	0.275813960

Table 3.1. RK45[2R+] Butcher tableau

Since this is an adaptive time-stepping scheme, there are two vectors b indicating the quadrature weights of the various stages. Using these different weights and the same stages, this method generates two distinct solutions for only a slightly higher computational cost. The step-size is adjusted by observing the difference between the two solutions and comparing it with a threshold, selected to be 10^{-6} in this study. When the difference is significantly above the threshold, the step-size is decreased in the following iteration. When the difference is significantly below the threshold, the step-size is increased in the following iteration. These adjustments are performed every iteration by a PI controller, until the step-size is close enough to the threshold. Using this methodology, the initial step-size only serves as a starting point and doesn't need to be calculated or determined precisely prior to any computations.

Chapter 4

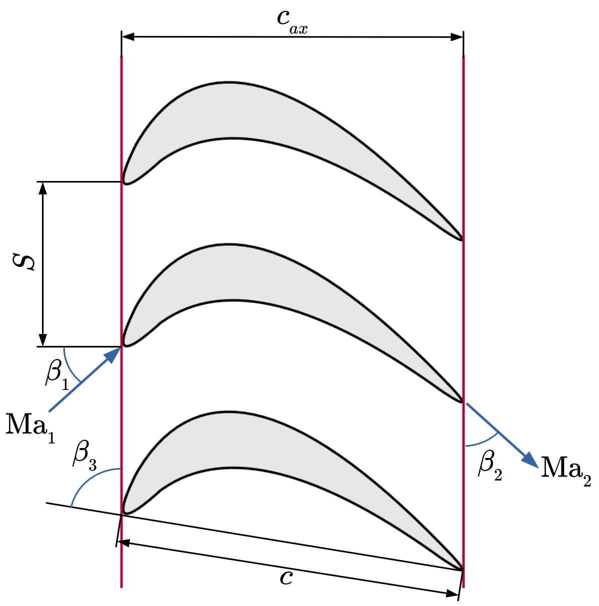
T106D Cascade Validation

With the employed numerical methods defined, their application to a LPT case must be tested before attempting any optimization activity. In this study, the focus is put on resolving mainly 2D flow phenomena, since they account for the large majority of the losses occurring in LPTs. Hence, a linear cascade using a mid-span of a LPT blade is considered for the study. A linear cascade consists of a number of identical parallel blades constructed by extruding a selected cross-section to study. Instead of being positioned around a shaft, they are placed on a flat plane with each Leading Edge (LE) aligned and the blades spaced by the right pitch. The following sections detail the specific case selected, the numerical setup, and the results obtained.

4.1 Case Description

In this study, the blade used for validation is a mid-span section of the PW2037 LPT rotor. Its original application is at a pitch to chord ratio of 0.799 with no inflow incidence. In order to increase blade loading and reduce the weight of the overall stage, an increase of the pitch to chord ratio to 1.05 was applied. Further experiments, focused on flow transitions on the Suction Surface (SS), added an incidence of 5° to the inflow. The reference T106D test case includes both of these modifications. The points used to construct the T106 blade come from the work of Stieger [66].

Figure 4.1 (not to scale) shows the experimental setup used by Stadtmuller when testing the T106D [31]. Tests were conducted over 5 blades at a span to chord ratio of 1.76 and over a range of exit Reynolds number of 60,000 to 200,000. Upstream cylindrical bars were used to simulate stator effects during some tests, and the inflow turbulence intensity was $\sim$1% for the runs without bars. The results at an exit Reynolds number of 80,000 and without the upstream cylindrical bars are used for comparison in this



Cascade Properties	
β_1	42.7°
β_2	26.8°
β_3	59.28°
c	$100mm$
c_{ax}	$85.97mm$
S/c	1.05
h/c	1.76
Ma_1	0.262
Ma_2	0.4
Re_2	$80,000$

Figure 4.1. Stadtmuller experimental setup for the T106D cascade

study. Isentropic Mach numbers were computed all around the middle blade of the experiment, which constitutes the primary set of data used for comparison with the numerical simulation.

4.2 Numerical Setup

With the blade and physical conditions defined, the numerical conditions can be set. In this section, the selected grid is detailed, the boundary and initial conditions are defined, and the technique used to generate proper time-averaged properties is explained.

4.2.1 Grid Generation

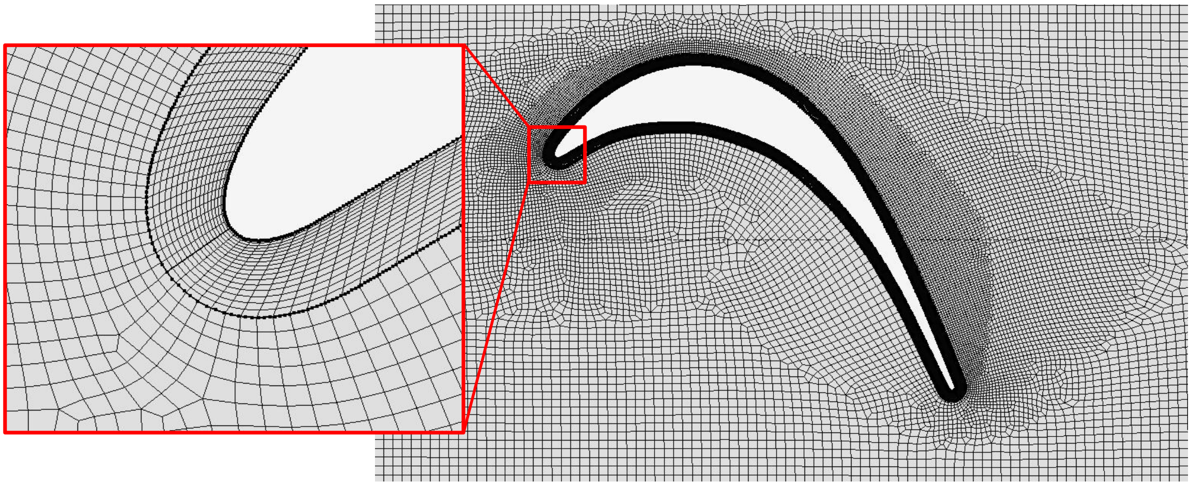


Figure 4.2. Unstructured mesh with zoom on the LE

The unstructured mesh for the baseline T106D, shown in Figure 4.2, is generated using Gmsh [67], and is used to simulate the T106D cascade. It is composed of 292,830 quadratically-curved hexahedral elements, concentrated at the LE and over the SS of the airfoil. The 2D mesh shown in Figure 4.2 is extruded to a span to chord ratio of 0.2, to allow interactions of the vortices in 3D, as is the case in a wind tunnel experiment of a linear cascade. Hexahedral elements are selected for their superior numerical dissipation properties, as previously shown by Pereira and Vermeire [68]. A layer of $0.02c$ in thickness, perpendicular to the airfoil's surface, is used to resolve the boundary layer region. This region is added to ensure proper resolution on the SS, to get accurate separation and transition onsets.

The full domain is delimited by parallel axial planes, spanwise planes and pitchwise planes. The inlet axial plane is located $1.2c$ upstream of the blade's LE, while the outlet axial plane is $4c$ downstream of the LE. The spanwise planes are $0.2c$ apart, and the pitchwise planes are $1.05c$ apart, which represents the pitch of the cascade.

4.2.2 Boundary Conditions

In this study, the boundary conditions are chosen to replicate the conditions found in the wind tunnel experiment. The pitchwise planes are set as periodic boundaries, meaning that the flow exiting one enters the other and vice versa. This has the effect of producing a flow field as if there is a full row of blades, effectively recovering the conditions found in the experiment. Periodic boundary conditions are also set for the spanwise planes, to simulate the 3D interactions occurring in the full linear cascade.

The inlet plane is set as *sub-in-ftpttang* in PyFR, which corresponds to a classic subsonic inflow boundary condition [69]. The inflow conditions are defined from the stagnation temperature, stagnation pressure, flow direction, and fluid properties. In this study, air is the fluid considered, with a Prandtl number of 0.71, and ratio of specific heats of 1.4. Using non-dimensionalized properties, the stagnation pressure is set to 1 and the stagnation temperature multiplied by specific heat at constant pressure (c_pT_0) is set to 3.5. The flow direction is set to 42.7° from the purely axial direction, as for the wind tunnel experiment. Even though a low inflow turbulence intensity was present in the experiment, none is added to the inflow in the numerical simulations.

The outlet plane is set as *sub-out-fp*, which corresponds to a pressure outflow boundary condition [69]. The outflow conditions are defined only by the outlet static pressure and its ratio with the inlet stagnation pressure. This ratio is taken to be 0.8956 for the T106D case. For both the inlet and outlet planes, the flow is assumed to be adiabatic and isentropic through the boundary. Since this outflow boundary can create small spurious reflections back in the domain, a sponge zone is added near the outlet plane. This zone of $0.5c$ in the axial direction contains a coarser mesh, which induces some numerical dissipation to ensure that the reflections do not affect the stability of the simulations, and permitting accurate results in important regions. The downstream portion of the mesh with the sponge zone is illustrated in Figure 4.3.

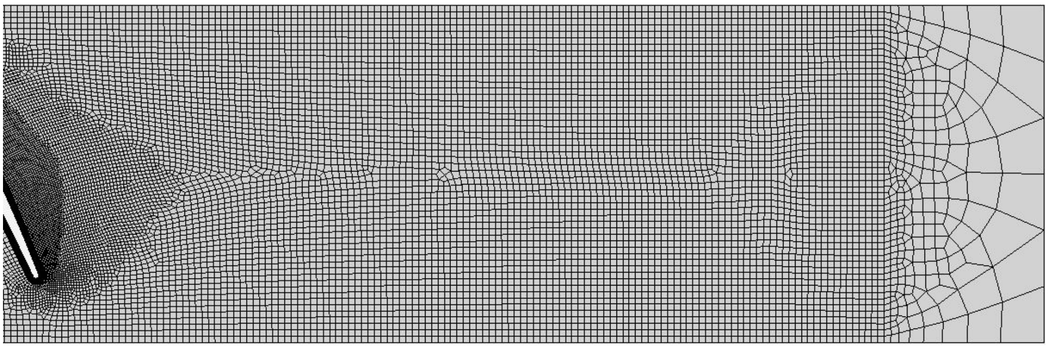


Figure 4.3. Downstream mesh with sponge zone at the outlet boundary

Finally, the blade's surface is set as an adiabatic no-slip wall boundary condition.

4.2.3 Initial Conditions

The inflow and outflow of a turbine cascade are vastly different, because of the turning caused by the blade. Hence, it is necessary to find a process to generate proper initial flow conditions inside the domain, to ensure a stable starting point permitting the advancement of the solution in time. The most conservative and efficient solution is to use a simple initial condition of no fluid motion, and an initial simulation at a polynomial degree of $\mathbb{P}_0$. After the initial $\mathbb{P}_0$ simulation, the polynomial degree is progressively increased to improve resolution. Hence, a simulation at $\mathbb{P}_2$ is restarted from the fully developed flow at $\mathbb{P}_1$, which is itself restarted from the fully developed flow at $\mathbb{P}_0$. The initial flow inside the domain requires a simulation of approximately 28 convective times over the blade at each subsequent polynomial degree. A convective time is defined as the time taken by the initial flow to travel a distance of $1c$.

4.2.4 Time-Averaging

In order to compare isentropic Mach number distributions over the blade's surface, accurate time-averaged properties must be obtained on the surface from the numerical simulations at every polynomial degree tested. First, it is ensured the flow over the blade is fully-developed to make sure the average is not contaminated by initial transient features. This is achieved by letting the flow develop over approximately 21 convective times. Then, the surface properties are averaged for approximately 7 convective times, while the properties of the entire domain are averaged for approximately 21 convective times. In this instance, surface properties refers to the data necessary to calculate the isentropic Mach number distribution around the blade's surface, which typically converges faster than the data in the boundary layer and the wake. The amount of flow passes over a blade necessary to obtain accurate data is not universal and changes from one situation to another. Hence, the amount of convective times necessary to obtain a fully-developed flow and a good time-average were found iteratively at the beginning of this study. Since these values change for different geometries and when different properties of interest are sought, a more elaborate method is employed for each blade during the optimization process, presented in Chapter 6.

4.3 Results

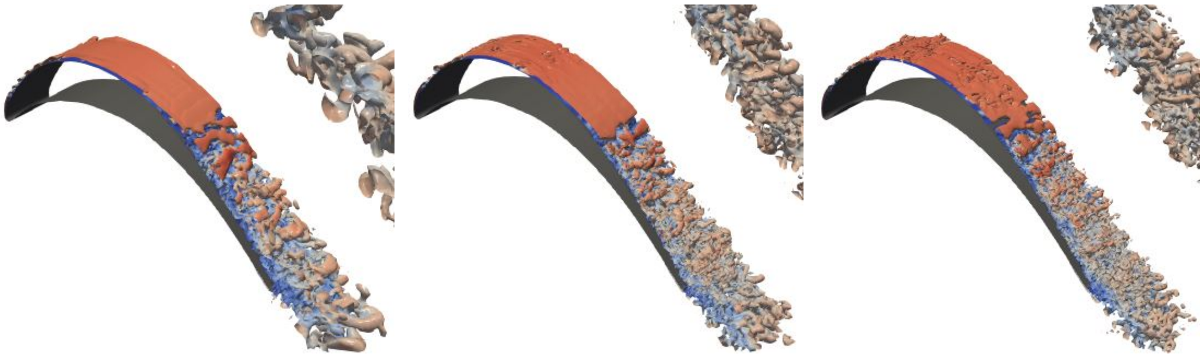


Figure 4.4. Iso-surfaces of q-criterion colored by velocity magnitude using solution polynomials of degree $\mathbb{P}_1$, $\mathbb{P}_2$ and $\mathbb{P}_3$ (left to right)

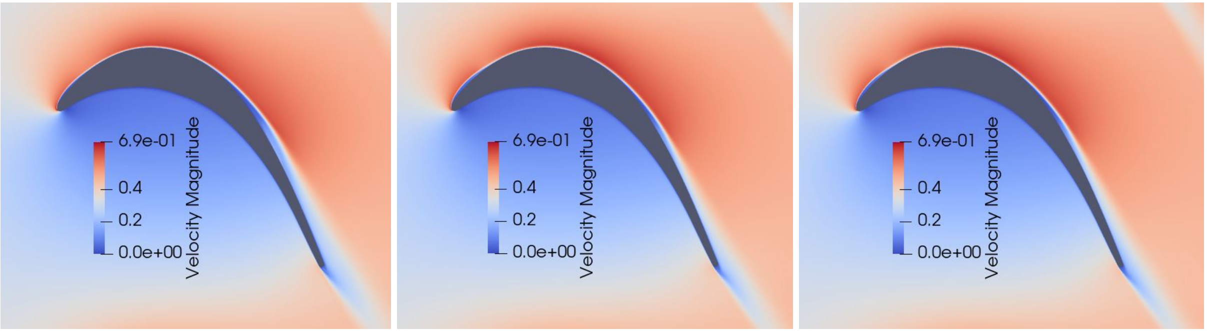


Figure 4.5. Time-averaged velocity magnitude using solution polynomials of degree $\mathbb{P}_1$, $\mathbb{P}_2$ and $\mathbb{P}_3$ (left to right)

Figure 4.4 shows iso-surfaces of q-criterion colored by velocity magnitude for solution polynomials of degree $\mathbb{P}_1$, $\mathbb{P}_2$ and $\mathbb{P}_3$. All simulations show a separation bubble that forms on the SS near the LE of the blade. The results with $\mathbb{P}_1$ show significant dissipation of turbulent structures downstream of the blade, with only large scale structures remaining visible. On the other hand, results with $\mathbb{P}_2$ and $\mathbb{P}_3$ show minimal dissipation of the solution, with the physical scale of turbulent structures remaining relatively constant along the wake. The presence of a separation bubble is also noticeable in Figure 4.5, which shows the time-averaged velocity magnitude for each polynomial degree. The size of the separation bubble is seen to change between $\mathbb{P}_1$ and $\mathbb{P}_2$, but is relatively constant between $\mathbb{P}_2$ and $\mathbb{P}_3$. A slight shift in the transition point aft of the blade between $\mathbb{P}_1$ and $\mathbb{P}_2$ can also be observed. The thickness of the boundary layer is also seen to be decreasing from $\mathbb{P}_1$ to $\mathbb{P}_2$, but relatively constant from $\mathbb{P}_2$ to $\mathbb{P}_3$.

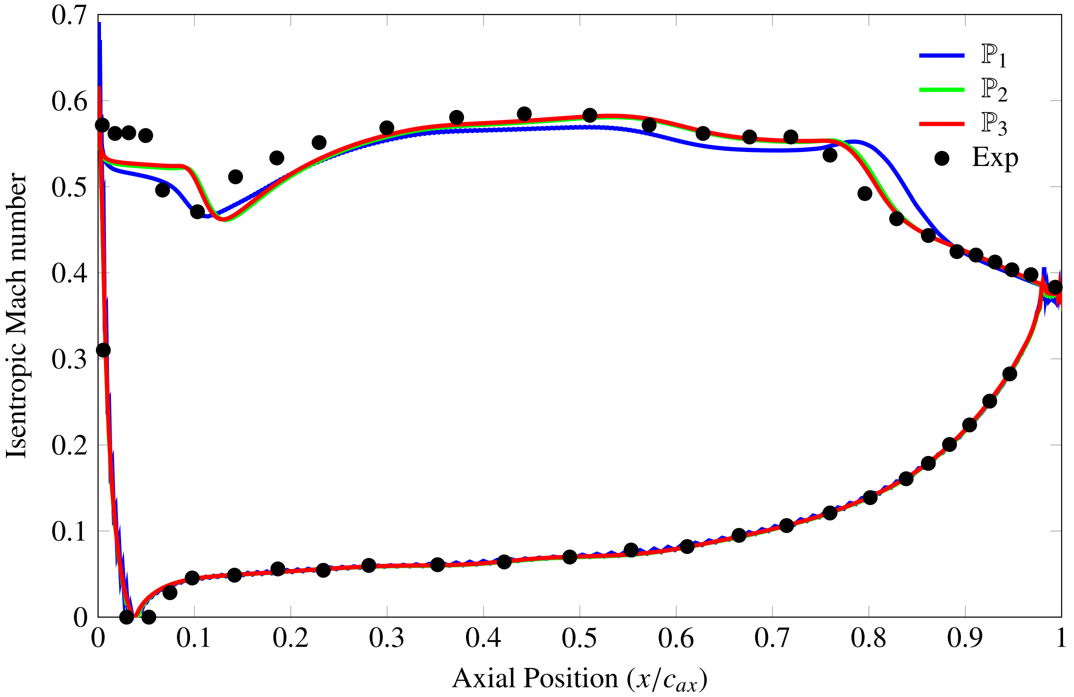


Figure 4.6. Isentropic mach number comparison between polynomials and experimental

Isentropic Mach number distributions for each polynomial degree are shown in Figure 4.6, alongside experimental results from Stadtmuller [31]. These results show two noticeable isentropic Mach number drops where separation occurs on the SS of the blade, namely at $x/c_{ax} \approx 0.1$ and $x/c_{ax} \approx 0.8$. The flow along the Pressure Surface (PS) shows good agreement with the reference results, since the flow remains laminar and attached to the surface. Small discrepancies between the numerical and experimental results are seen at the two separation points on the SS, as the numerical results show the flow remaining attached further along the blade at each point. These are due to the wind tunnel turbulence intensity of about 1%, some mass flow leakage, and small angular discrepancies of the experimental setup [31]. Nevertheless, the current results are generally in good agreement with the reference experimental data.

From these results, it is observed that $\mathbb{P}_2$ and $\mathbb{P}_3$ give nearly identical results, both with good agreement to the reference experimental data. Hence, the subsequent optimization studies will utilize a solution polynomial of degree $\mathbb{P}_2$.

4.4 Comments

The validation exercise was successful in demonstrating that the previously mentioned combination of the selected mesh and the FR method could produce an accurate depiction of a LPT cascade experiment. All validation simulations were conducted on 4x NVIDIA Tesla P100 GPUs on the Deviate cluster at Concordia University, and PyFR version 1.10.0. Running the $\mathbb{P}_2$ simulation for approximately 28 convective times took roughly 8 hours, while 28 convective times of $\mathbb{P}_3$ took around 24 hours to complete. This large difference in computational cost for little to no accuracy improvements is the main reason why the optimization cycles will utilize a polynomial degree $\mathbb{P}_2$ to extract the properties of interest. Hence, every result presented in the following chapters is taken from a simulation using a polynomial degree $\mathbb{P}_2$.

In the following chapter, the methodology employed to conduct optimization cycles will be detailed, along with the selection of design variables, and the presentation of the constrained design space.

Chapter 5

Optimization Methodology

With the numerical methodology defined and validated, the process of optimizing the blade's shape can now commence. In this study the MADS algorithm is employed to conduct a gradient-free optimization of the T106D blade shape, using various objective functions. Since gradient-free methods become computationally expensive when a high number of design variables are used, a methodology with a low number of shape deformation parameters is employed in this work. The following sections detail the MADS algorithm, how the blade is represented and deformed, and the design variables and design space used for all optimization activities.

5.1 MADS

A MADS cycle begins with the evaluation of the objective function at an initial design point, chosen by the designer, which yields the current optimum. Then, design iterations are conducted to look for a new optimum, starting with an initial search step. Each design iteration can be separated in two sequential steps, namely the identification of a finite number of points in the design space, and the evaluation of the objective function at each of these points. If some of the selected points are outside of the defined design space, they are discarded to avoid costly objective function computations [70]. The objective function of each feasible point is then compared to the current optimum to determine if a new optimal point has been found. All trial points lie on a mesh constructed by a finite set of n_D directions, which is scaled by the mesh size parameter at the current iteration Δ_m^k. D is a matrix of directions, which must be a positive spanning set and a nonnegative integer combination of the directions [48][71]. The mesh points M^k are defined by

$$M^k = \bigcup_{S^k \in T^k} \left\{ S^k + \Delta_m^k D z : z \in \mathbb{N}^{n_D} \right\}, \tag{5.1}$$

where T^k is the set of trial points already evaluated before the start of iteration k, and S^k is the current optimal point, also called the frame center. The trial points $\mathcal{P}^k$ are defined by

$$\mathcal{P}^k = \left\{S^k + \Delta_m^k d : d \in D^k\right\} \subset M^k. \tag{5.2}$$

Once the objective function has been evaluated at all trial points, three scenarios are possible, namely a search step, a poll step, or the end of the optimization cycle. If no point yields a new optimum, MADS can either end, if a specified value of Δ_m has been reached, or go into a poll step to reduce the frame size. In this case, the mesh size parameter Δ_m is also reduced faster than the poll size parameter Δ_p, which controls the frame size. To show that MADS creates an asymptotically dense set of poll directions, the House-holder matrix is used

$$H^k = I - 2w^k w^{k,T} \subset \mathbb{R}^{n_s \times n_s}, \tag{5.3}$$

where I, w and n_s are the identity matrix, a normalized random vector and the number of design parameters respectively [53]. To create a poll set, H^k is normalized to the range of the design values by

$$B^k = \{b_1, b_2, ..., b_{n_s}\}, \tag{5.4}$$

$$b_j = round\left(\frac{\Delta_p}{\Delta_m} \frac{h_j}{\max\left(\|h_j\|\right)}\right), \tag{5.5}$$

$$\begin{cases} D^k = B^k \cup \left\{z^{n_s+1} = -\sum_{i=1}^{n_s} z_i\right\}, \\ D^k = B^k \cup (-B), \end{cases} \tag{5.6}$$

and the poll size is defined by

$$\Delta_p^k = \begin{cases} n_s \sqrt{\Delta_m^k} & \geq \Delta_m^k, \\ \sqrt{\Delta_m^k} & \geq \Delta_m^k. \end{cases} \tag{5.7}$$

If a point in the current iteration yields a new optimum, MADS will go into a search step by enlarging the frame size or keeping the same frame size if it was already at the maximum ($\Delta_m^k = \Delta_p^k = 1$). Hence, the possible poll size parameters for the $k + 1$ iteration are defined by

$$\Delta_p^{k+1} = \begin{cases} \frac{\Delta_p^k}{2} & \text{for unsuccessful step } k, \\ 2\Delta_p^k & \text{for successful step } k, \\ \Delta_p^k & \text{for successful step } k \text{ with } \Delta_p^k = 1. \end{cases} \tag{5.8}$$

Following the definition of Equation 5.7, the mesh size parameter for the $k + 1$ iteration is defined by

$$\Delta_m^{k+1} = \begin{cases} \frac{\Delta_m^k}{4} & \text{for unsuccessful step } k, \\ 4\Delta_m^k & \text{for successful step } k, \\ \Delta_m^k & \text{for successful step } k \text{ with } \Delta_m^k = 1. \end{cases} \tag{5.9}$$

Hence, $\Delta_m^k \leq \Delta_p^k$ at all times, and the mesh size allows an increase of the number of polling directions after every successful iteration.

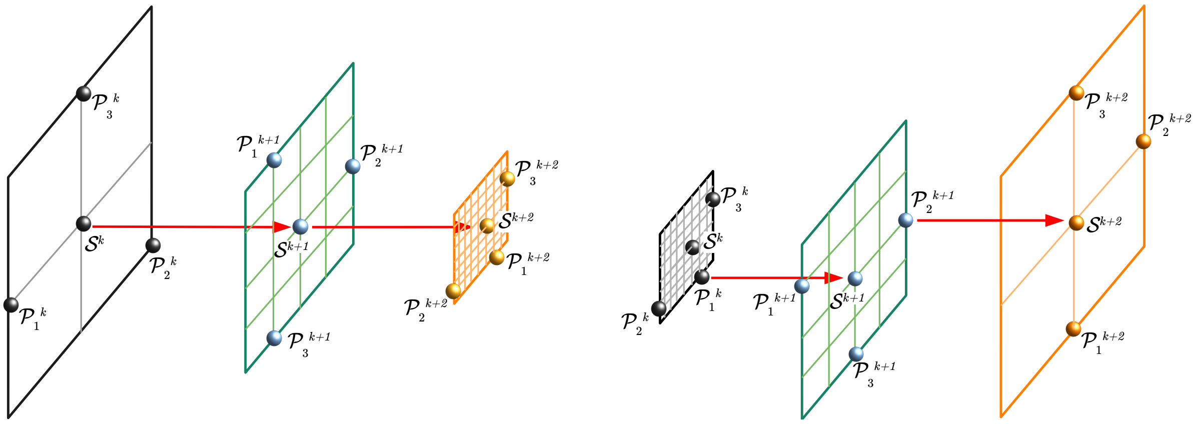


Figure 5.1. Schematic of poll (left) and search (right) steps with MADS

Figure 5.1 illustrates the effects of successive poll steps and the effects of successive search steps on the frame used each iterations. The illustration assumes only 2 design variables are employed in this example in order to facilitate the visualization of the method and its possible progressions. For poll steps, the optimal point remains unchanged for all iterations, meaning that $S^k = S^{k+1} = S^{k+2}$ and the mesh becomes much denser at iteration $k + 2$. For the search steps, it is assumed that P_1^k is an improvement over S^k, meaning that $S^{k+1} = \mathcal{P}_1^k$. Similarly, $\mathcal{P}_2^{k+1}$ is an improvement over S^{k+1}, so $S^{k+2} = \mathcal{P}_2^{k+1}$. It is also assumed that the poll size Δ_p and mesh size Δ_m are not at their maximum for iterations k and $k + 1$ of the search steps. If it wasn't the case, the frame size and mesh size would remain unchanged and only displaced to the new optimal point. Following this approach, MADS is able to efficiently search the design space and converge to the optimal set of design parameters.

5.2 Blade Shape Parameterization

In this section, the techniques employed to represent and modify the blade shape are presented, along with the selected design space and the main aerodynamic objectives of the optimization process.

5.2.1 Blade Shape Representation

Starting from the T106D blade shape, a representation scheme must be generated to enable the creation of a multitude of other possible shapes. Before any decisions are made, care must be taken to take into account every factor that could impact a blade's aerodynamic properties and the optimization process itself. From the validation study presented earlier, flow physics pertaining to this specific case can be better understood, and the shape representation should focus on troublesome areas of the T106D blade. These include separation and transition areas, which are located on the SS of the T106D. Other important factors to consider are the number of design parameters and the flexibility of each parameter to achieve a design space vast enough to encompass a high quantity of possible shapes. With more design parameters, a wider and more precise variety of changes can be applied to the shape, which can enable the optimizer to get closer to the global optimum. On the other hand, employing too many parameters can also overcomplicate the optimization process, a phenomena known as the curse of dimensionality [18]. It is thus paramount to find a good balance between precision, flexibility, and complexity of the shape representation scheme for all blade areas, especially the ones critical to performance.

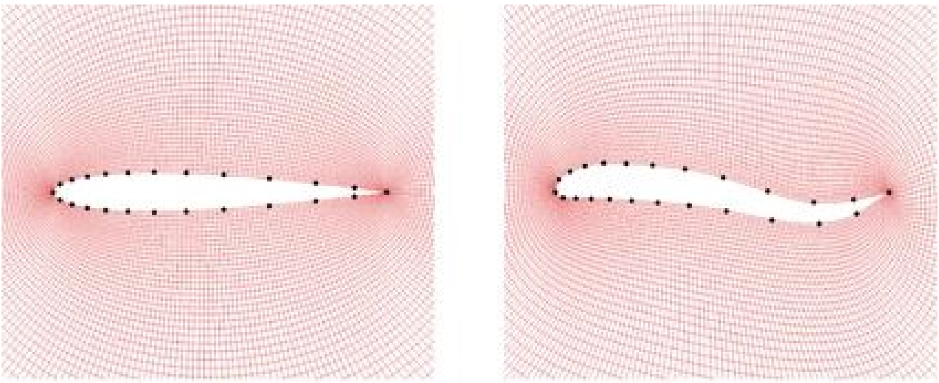


Figure 5.2. Example of typical design variables employed in aerodynamic shape optimization [2]

For the majority of optimizations done using gradient-based methods, the blade is constructed by moving many design points individually to control the physical boundary of the blade. With this representation, a very large amount of design variables must be employed, increasing the complexity of the optimization process. Figure 5.2 [2] illustrates a typical set of design variables used in conjunction with gradient-based optimization methods. Furthermore, employing many design variables with a gradient-based method would increase the computational cost, but significantly less than it would with gradient-free methods, because of their reliance on a high quantity of objective function evaluations. Hence, the need to keep the optimization process at a reasonable computational cost and complexity drives the decision to employ a different blade representation and deformation approach than what is typically used.

In this study, the camber line of the original T106D is used as the benchmark to which deformations are applied. The original camber line is modified by adding a Bézier curve to it, which creates the new camber line. The modifications are applied to the surface points perpendicular to the original camber line. Since the problematic flow phenomena occur on the SS of the T106D, the modifications are mainly targeted at this surface, but not necessarily the PS. To separate the modifications applied to each surface, an additional parameter is used to relate the motion of the PS with respect to the SS. From the initial blade surface, the Bézier curve, and the additional parameter, the new blades can be constructed efficiently and rapidly for every design iterations. The following sections expand on the Bézier curve and the design variables selected for this study.

5.2.2 Bézier Curve

The curve used to incur deformations to the original camber line is a cubic Bézier curve, which is a third-order non-rational spline controlled by 4 points [72]. A Bézier curve uses Bernstein polynomials as basis, is smooth, and can be efficiently manipulated with just a few control points. These properties make Bézier curves attractive to apply shape modifications using a low number of design parameters, while ensuring smooth deformations. The definition of a Bézier curve of order n is

$$x(s) = \sum_{i=0}^{n} \frac{n!}{i!(n-1)!} s^i (1-s)^{n-i} x_i, \qquad y(s) = \sum_{i=0}^{n} \frac{n!}{i!(n-1)!} s^i (1-s)^{n-i} y_i, \tag{5.10}$$

where (x,y) are the coordinates of the points constituting the curve, (x_i,y_i) are the coordinates of the control points and $s \in [0,1]$ [73]. The initial curve is a straight line from the LE to the TE, with all control points lying on the curve. Figure 5.3 illustrates the original camber line in black, the unperturbed Bézier curve in red, and the 4 control points in their initial positions in black.

Each point on the Bézier curve is associated vertically with its corresponding point on the camber line. Perturbations to the initial Bézier curve generated by the movements of the control points are then added to each original camber line location to create the new camber line. Figure 5.4 illustrates an example of modified camber line and Bézier curve achieved by moving the second control point up. The original curves, represented by dashed lines, are shown as reference to better illustrate the relative motion of both curves. The green points and lines are shown as an example of how each point on the curves are associated and moved.

Since only a few control points are employed in this study, points 2, 3 and 4 are allowed to move

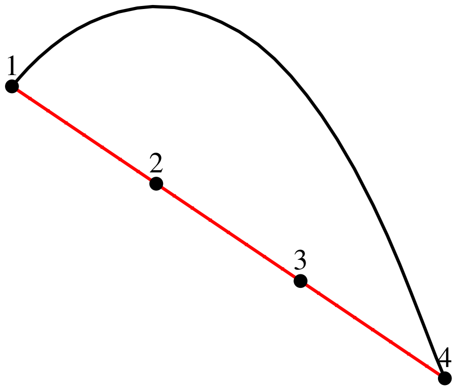


Figure 5.3. T106D camber line and Bézier curve

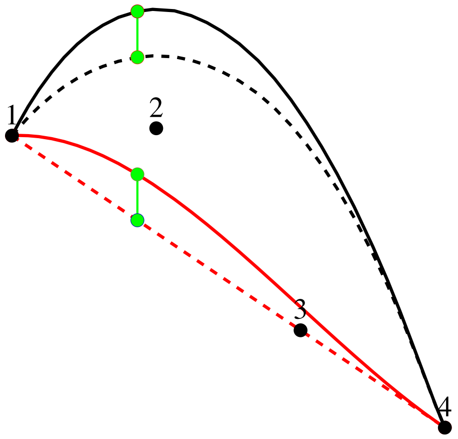


Figure 5.4. Vertically modified camber line and Bézier curve

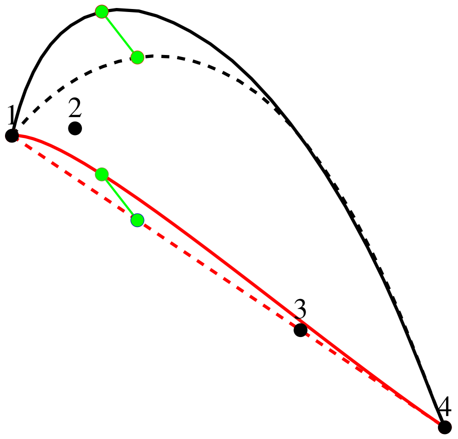


Figure 5.5. Horizontally and vertically modified camber line and Bézier curve

horizontally and vertically. The LE point is used as an anchor point for the blade and is never allowed to move. Thus, displacing the camber line horizontally is possible, as shown in Figure 5.5, where the second control point is moved up and to the left.

From the deformed camber line, the full blade can be reconstructed by moving the surface points by the same amount as their corresponding camber line points were moved. Figure 5.6 illustrates the new blade formed by the camber line modifications shown in Figure 5.5, compared with the original T106D.

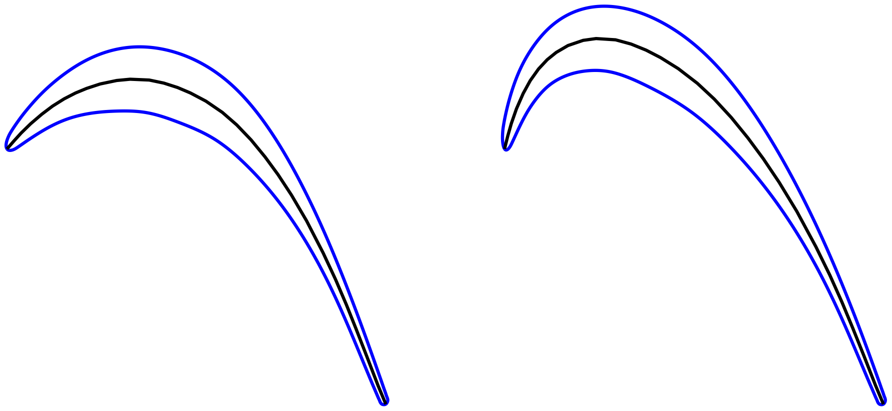


Figure 5.6. T106D (left) compared with example of reconstructed blade (right)

5.2.3 Design Variables

The choice of design variables has a great impact on the optimization problem and its outcome, and it is especially true in this study. Since a gradient-free method is employed to conduct the optimization, keeping the number of design variables low is of great importance, because it would enable a more manageable computational cost for the full optimization process. The design variables employed in this study are primarily the Bézier curve control points. The most important control points are number 2 and 3, which can be freely moved horizontally and vertically. As mentioned previously, the LE control point is fixed and used as an anchor, which also helps in maintaining the integrity and smoothness of the LE surface. Also, only the vertical position of the TE control point is used as a design variable, and its horizontal position is calculated to maintain a constant chord length. This ensures that the new blades would be able to fit relatively easily on the same rotor without interfering with neighboring stages or leave large gaps between stages. This also implies that the Reynolds number remains almost constant for each blade iterations. With the other 2 control points being able to move freely in all directions, there is a total of 5 design variables controlling the camber line profile. Changes to the camber line were applied directly to the SS, since this is where the majority of the turbulent separation occurs for the T106D case

[74]. An additional design parameter was added to adjust the deformation of the PS with respect to the SS. This parameter could take values in the range [0, 1], with 0 signifying no deformation of the PS and 1 signifying equal deformation of the PS and SS. Examples of the effect of this additional design variable are shown in the following section. With these 6 design variables at hand, a wide range of candidate blade shapes can be studied, and this wide range is illustrated and explained next.

5.2.4 Design Space

With the 6 design variables defined, the range of each must be determined. These ranges must be restricted so that the possible blades remain in a feasible structural and manufacturing space, while also being wide enough to allow for sufficient aerodynamic performance exploration. For example, a large upward motion of the second control point would create blade platform interference, meaning that they cannot fit onto the rotor by purely sliding them on. While feasible, this would increase the manufacturing cost and complexity [16], hence why it is decided to be off-limits. Another possible undesirable effect of the design variables would be to generate a portion of blade that would be of similar or lower thickness to the TE, which would be structurally unsound. Therefore, limits are applied to certain parameters to ensure these physical issues do not occur for any of the tested blades.

The range of some design variables can also be linked to the direction in which improvements are most likely. Since the T106D operates at a positive incidence of 5°, the optimization will most likely try bring the LE metal angle closer to the inlet flow angle, which translates to the second control point moving to the left. Hence, it would be desirable to allow for a large range in this direction for this design variable, but the range in the opposite direction could be quite small or nonexistent. In addition to setting constraints based on theoretical principles, a few tests were run to aid in setting ideal boundaries. The tests consisted of starting a MADS optimization with very loose constraints using only 2D simulations to evaluate the objective functions in order to accelerate the process. After a few MADS cycles, it was stopped and the process was restarted from the original design point a few times to observe the evolutionary tendencies of the algorithm. If it was noticed that a variable should have its design space limited or extended in a specific direction, these changes were applied before commencing aerodynamic design. Table 5.7 details the specific range used for each design variable throughout this study.

Figure 5.8 illustrates some of the blades located on the edges of the design space. The effects of the PS to SS ratio design variable can be better seen when comparing example blades *a* and *b*, where the former has a ratio of 1 and the latter a ratio of 0. Even though both blades have the same values for the first 5 design variables, changing the PS to SS ratio creates blades of drastically different maximum

#	Design variables	Minimum	Maximum
X_1	TE vertical	-0.01	0.1
X_2	Pt 2 horizontal	-0.18	0.05
X_3	Pt 2 vertical	-0.1	0.25
X_4	Pt 3 horizontal	-0.11	0.13
X_5	Pt 3 vertical	-0.13	0.13
X_6	PS to SS ratio	0	1

Figure 5.7. Design space limits

thickness. The same observation can be made for blades *c* and *d*, where only the PS to SS ratio differs. Blade *e* shows the thinnest possible shape that can be achieved within the design space, which would be structurally undesirable. All 8 blade examples show the wide variety of shapes that can be obtained using only 6 design variables. The extremes of the design space are never converged to during any of the optimization cycles presented in this study, signifying that the design space is large enough and the optimization is not overly constrained.

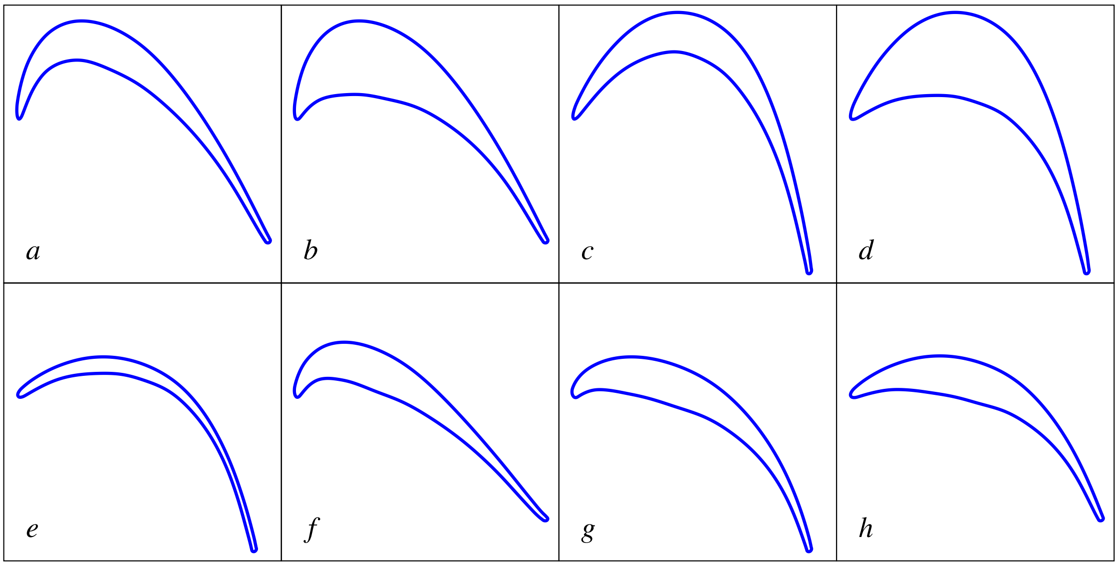


Figure 5.8. Blade shape examples

5.3 Design Objectives

In this study, two performance parameters are used to build the objective functions that guide the optimization process. The first one is the total pressure loss coefficient, which is an indicator of the amount of stagnation pressure lost because of entropy generation through a turbomachine. A low value of this performance parameter is sought during aerodynamic design, since it correlates to a high isentropic efficiency for the stator or rotor evaluated [18]. The total pressure loss coefficient is calculated from

$$Y = \int_{out} \frac{p_{01} - p_{02}}{p_{02} - p_2} dA, \tag{5.11}$$

where p_0 is the stagnation pressure at the inlet (1) and outlet (2), and p_2 is the static pressure at the outlet [17]. For all blade iterations, p_{01} corresponds to the input value of 1, in non-dimensionalized units, at the inlet boundary. The outlet static pressure is taken directly from the time-averaged results, and the outlet stagnation pressure is calculated from

$$p_{02} = p_2 + \frac{1}{2}\rho_2 v_2^2, \tag{5.12}$$

where ρ_2 is the time-averaged static density at the outlet, and v_2 is the time-averaged velocity magnitude at the outlet. The outlet values are all taken $1c$ downstream of the blade's TE.

While typical designs have long focused on reducing the total pressure loss coefficient, the focus has recently shifted towards reducing weight, which translates to increasing the blade loading in order to necessitate fewer and shorter blades and fewer blade rows, while still generating equivalent total power [17]. Increasing blade loading can be achieved by increasing the Zweifel coefficient [13] or, more generally, by increasing the tangential force produced by the blade [75]. In this study, the second performance parameter used is the tangential force produced by the blade. This parameter offers a measure of the amount of effective force produced by the blade that will be used to spin the rotor on which the blade sits. The tangential force is formulated as

$$f_t = p_y + v_y, \tag{5.13}$$

where p_y is the summation of all pressure forces acting vertically on the blade's surface, and v_y is the summation of all viscous forces acting vertically on the blade's surface. Both p_y and v_y are collected periodically on the blade's surface using the *fluidforce* plugin offered in PyFR, and averaged over the same time period as the total pressure loss coefficient.

Each objective function used in this study contains the main objective of the optimization and a penalty term, a similar approach to what has been used in previous studies [40]. The objective functions are formulated as

$$\mathcal{I} = \min_{X \in DS} \left[Obj + PT \right], \tag{5.14}$$

where X is the vector of design variables, DS is the design space previously defined, Obj is either Y

or $-f_t$, and PT is the penalty term. This penalty term is a combination of one or more constraints that are imposed on the design to ensure some properties of interest remain at, above or below a certain threshold. A vast amount of constraints can be applied, such as structural, geometrical or aerodynamical constraints. In this study, only aerodynamical constraints are used to construct the penalty term and constrain the design. The full objective function employed for each optimization process is provided in their corresponding sections of Chapter 6.

Chapter 6

Constrained Optimization

With the optimization methodology defined and the numerical methods validated, the results of the optimization process can now be obtained and discussed. As mentioned previously, a preliminary optimization is performed using 2D simulations for objective function evaluations. The goal of this first activity is to provide accurate bounds for the design space, and provide better insight into the time-averaging requirements for varying shapes. Then, a few preliminary 3D flow simulations of a wide variety of blades are conducted to assess the validity of the time-averaging requirements and verify if the procedure is to be changed from 2D to 3D. Once the bounds and requirements are set from preliminary trials, a full optimization is conducted with 2D simulations. This initial design process will provide a better overview of the direction of optimal design, an estimate of the number of blade iterations necessary for MADS to converge, and initial results that could impact how the next designs could be conducted to minimize computational costs. Then, aerodynamic design is performed using 3D simulations with a span to chord ratio of 0.2, starting with the T106D as initial blade. Two separate optimizations are conducted using different combinations of the design objectives to construct each objective function. The main goal of these design activities is to prove that optimization using 3D LES can be achieved at a reasonable computational cost, to entice further research in this field. The following sections include a presentation of the time-averaging requirements, as well as the description, specifications and results for each aerodynamic design conducted.

6.1 Time-Averaging

Scale-resolving simulation properties need to be time-averaged properly to generate adequate objective function evaluations. Extremely long time-averaging periods would ensure precise and accurate time-averaged properties, but their computational cost would be prohibitively expensive. Hence, a suitable strategy must be defined and validated to ensure sufficient precision and accuracy of the flow properties can be obtained at a reasonable computational cost. To do so, the evaluation of solution convergence is provided by the combination of a probabilistic approach [76] and a running average [77]. This methodology implies that short segments of time-averaged properties are output by the simulation periodically, and statistics are computed on these segments. As the simulation progresses, segments are added and statistics are recalculated, which enables to computation of a refined confidence interval every time a new segment is output. Assuming a Gausian probability distribution, the 95% confidence interval is constructed from

$$C = \frac{z^* s_d}{\sqrt{N_s}}, \tag{6.1}$$

where z^* is a standardized variable with a value of 1.96 for a 95% interval, N_s is the number of sampled segments, and s_d is the sample standard deviation calculated from

$$s_d = \sqrt{\frac{1}{N_s - 1} \sum_{i=1}^{N_s} (u_i - \bar{u})^2}, \tag{6.2}$$

where u_i is the property of interest from each sampled segment, and $\bar{u}$ is their average [76]. In this study, the property of interest is chosen to be the total pressure loss coefficient, since it takes longer to converge than the tangential force. The target confidence interval is chosen to be ±0.001 total pressure loss coefficient, since this is an order of magnitude smaller than the possible improvements observed in the 2D preliminary trials. To decide on the size of each segment, an iterative approach was used. In this approach, multiple blades, ranging from low to very high total pressure loss coefficient, are simulated in 3D using varying segment's length to see what time length enables the total pressure loss coefficient to converge to the desired interval the fastest. From these preliminary simulations, it is also observed that initial transient features reduce the quality of the average at the beginning of each simulation. This is because the initial flow in the domain is obtained using a simulation at lower fidelity, meaning that some time needs to pass before the properties captured are truly at high fidelity. After the preliminary trials, a procedure to assess initial stability and commence the running average is defined, and the segment's length is set. The resulting procedure is created for objective function evaluations using 3D simulations, since this is the true objective of this study. For the design using 2D simulations, a slightly different process is needed to obtain accurate time-averages, and the changes are detailed in Section 6.2. The following section provides a detailed rundown of the time-averaging procedure, with the sizes and lengths set from the preliminary trials.

6.1.1 Procedure

Time-averaged flow quantities are output every ~7 convective times, and the total pressure loss coefficient is evaluated for each of these segments. First, it is ensured the flow over the blade is fully-developed to make sure the running average is not contaminated by initial transient features. This is assessed by requiring three consecutive segment averages to be within a 0.006 total pressure loss coefficient range, ensuring the flow has stabilized. Then, time-averaged statistics are collected over a minimum of 6 segments. After the sixth segment, a 95% confidence interval is computed from the segments' standard deviation. If the interval is narrower than ±0.001 total pressure loss coefficient, the simulation is stopped. If not, the simulation is continued for as many segments as necessary in order to converge to within this criteria. For most blades tested, stability is attained in 3 to 5 segments and convergence in 6 to 9 segments. Limits of 6 stability segments and 12 convergence segments are applied to provide bounds for total computational cost.

6.2 2D Case

In this section, the initial optimization using 2D simulations for objective function evaluations is presented. This case comes after the preliminary trials that were conducted to assess the time-averaging requirements and design space. Since it uses 2D flow simulations, the computational cost of each blade iteration is far inferior to the validation case using a 3D simulation of the T106D reported in Chapter 4. Hence, this case represents a good opportunity to obtain a preliminary design, while ensuring good functioning of the coded MADS framework and the time-averaging procedure.

6.2.1 Case Description

As for all optimization activities conducted, the baseline design is the T106D. Due to the fact that 2D simulations are used to provide the necessary results of this design process, the time-averaging procedure and the mesh are modified to account for the change from 3D to 2D. Since this case is simulating the flow strictly in 2D, the natural dissipation that should occur from the breakdown of flow structures from 3D interactions cannot happen. Hence, the high levels of turbulent kinetic energy that are generated along the SS of each blade remain relatively high in the wake. This yields higher fluctuations of unsteady properties along the plane where outlet flow values are time-averaged. In turn, this creates noisier time-averages than what is observed with 3D flow simulations. Hence, a modified time-averaging process is used for this case. Instead of using an average of the total pressure loss coefficient at $1c$ downstream of each blade's TE, the properties are averaged between $1c$ and $5c$ downstream of the blades. This modification helps to damp the high fluctuations observed between each segment's total pressure loss coefficient. This also implies that the simulation domain must be extended from $4c$ to $8c$ downstream, to allow sufficient room for the volume average, and enough distance between the sponge zone and the averaging zone. The modified domain and sponge zone are illustrated in Figure 6.1.

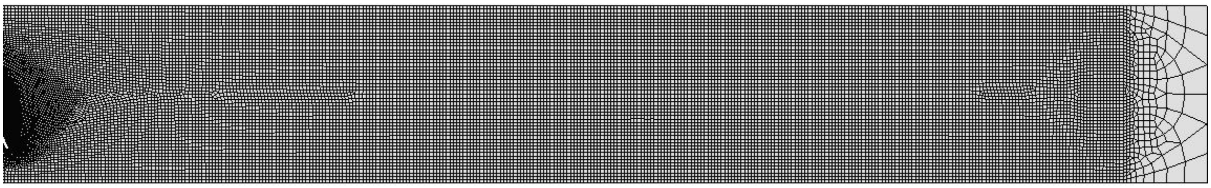


Figure 6.1. Modified downstream mesh with sponge zone at the outlet boundary

Since this case uses 2D simulations, the mesh detailed in Chapter 4 is not extruded, and is extended as mentioned above. This generates a 2D mesh of 39,699 quadratically-curved quadrilateral elements with the same distribution of elements around the LE and SS of the blade as previously detailed. Aside from the extension of the outlet boundary further downstream, the domain is the same as it was for the validation case presented. The boundary conditions and procedure to generate the initial low-fidelity conditions inside the domain are also the same as those presented in Chapter 4. All blade iterations are simulated on 4x NVIDIA Tesla P100 GPUs on the Deviate cluster at Concordia University, and PyFR version 1.10.0.

6.2.2 Case Objective

The objective function seeks to decrease the total pressure loss coefficient, while keeping at least the same tangential force as the baseline design. Based on the previously computed properties of the T106D blade, the objective function is formulated as

$$\mathcal{I}_1 = \min_{X \in DS} \begin{cases} Y + \alpha \left(\frac{f_{t,0} - f_t}{f_{t,0}} \right)^2 & f_t < f_{t,0}, \\ Y & f_t \geq f_{t,0}, \end{cases} \tag{6.3}$$

where f_t is the tangential force generated by the blade, $f_{t,0}$ the tangential force generated by the original blade, α is a penalty weight factor chosen to be 2, and Y is the total pressure loss coefficient. This case uses a comparison of tangential force between the current blade iteration and the T106D to construct the penalty term. A blade producing more tangential force is not penalized, while the opposite is penalized to ensure the optimization does not follow a direction where the rotor would not be able to drive the compressor sufficiently. A 2D simulation of the T106D is conducted beforehand to adjust the penalty term, more specifically the reference tangential force used for comparison.

6.2.3 Optimization Specifications

For this initial aerodynamic design, no additional specifications are implemented to accelerate each objective function evaluation, and reduce computational costs. Hence, the MADS algorithm is allowed to explore the full design space, and uses the previously defined time-averaging procedure for all blade iterations. The criteria to stop MADS is linked to the mesh size parameter of the algorithm. The initial cycle uses $\Delta_p = \Delta_m = 1$, and the optimization continues until the residual limit is passed. The residual limit is chosen to be 0.01, so that the optimization is stopped immediately when $\Delta_m < 0.01$ at the beginning of a MADS cycle.

6.2.4 Results

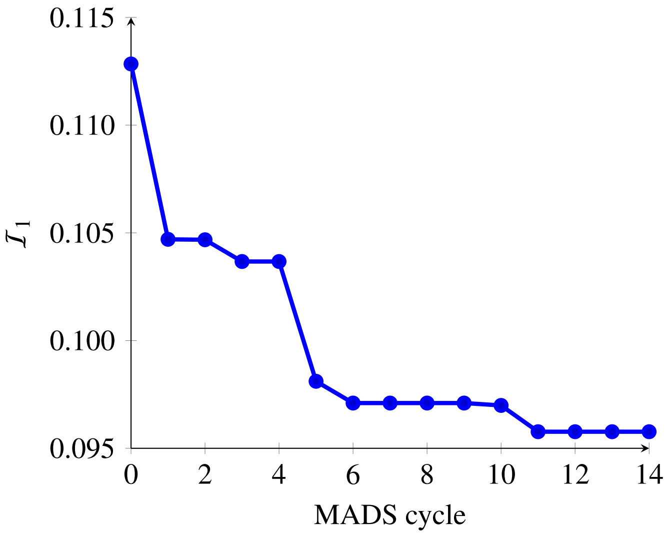


Figure 6.2. 2D objective function progression per MADS cycles

The MADS optimization using objective function $\mathcal{I}_1$, for total pressure loss coefficient reduction, was completed in 14 design cycles, as shown in Figure 6.2. The optimization decreased the objective function by ~15%, compared to the original blade. The optimized blade generates slightly more tangential force, compared to the original T106D, but a ~15% smaller total pressure loss coefficient. The majority of improvements came in the first 6 design cycles, and the design variable progression through the cycles

Cycle	X_1	X_2	X_3	X_4	X_5	X_6
0	0	0	0	0	0	0.5
1	0	-0.16	0	0	0	0.5
2	0.05	-0.16	0	0.07	-0.03	0.85
3	0.05	-0.16	0.18	0.07	-0.03	0.85
4	0.05	-0.16	0.18	0.07	-0.03	0.85
5	0.05	-0.16	0.18	0.07	-0.015	0.85
6	0.05	-0.16	0.18	0.07	0.015	0.85
7	0.05	-0.16	0.18	0.07	0.015	0.85
8	0.05	-0.16	0.18	0.07	0.015	0.85
9	0.05	-0.16	0.18	0.07	0.015	0.85
10	0.04766	-0.165	0.18	0.06563	0.01359	0.89375
11	0.04766	-0.165	0.19125	0.07	0.00609	0.9375
12	0.04766	-0.165	0.19125	0.07	0.00609	0.9375
13	0.04766	-0.165	0.19125	0.07	0.00609	0.9375
14	0.04766	-0.165	0.19125	0.07	0.00609	0.9375

Table 6.1. Evolution of shape deformation parameters for the 2D optimization

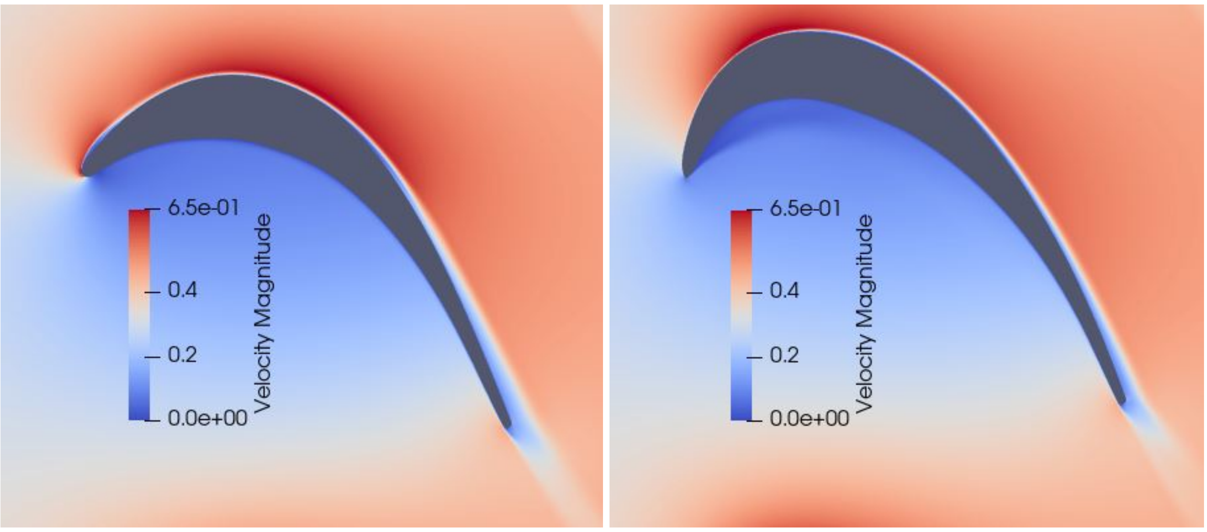


Figure 6.3. Averaged velocity magnitude comparison between T106D (left) and 2D optimization (right)

can be observed in Table 6.1. There are 2 large improvements in those first few design cycles which can be attributed to changes to X_2 and X_5. A time-averaged velocity magnitude comparison, shown in Figure 6.3, illustrates the main flow modifications introduced by the optimized shape. The primary difference is the removal of the LE separation bubble for the optimized blade, which is in large part caused by the design changes to X_2 and X_3. These parameters are responsible for the motion of the left portion of the blade, which was expanded upward and to the left in order to match the incoming flow, which is at positive incidence for the T106D. Another major change is the decrease in boundary layer thickness on the SS as the flow approaches the TE of the optimized blade. This last modification originates from the adjustments to X_1, X_4 and X_5, changing the TE to more accurately guide the outlet flow to minimize deviation. Finally, X_6 enabled the PS to move almost exactly like the SS, which created an optimized blade only slightly thicker than the original. It should be noted that this large value of X_6 seems to have generated a stagnation region near the LE on the PS. The design process using 3D simulations should give better insight as to whether this flow phenomenon greatly affects the total pressure loss coefficient or not.

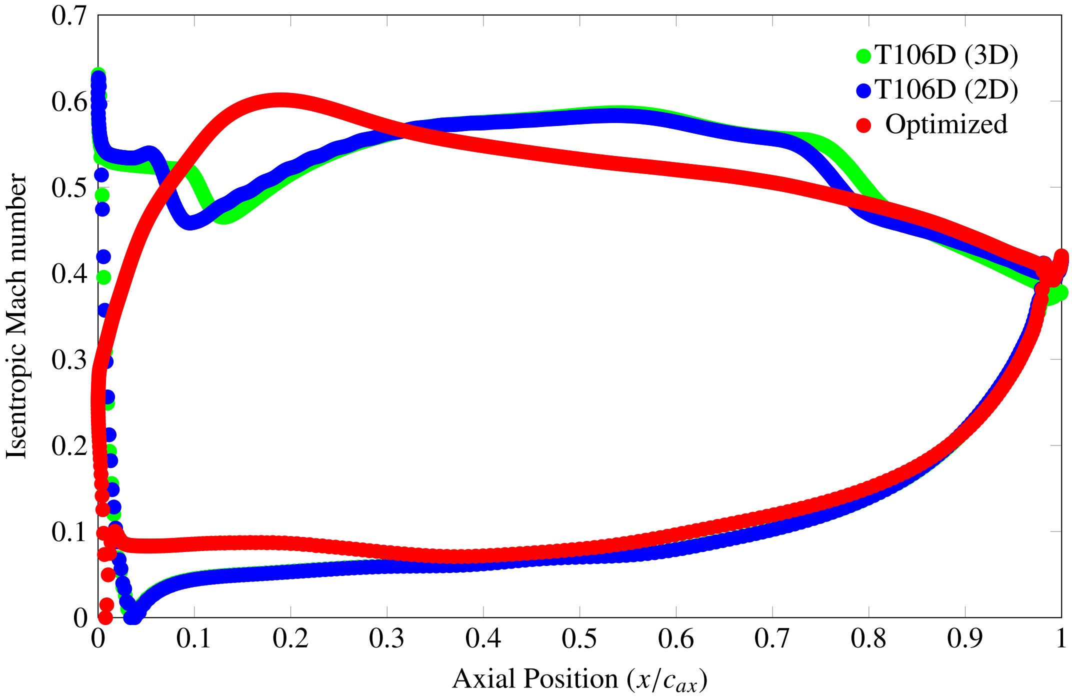


Figure 6.4. Isentropic mach number comparison after 2D optimization

Figure 6.4 illustrates the isentropic Mach number distribution for the baseline blade in 2D and 3D, and the optimized blade. The T106D results using a 3D simulation are only shown to highlight how the dimensionality of the flow solution can affect how the separation and transition onsets are captured, and prove the need for 3D simulations in the design process even for a cross-section optimization. The sharp drop along the SS at $x/c_{ax} \approx 0.1$ for the T106D occurs due to transition caused by the LE separation bubble, which is not present for the optimized blade. Instead, the first portion of the SS of the optimized blade creates a constant and smooth velocity increase up to the suction peak around $x/c_{ax} \approx 0.2$, followed by a constant and smooth velocity decrease down to the TE. The rapid velocity drop at $x/c_{ax} \approx 0.8$ for the T106D is where the majority of turbulent kinetic energy is generated, and the substitution of it by a smooth velocity gradient for the optimized blade greatly reduces the entropy generated. Coupling these observations highlights how the optimization using MADS brought the design towards smoother and attached flow along the SS of the optimized blade.

Only minor differences are present for the isentropic Mach number distribution on the PS of the optimized blade, since the flow was already attached and laminar on the PS of the T106D. The changes to the PS are observed close to the LE, where the flow is allowed to continue its tangential motion up to $x/c_{ax} \approx 0.25$ because of the vastly different shape of the PS at the LE. Adjustments made by the optimization process also altered the position of the stagnation point, represented by the position where the isentropic Mach number reaches 0 for each blade. The initial T106D being at a positive incidence of 5 degrees meant that the stagnation point was not directly at the LE of the blade. The shape modifications effectively changed the inlet metal angle of the blade to more closely match the incoming flow angle, which also repositioned the stagnation point at the blade's LE. This new stagnation point and the shape of the PS allow a sharp increase in flow velocity immediately after the LE, which is not the case for the T106D with its stagnation point on the PS. These changes suggest that the separation bubble was mainly removed by reducing the positive incidence observed by the blade, without changing the inlet flow angle.

6.2.5 Discussion

The initial optimization process was successful in finding multiple blades that could drastically improve the objective function of choice. The optimal design achieved its goal of significantly reducing the total pressure loss coefficient, compared to the T106D, while still generating the same amount of tangential force. On the Deviate cluster, running the simulations with polynomial degrees of $\mathbb{P}_0$ and $\mathbb{P}_1$ to generate proper initial conditions inside the domain took roughly 40 minutes combined for each blade. For the simulations with $\mathbb{P}_2$, each iteration took roughly 70 minutes to stabilize, and 140 minutes to produce an adequate time-average. In total, 90 blades were tested, including the T106D, and the combined simulation time was 365 hours. Hence, the computational resources needed for this design amounted to 1,460 GPU-hours, or 0.17 GPU-year.

Since the Deviate cluster is composed of exactly 4 GPUs, each blade had to be run one after the other, meaning that the design took 365 hours, or 15 days, to complete. In future optimization cases, Compute Canada's Beluga cluster will be used, enabling every blade iterations in each MADS cycle to be computed simultaneously, in parallel. This change to a much larger cluster should decrease the amount of wall-clock time necessary to conduct a full aerodynamic design. In addition to more GPUs and parallel computing, insights gathered throughout this initial optimization case can enable faster evaluation of each blade. Even though the first 6 segments to compute the time-average might not be enough to achieve the required precision, they provide a good outlook at the total pressure loss coefficient that could be expected from the complete time-averaging. Hence, the next cases will make use of this information to prevent unnecessary computations of blades that are clearly above a certain threshold of total pressure loss coefficient.

In the following section, the first aerodynamic design utilizing 3D simulations for objective function evaluations is presented in details.

6.3 First 3D Case

In this section, the first optimization using 3D simulations for objective function evaluations is presented. This case comes after the initial 2D optimization that was conducted successfully and provided insights into ways to diminish the computational cost associated with each blade iteration. Since this case uses 3D flow simulations, the computational cost of each blade iteration is far superior to the initial 2D case, and will require additional resources. This case represents the first opportunity to evaluate whether the computational cost associated with gradient-free optimization using 3D scale-resolving flow simulations is prohibitive or not.

6.3.1 Case Description

As previously defined, the baseline design is the T106D blade. The mesh, boundary conditions, domain, and procedure to generate the initial low-fidelity conditions inside the domain are the same as those presented in Chapter 4. The use of 3D simulations also means that the total pressure loss coefficient data is taken $1c$ downstream of the blade and that the domain does not need to be extended like it was in the 2D case. PyFR version 1.10.0 is again employed to conduct each simulation.

While using 3D flow simulations to evaluate the objective function will lead to more accuracy and lower levels of fluctuation, it will also require more computational resources to manage the large memory requirements of larger simulations. Hence, the Beluga cluster from Compute Canada, composed of 172 nodes each containing 4x NVIDIA V100 SXM2 GPUs and large storage capabilities [78], is used to conduct the flow simulations. To get the number of nodes that would produce the ideal speedup to conduct each simulation, a strong scaling study was performed, of which the results are shown in Figure 6.5. From this study, the number of nodes that can achieve a high speedup that is still close to the ideal speedup is selected to conduct each blade iteration. In this case, 5 nodes will be used for each blade, meaning that all blade iterations are simulated on 20x NVIDIA V100 SXM2 GPUs.

It should be noted that the combination of different nodes produced different speedups, which implies that the communication between nodes is not always as efficient. This effect is enhanced as the number of nodes is increased, since the simulation time becomes more dependent on the communication between

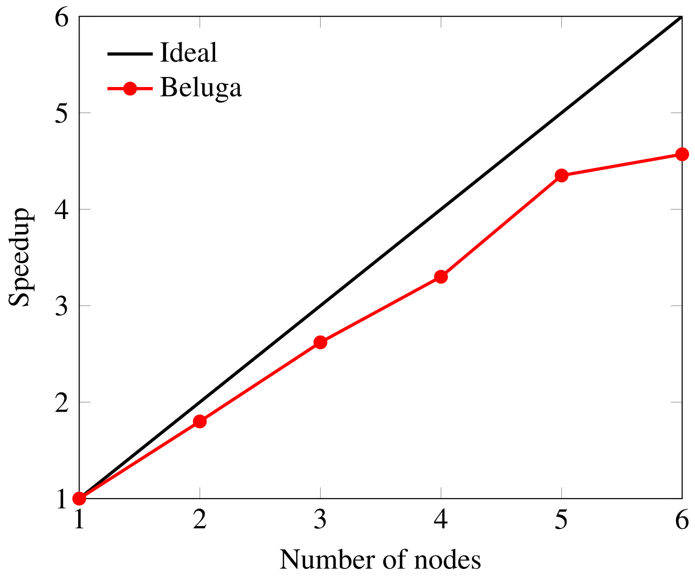


Figure 6.5. Strong scaling study on Compute Canada's Beluga cluster

nodes. The strong scaling presented here is the best possible speedup observed for each number of nodes, but not necessarily what will be achieved each simulation. Ideally, the speedup obtained would follow the number of nodes linearly, but the communication capabilities between nodes and the number of operations per node reduce the speedup observed in practice. Since the number of point-wise direct operations increases exponentially with the polynomial degree, employing a polynomial degree higher than $\mathbb{P}_2$ would be more likely to achieve the ideal speedup [27], which could be considered in future experiments. For the current study, the observed speedup of approximately 4.35 with 5 nodes is considered sufficient. Additionally, a V100 SXM2 GPU is a newer version that can be up to two times faster than a P100 GPU [79, 80], which implies that the observed speedup from 4 GPUs on Deviate to 20 GPUs on Beluga will be higher than the one provided by the number of nodes only.

6.3.2 Case Objective

The objective function seeks to decrease the total pressure loss coefficient, while keeping at least the same tangential force as the baseline design. It is formulated as

$$\mathcal{I}_1 = \min_{X \in DS} \begin{cases} Y + \alpha \left(\frac{f_{t,0} - f_t}{f_{t,0}}\right)^2 & f_t < f_{t,0}, \\ Y & f_t \geq f_{t,0}, \end{cases} \tag{6.4}$$

where f_t is the tangential force generated by the blade, $f_{t,0}$ the tangential force generated by the original blade, α is a penalty weight factor chosen to be 10, and Y is the total pressure loss coefficient. This case uses a comparison of tangential force between the current blade iteration and the T106D to construct the penalty term. A blade producing more tangential force is not penalized, while the opposite is penalized to ensure the optimization does not follow a direction where the rotor would not be able to drive the compressor sufficiently. The results of the $\mathbb{P}_2$ validation simulation of the T106D are used to set the penalty term, more specifically the reference tangential force used for comparison.

6.3.3 Optimization Specifications

As for the 2D case, the criteria to stop MADS is linked to the mesh size parameter of the algorithm. The initial cycle uses $\Delta_p = \Delta_m = 1$, and the optimization continues until the residual limit is passed. The residual limit is chosen to be 0.01, so that the optimization is stopped immediately when $\Delta_m < 0.01$ at the beginning of a MADS cycle. The modification that is added for this first 3D case is that a preliminary total pressure loss coefficient estimate can be used to stop some blade iterations. After the 6 minimum segments of time-averaging and the first confidence interval evaluation, the potential of the blade iteration is assessed. If the lower bound of the total pressure loss coefficient confidence interval is higher than the total pressure loss coefficient of the T106D, the simulation is stopped even if the confidence interval has not yet reached the desired precision. This ensures that the computational resources are not wasted on a blade iteration which is sub-optimal. Since the total pressure loss coefficient of the T106D is approximately 0.098, the threshold is set at 0.10. Hence, if the lower bound of the confidence interval of a blade iteration is above this threshold, the blade is considered obsolete. For example, a blade with a total pressure loss coefficient estimate of 0.11, and a confidence interval of ±0.002 after the initial 6 segments would be discarded immediately, instead of waiting for the interval to go below ±0.001.

6.3.4 Results

The MADS optimization using objective function $\mathcal{I}_1$, for total pressure loss coefficient reduction, was completed in 14 design cycles, as shown in Figure 6.6. The optimization decreased the objective function by $\sim$16%, compared to the original blade. The optimized blade generates approximately the same tangential force, compared to the original T106D, but a $\sim$16% smaller total pressure loss coefficient. The majority of improvements came from modifications to the first 4 shape parameters in the first 2 design cycles, as illustrated by the progression of design parameters in Table 6.2. A time-averaged velocity magnitude comparison, shown in Figure 6.7, illustrates the main flow modifications introduced by the optimized shape. The primary difference is the removal of the LE separation bubble, which is in large part caused by the design changes to X_2 and X_3. These parameters are responsible for the motion of the

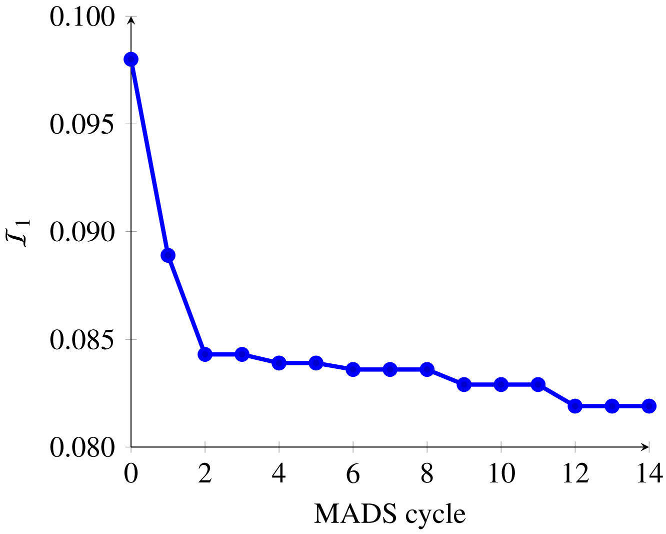


Figure 6.6. First 3D objective function progression per MADS cycles

Cycle	X_1	X_2	X_3	X_4	X_5	X_6
0	0	0	0	0	0	0.5
1	0	-0.16	0	0	0	0.5
2	0.05	-0.16	0.18	0.07	0	0.5
3	0.05	-0.16	0.18	0.07	0	0.5
4	0.05	-0.16	0.18	0.07	0	0.325
5	0.05	-0.16	0.18	0.07	0	0.325
6	0.05	-0.16	0.18	0.07	-0.015	0.325
7	0.05	-0.16	0.18	0.07	-0.015	0.325
8	0.05	-0.16	0.18	0.07	-0.015	0.325
9	0.05313	-0.16	0.18	0.07438	-0.01313	0.2375
10	0.05313	-0.16	0.18	0.07438	-0.01313	0.2375
11	0.05313	-0.16	0.18	0.07438	-0.01313	0.2375
12	0.05234	-0.1625	0.17438	0.07328	-0.00938	0.22656
13	0.05234	-0.1625	0.17438	0.07328	-0.00938	0.22656
14	0.05234	-0.1625	0.17438	0.07328	-0.00938	0.22656

Table 6.2. Evolution of shape deformation parameters for the first 3D optimization

left portion of the SS, which was expanded upward and to the left in order to match the incoming flow at positive incidence. Another major change is the decrease in boundary layer thickness on the SS as the flow approaches the TE of the optimized blade. This last modification originates from the adjustments to X_1, X_4 and X_5, changing the TE to more accurately guide the outlet flow to minimize deviation. Finally, X_6 enabled the PS to remain almost intact compared to the SS, which created an optimized blade much thicker than the original. This represents the major difference between the 2D case and this first 3D case, which both used the same objective function. The stagnation region along the PS of the optimized blade near the LE, represented by low flow velocity, is seen to be much smaller than it is for the 2D case. Although this change is visually striking, it seems to have had only a slight effect on the total pressure loss

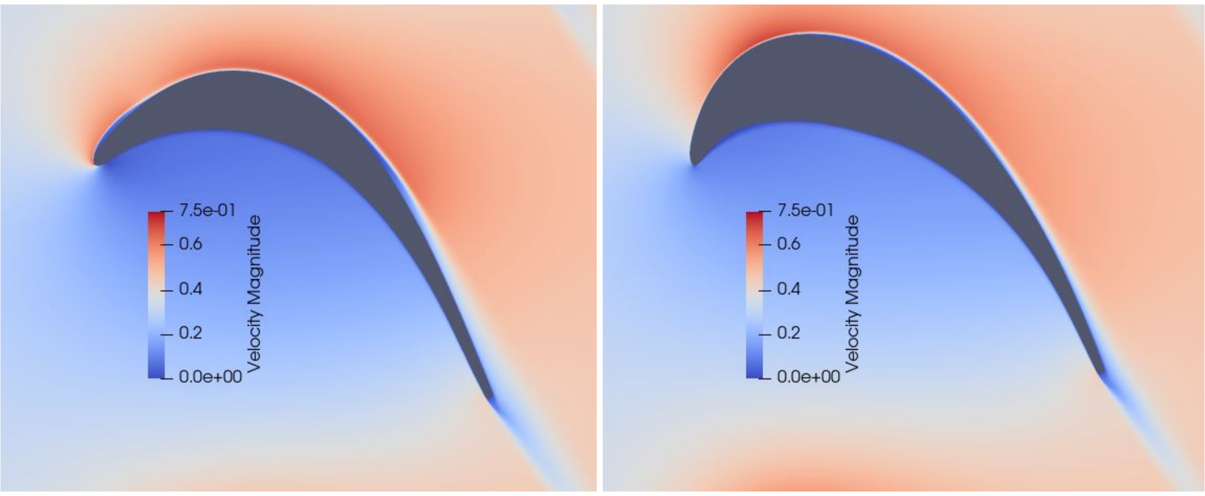


Figure 6.7. Averaged velocity magnitude comparison between T106D (left) and first 3D optimization (right)

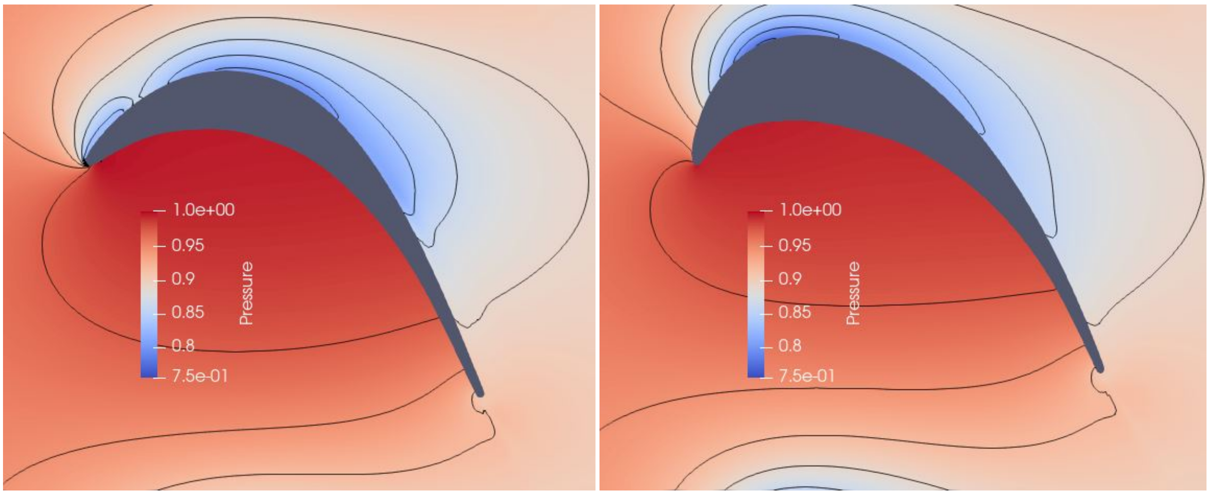


Figure 6.8. Averaged pressure contours comparison between T106D (left) and first 3D optimization (right)

coefficient produced by both blades, indicating that the main loss mechanisms occur on the SS, where the flow becomes highly turbulent. Figure 6.8, comparing the pressure contours of both blades, confirms the absence of a separation bubble and emphasizes the presence of smoother flow evolution over the SS of the optimized blade. This suggests that the initial improvement of the total pressure loss coefficient was in large part due to mitigation of the LE separation bubble.

Figure 6.9 illustrates the isentropic Mach number distribution for the baseline and optimized blades. The sharp drop along the SS at $x/c_{ax} \approx 0.1$ for the T106D occurs due to transition caused by the LE separation bubble, which is not present for the optimized blade. Instead, the first portion of the SS of the optimized blade creates a constant and smooth velocity increase up to the suction peak around $x/c_{ax} \approx 0.2$, followed by a constant and smooth velocity decrease down to the TE. The rapid velocity

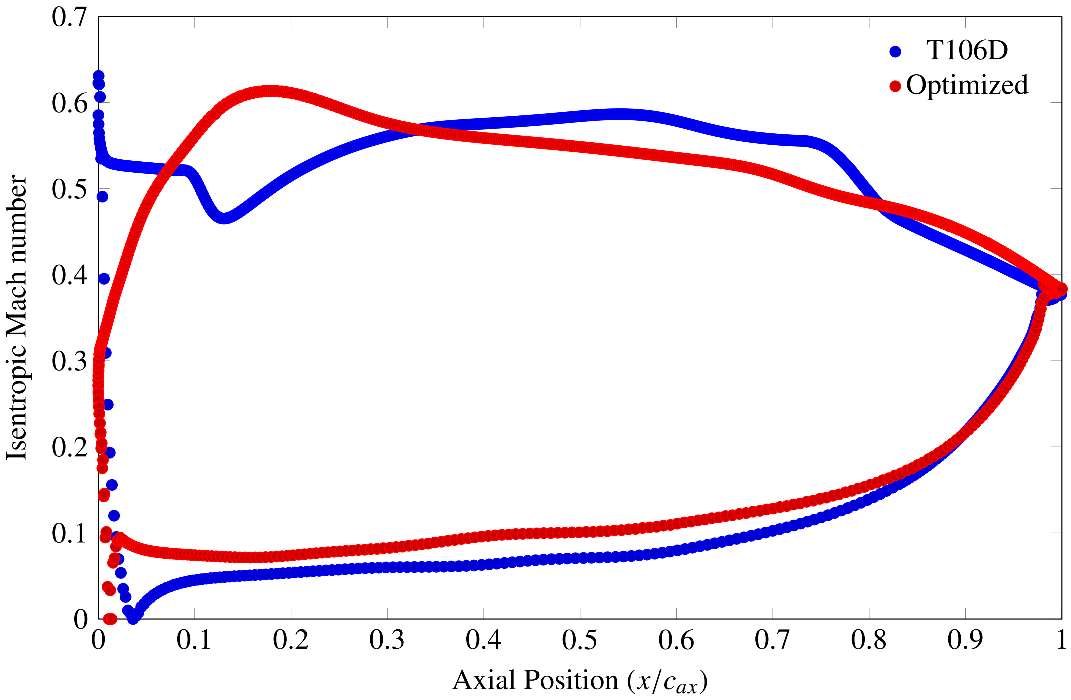


Figure 6.9. Isentropic Mach number comparison after first 3D optimization

drop at $x/c_{ax} \approx 0.8$ for the T106D is where the majority of turbulent kinetic energy is generated, and the substitution of it by a smooth velocity gradient for the optimized blade greatly reduces the entropy generated. Coupling these observations highlights how the optimization using MADS brought the design towards smoother and attached flow along the SS of the optimized blade.

Only minor differences are present on the PS, since the optimization favored maintaining a similar blade profile there. This can be explained by the fact that little to no losses occur on the PS of the T106D, since the flow remains attached and laminar. Adjustments made by the optimization process altered the position of the stagnation point, represented by the position where the isentropic Mach number reaches 0 for each blade. The initial T106D being at a positive incidence of 5 degrees meant that the stagnation point was not at the LE of the blade. The shape modifications effectively changed the inlet metal angle of the blade to more closely match the incoming flow angle, which also repositioned the stagnation point at the blade's LE. These changes suggest that the separation bubble was mainly removed by reducing the positive incidence observed by the blade, without changing the inlet flow angle.

The average turbulence kinetic energy for each blade, shown in Figure 6.10, confirms the absence of any turbulence at the LE of the optimized blade. A reduction in the boundary layer thickness and turbulence kinetic energy magnitude is also observed when comparing blades. Figure 6.11 and Figure 6.12 present Reynolds stress distributions for both blades. The optimized figures corroborate previous observations, since there is no indication of a LE separation bubble. A decrease in Reynolds stress

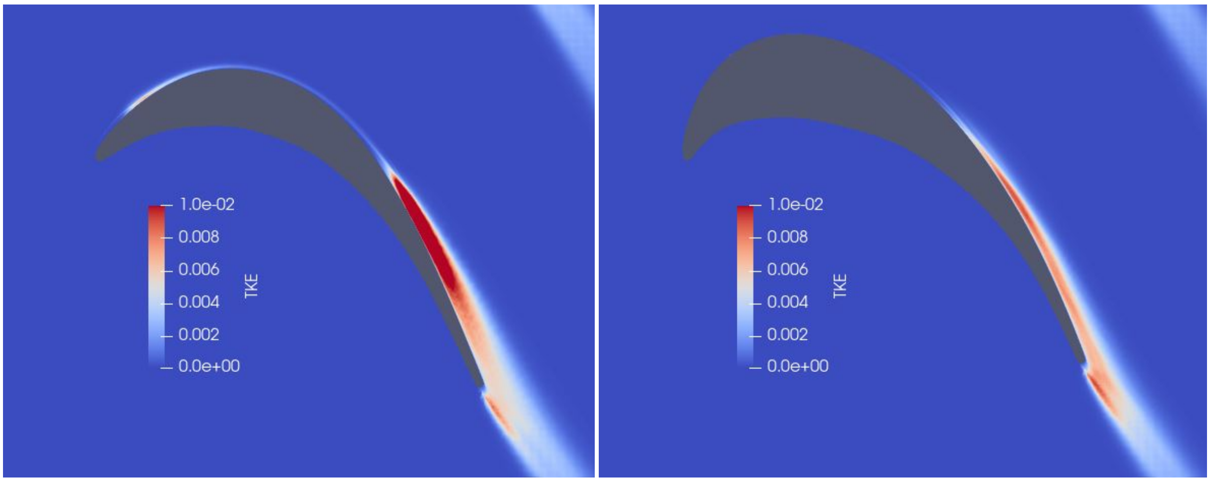


Figure 6.10. Turbulent kinetic energy comparison between T106D (left) and first 3D optimization (right)

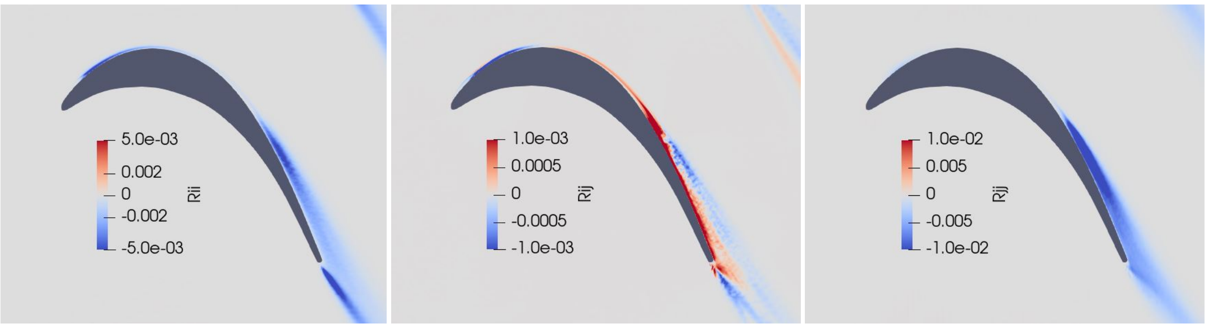


Figure 6.11. T106D normalized Reynolds stresses (τ_{ii} left, τ_{ij} middle, τ_{jj} right)

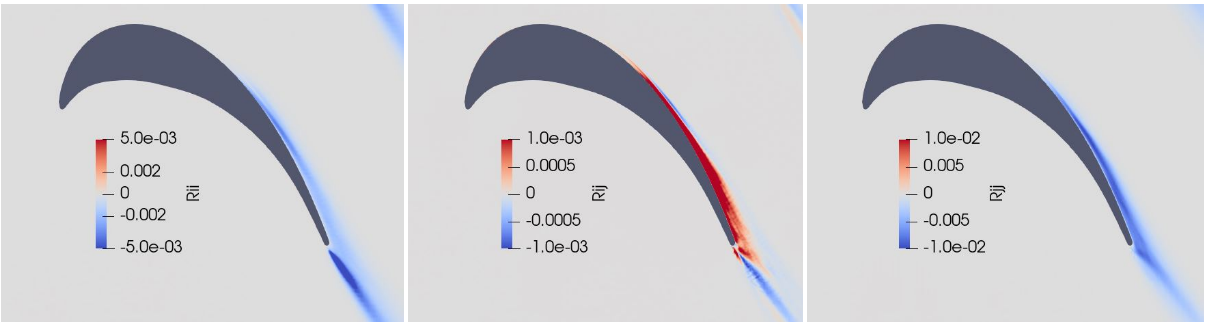


Figure 6.12. First 3D optimized blade normalized Reynolds stresses (τ_{ii} left, τ_{ij} middle, τ_{jj} right)

magnitude is observed inside the boundary layer, confirming the aforementioned reduction in turbulence intensity.

Wake properties and turbulent velocity power spectra were computed to corroborate the previous conclusions. Since both blades have different shapes at the TE and don't generate a wake at exactly the same angle, sampling was done at the meanline of each wake. The meanline was determined from the peak turbulent kinetic energy at intervals of $0.25c$, from the TE up to $2c$ downstream.

Figure 6.13 illustrates the turbulent kinetic energy distribution along the wake of each blade, where the x-axis represents the tangential position at a distance of $0.25c$ from the TE. The optimized blade

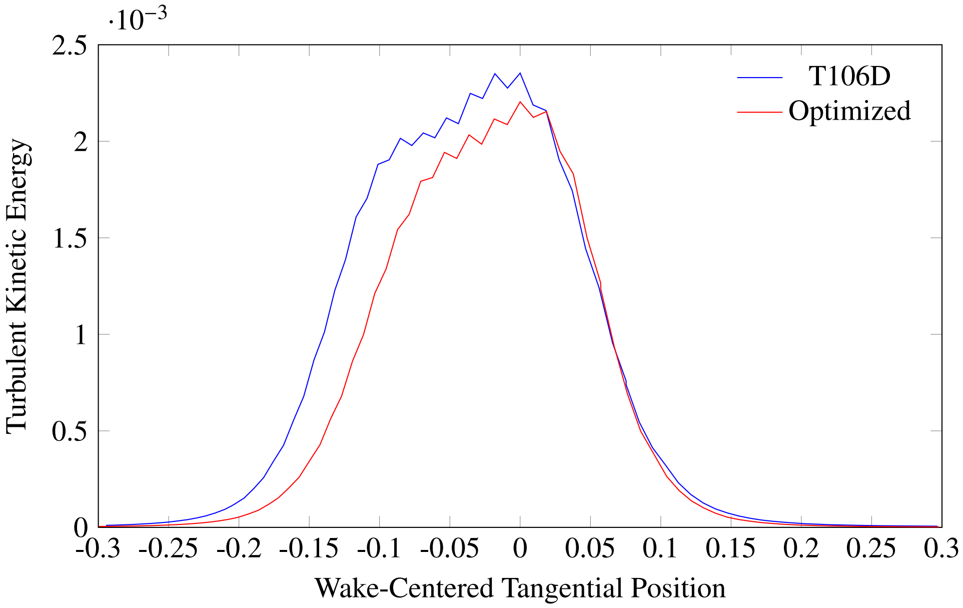


Figure 6.13. Wake profile comparison 0.25c downstream after first 3D optimization

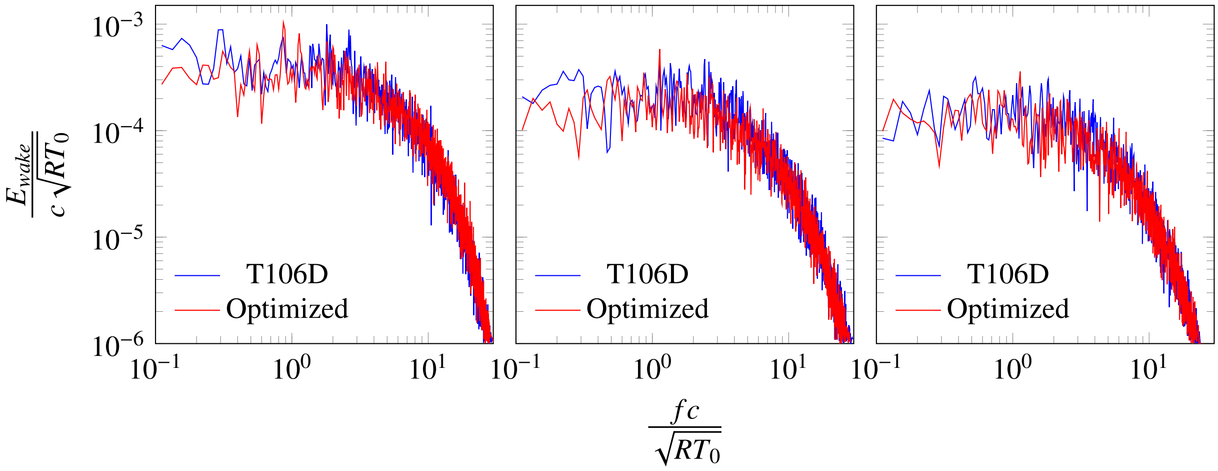


Figure 6.14. Power spectra density comparison at 3 points along the wake for the first 3D optimization

produces a slightly lower peak in turbulent kinetic energy in its wake, while also generating a thinner wake. Figure 6.14 compares the turbulent velocity power spectra density of both blades at the first 3 studied positions along their respective wakes. This confirms that the turbulence encountered in both wakes is of similar magnitudes, with the optimized blade being slightly lower than the original blade, which is most apparent at the lowest frequencies.

While the optimized blade has one or two clear peaks at each point around a frequency of 1, the T106D has no single definite peak or any dominant frequency. Comparing these observations with the iso-surfaces of q-criterion, shown Figure 6.15, helps to qualitatively differentiate the type of boundary layer separation that occurs for each blade. The T106D q-criterion show turbulent separation occurring between the mid-chord and TE, which is characterized by a broad spectrum of power spectral density at

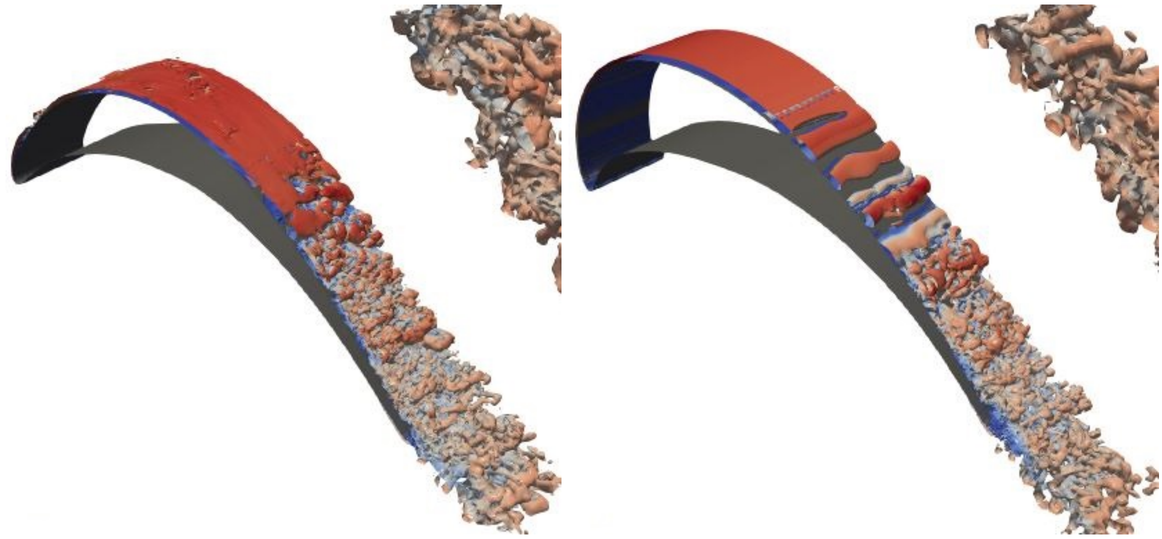


Figure 6.15. Iso-surfaces of q-criterion comparison between T106D (left) and first 3D optimization (right)

low frequencies. On the other hand, the optimized blade q-criterion show laminar separation of large unsteady vortices in the wake, which is characterized by localized power spectra density peaks. This suggests that the improvement of the total pressure loss coefficient is in part due to delayed turbulent transition on the SS.

6.3.5 Discussion

This first optimization using 3D objective function evaluations was successful in finding multiple blades that could drastically improve the initial design. The optimal design achieved its goal of significantly reducing the total pressure loss coefficient, compared to the T106D, while still generating the same amount of tangential force. On Compute Canada's Beluga cluster, running the simulations with polynomial degrees of $\mathbb{P}_0$ and $\mathbb{P}_1$ to generate proper initial conditions inside the domain took roughly 30 minutes combined for each blade. For the simulations with $\mathbb{P}_2$, each iteration took on average roughly 230 minutes to produce satisfactory results. In total, 100 blades were tested, including the T106D, and the combined simulation time was 439 hours. Hence, the computational resources needed for this design amounted to 8,777 GPU-hours, or 1 GPU-year.

Since the Beluga cluster is composed of 688 GPUs, multiple blade iterations could be conducted at the same time, in parallel. If all optimization cycles had been performed in parallel, the design would have been completed in approximately 4 days. On the other hand, if all optimization cycles had been performed in series, as for the 2D case, the design would have been completed in approximately 18 days. It should be noted that the entire cluster is not always available, as hundreds of individuals use it daily, which signifies that there were many moments where no simulations were in progress, and simply waiting in queue. Hence, the total design times provided previously are theoretical, based on the time that each simulation took to converge. Including the queue time and the downtimes caused by cluster updates and nodes crashing, the design took closer to 70 days. Some of the downtimes are inevitable when running simulations on large public clusters, such as a downtime of a full week for updates, but some of the downtimes can be mitigated. Demanding 5 nodes for a long period of time can cause the scheduler to leave the simulation longer in the queue, implying that another reduction in computational cost could be significantly beneficial for the overall wall-clock time of the design.

From the 3 to 6 stabilization segments computed before time-averaging, a coarse estimate of the total pressure loss coefficient can be obtained. While this estimate still contains remnants of low-fidelity solution, it can be used to evaluate if a blade iteration is far from the T106D or not in terms of efficiency. Hence, the next case will make use of the information provided by the initial $\mathbb{P}_2$ segments to prevent unnecessary computations of very inefficient blades, and bring the average iteration time down.

In the following section, the second aerodynamic design utilizing 3D simulations for objective function evaluations is presented in details.

6.4 Second 3D Case

In this section, the second optimization using 3D simulations for objective function evaluations is presented. This case comes after the first 3D optimization that was conducted successfully and generated a significantly more efficient design, while also providing insights into ways to diminish the computational cost associated with each blade iteration. Since this case uses 3D flow simulations and the same baseline mesh, the computational cost of each blade iteration is similar to the first 3D case, and will use the same resources. This design shifts its focus from increasing the blade efficiency to increasing the power that can be generated by the blade. It represents an opportunity to observe if the methodology selected can accurately predict separation and transition for a multitude of high-lift blades of vastly different shapes and efficiencies. This case also represents an opportunity to evaluate whether the computational cost associated with gradient-free optimization using 3D scale-resolving flow simulations can be lowered enough to make this application attractive.

6.4.1 Case Description

As previously defined, the baseline design is the T106D blade. The mesh, boundary conditions, domain, and procedure to generate the initial low-fidelity conditions inside the domain are the same as those presented in Chapter 4. The use of 3D simulations also means that the total pressure loss coefficient data is taken $1c$ downstream of the blade and that the domain does not need to be extended like it was in the 2D case. PyFR version 1.10.0 is again employed to conduct each simulation.

Since 3D flow simulations are employed for objective function evaluations in this case, the computational resources used will be the same as they were for the first 3D case. Hence, 5 nodes from Compute Canada's Beluga cluster will be used for each blade, bringing the total to 20x NVIDIA V100 SXM2 GPUs employed for each iteration.

The major difference between this case and the previous cases is the objective of the design. Instead of focusing on increasing the blade efficiency by looking for a minimum total pressure loss coefficient, this case focuses on increasing the power generated by the blade by increasing the tangential force it produces. This modification should lead to a drastically different optimal shape, where the high levels of turbulent kinetic energy seen for the T106D would remain relatively constant, since decreasing those is no longer the primary focus of optimization.

6.4.2 Case Objective

The objective function seeks to increase the tangential force generated by the blade, while keeping at most the same total pressure loss coefficient as the baseline design. It is formulated as

$$\mathcal{I}_2 = \min_{X \in DS} \begin{cases} -\left[f_t - \alpha\left(\frac{Y-Y_0}{Y_0}\right)^2\right] & Y \geq Y_0, \\ -f_t & Y < Y_0, \end{cases} \tag{6.5}$$

where Y_0 is the total pressure loss coefficient of the original blade and α is a penalty weight factor chosen to be 2. This case uses a comparison of total pressure loss coefficient between the current blade iteration and the T106D to construct the penalty term. A blade producing an lower total pressure loss coefficient is not penalized, while the opposite is penalized to ensure the optimization does not follow a direction where the rotor would have a reduced isentropic efficiency. The results of the $\mathbb{P}_2$ simulation of the T106D in the first 3D case are used to set the penalty term, more specifically the reference total pressure loss coefficient used for comparison.

6.4.3 Optimization Specifications

As for the previous case, the criteria to stop MADS is linked to the mesh size parameter of the algorithm. The initial cycle uses $\Delta_p = \Delta_m = 1$, and the optimization continues until the residual limit is passed. The residual limit is chosen to be 0.01, so that the optimization is stopped immediately when $\Delta_m < 0.01$ at the beginning of a MADS cycle. The modification that is added for this first 3D case is that an additional total pressure loss coefficient estimate can be used to stop some blade iterations, earlier than for the previous case. After the $\mathbb{P}_2$ stabilization segments, the potential of the blade is assessed by evaluating the total pressure loss coefficient at each segment. If the lower total pressure loss coefficient of all the segments is higher than a prescribed value, the simulation is stopped even if the time-average has not been computed yet. This ensures that the computational resources are not wasted on a blade iteration which is clearly not optimal. Since the total pressure loss coefficient of the T106D is approximately 0.098, this additional threshold is set at 0.12. This value is selected since the penalty term associated with this level of total pressure loss coefficient would prevent the blade from reaching the objective function value of the baseline shape, even if it produced a very large tangential force. Hence, if even the most optimistic total pressure loss coefficient segment of a blade iteration is above this threshold, the blade is considered obsolete. For example, a blade whose initial total pressure loss coefficient estimates go from 0.16 to 0.15 to 0.156 to 0.151, would be discarded before any statistical properties were extracted. Observations of the stability segments of the first 3D case showed that the segments used for time-averaging never went to higher or lower extremes of total pressure loss coefficient than those observed in the initial stability segments. This additional stopping criteria is independent of the one used for the first 3D case, and both are employed for this case.

6.4.4 Results

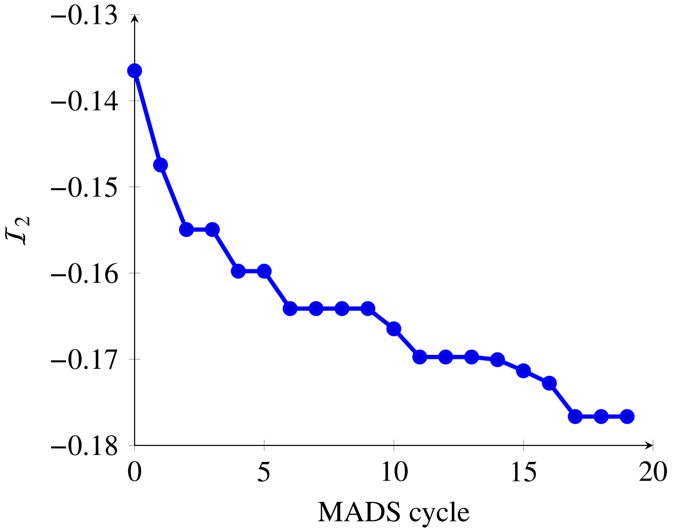


Figure 6.16. Second 3D objective function progression per MADS cycles

Optimization using objective function $\mathcal{I}_2$, focusing on tangential force augmentation, was completed

Cycle	X_1	X_2	X_3	X_4	X_5	X_6
0	0	0	0	0	0	0.5
1	0	-0.16	0	0	-0.03	0.5
2	0.05	-0.16	0	0.07	-0.03	0.5
3	0.05	-0.16	0	0.07	-0.03	0.5
4	0.05	-0.16	-0.045	0.0525	-0.0225	0.325
5	0.05	-0.16	-0.045	0.0525	-0.0225	0.325
6	0.0375	-0.12	-0.09	0.0175	-0.0375	0.2375
7	0.0375	-0.12	-0.09	0.0175	-0.0375	0.2375
8	0.0375	-0.12	-0.09	0.0175	-0.0375	0.2375
9	0.0375	-0.12	-0.09	0.0175	-0.0375	0.2375
10	0.04375	-0.12	-0.09	0.0175	-0.0375	0.2375
11	0.05313	-0.12	-0.09	0.00875	-0.03	0.30313
12	0.05313	-0.12	-0.09	0.00875	-0.03	0.30313
13	0.05313	-0.12	-0.09	0.00875	-0.03	0.30313
14	0.0539	-0.115	-0.08438	0	-0.02953	0.31406
15	0.05703	-0.105	-0.07312	-0.0175	-0.02766	0.33593
16	0.06328	-0.085	-0.07312	-0.0175	-0.02766	0.24843
17	0.07578	-0.085	-0.07312	-0.0175	-0.02766	0.24843
18	0.07578	-0.085	-0.07312	-0.0175	-0.02766	0.24843
19	0.07578	-0.085	-0.07312	-0.0175	-0.02766	0.24843

Table 6.3. Evolution of shape deformation parameters for the second 3D optimization

in 19 cycles, as shown in Figure 6.16. The optimized blade generates a slightly lower total pressure loss coefficient, compared to the original T106D, and generates a tangential force increase of ~29%. Table 6.3 illustrates the progression of optimum design variables through each optimization cycle. Whereas the first optimization converged quickly and only small modifications occurred after cycle 2, the second optimization went through vastly different blades and successfully exited two local minima, from cycles 6 to 9 and 11 to 13.

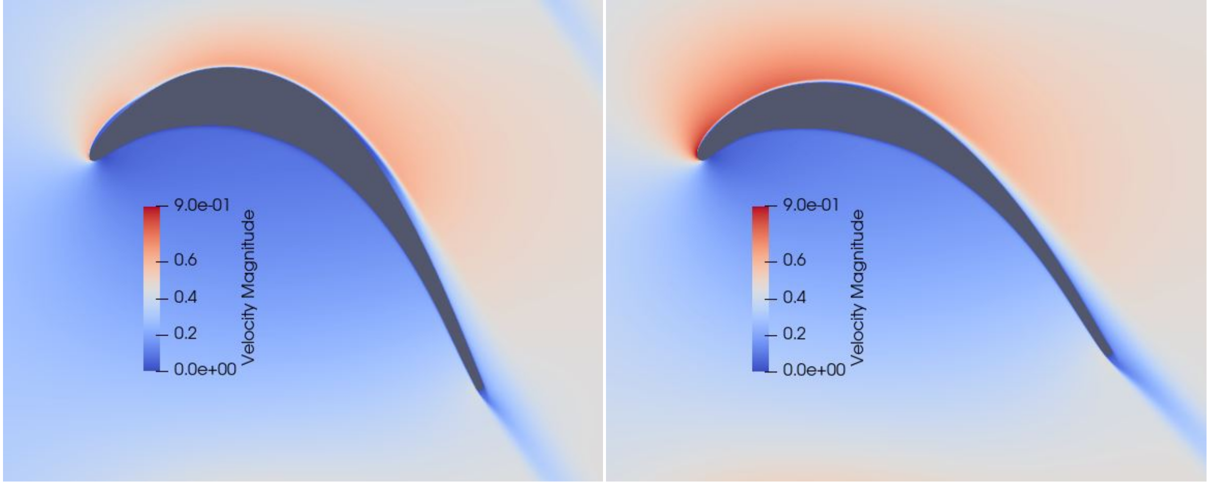


Figure 6.17. Averaged velocity magnitude comparison between T106D (left) and second 3D optimization (right)

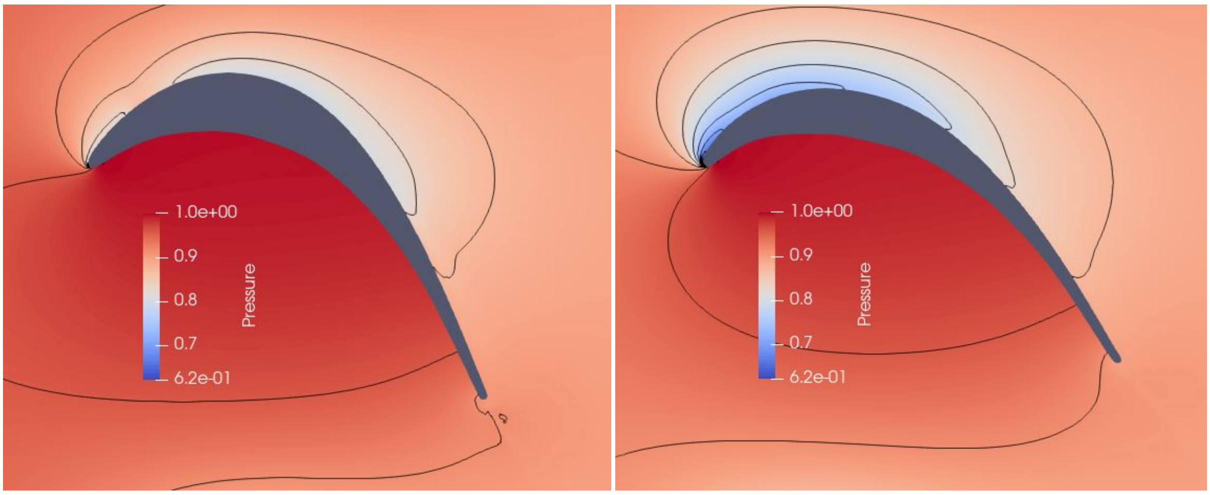


Figure 6.18. Averaged pressure contours comparison between T106D (left) and second 3D optimization (right)

Figure 6.17 compares the time-averaged velocity magnitude of both blades and illustrates the changes of flow features associated with the optimization. The primary differences are the removal of the LE separation bubble and the increased velocity along the first half of the SS of the optimized blade. The transition along the second half of the SS can also be seen to be more gradual for the optimized blade, compared to the sharp velocity decrease for the T106D. Figure 6.18 compares the time-averaged pressure contours of both blades and illustrates that the increased velocity of the optimized blade along its SS results in a significantly lower pressure, especially along the first half. It also shows the absence of a LE separation bubble, which was achieved by moving the left portion of the SS to the left, effectively decreasing the incoming flow incidence on the blade's LE. This suggests that the increase in tangential force was achieved by mitigating the laminar separation bubble and more efficiently accelerating the flow along the SS of the optimized blade. The aforementioned figures also illustrate that the optimized blade is smaller than the original blade, which enhances the capability of this blade profile to reduce the overall weight of the LPT.

The pressure coefficient distribution along the surface of each blade is shown in Figure 6.19. The optimized blade shows significantly lower pressure coefficients along its SS than the T106D, especially on the first half of the surface. On the second half of the SS, the coefficients of the optimized blade increase steadily, indicating a smoother transition of the flow into the separated turbulent region. The onset of turbulent separation is illustrated by a small pressure fluctuation at $x/c_{ax} \approx 0.45$ for the optimized blade, which is significantly smoother than onset of separation at $x/c_{ax} \approx 0.8$ for the T106D. Along the PS, the coefficients are similar, suggesting that these changes in tangential force come from modifications of the SS mainly. From the data presented, the optimized blade can be characterized as a front-loaded

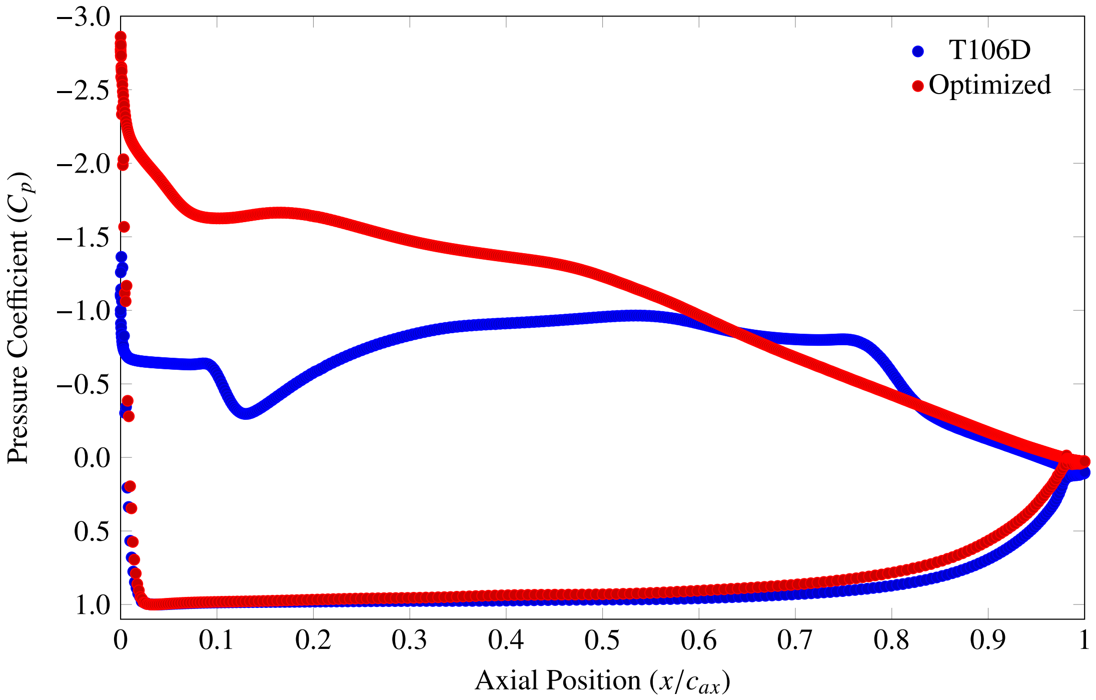


Figure 6.19. Pressure coefficient distribution comparison after second 3D optimization

LPT blade, since the minimum pressure coefficients are located at $0 \leq x/c_{ax} \leq 0.3$. Front-loaded blades have been shown to be the most robust high-lift designs over all flow conditions, since they allow large pressure gradients to be spread out over a longer SS distance, while also promoting transition before separation [13].

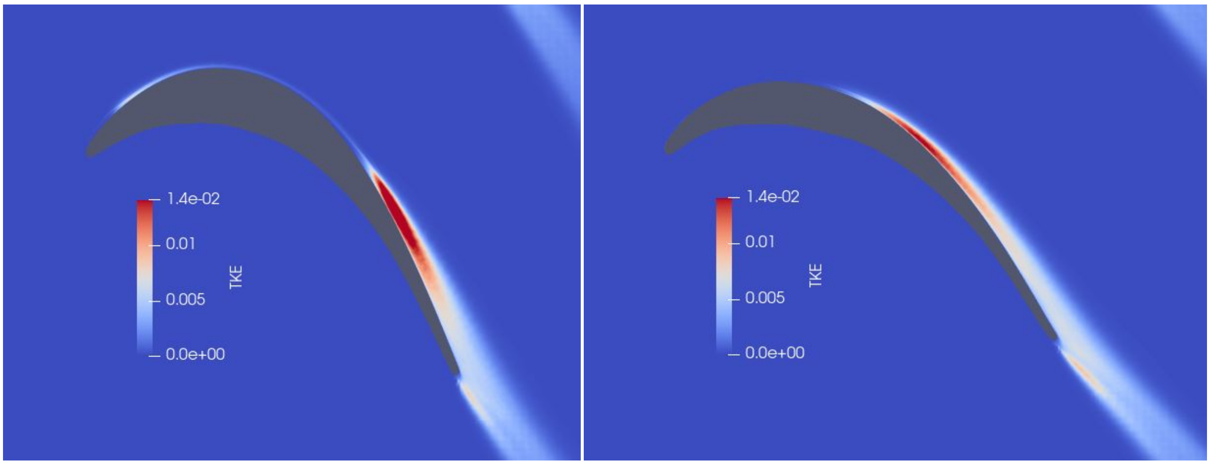


Figure 6.20. Turbulent kinetic energy comparison between T106D (left) and second 3D optimization (right)

A comparison of time-averaged turbulent kinetic energy, presented in Figure 6.20, confirms that there is no laminar separation bubble on the optimized blade. It shows that separation on the second half of the SS occurs earlier for the optimized blade than the T106D. The turbulent kinetic energy peak is also

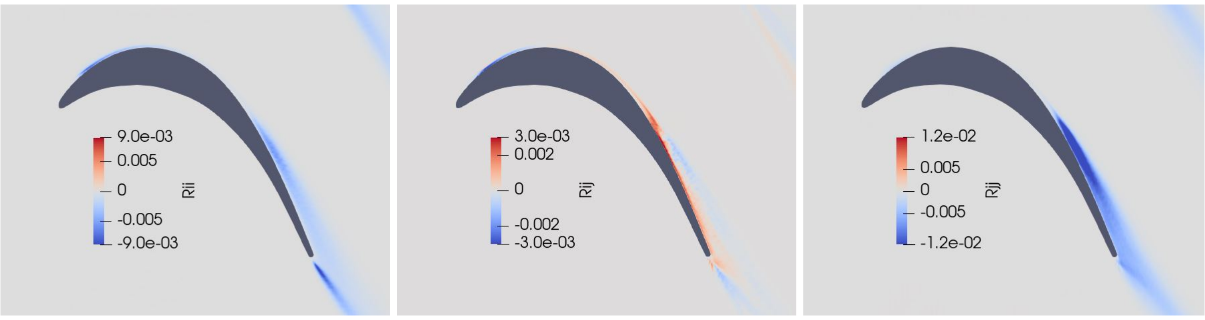


Figure 6.21. T106D normalized Reynolds stresses (τ_{ii} left, τ_{ij} middle, τ_{jj} right)

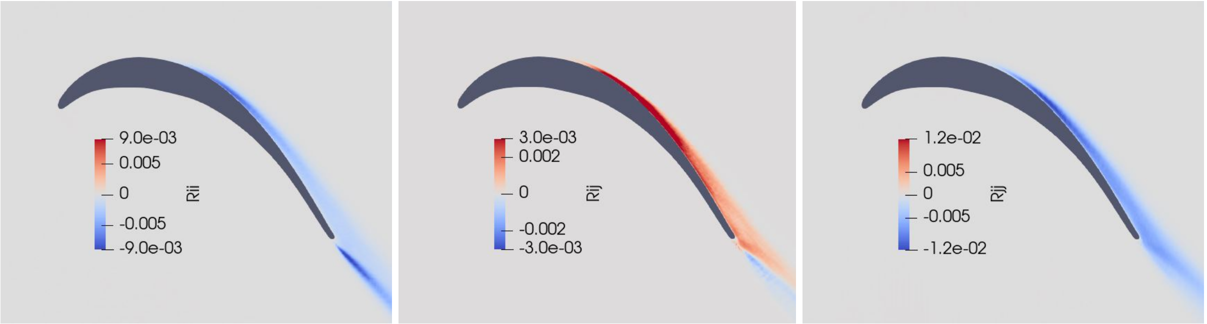


Figure 6.22. Second 3D optimized blade normalized Reynolds stresses (τ_{ii} left, τ_{ij} middle, τ_{jj} right)

shown to be lower and the turbulent boundary layer is shown to be slightly thinner than on the original blade. Figure 6.21 and Figure 6.22 present the Reynolds stress distributions for the original and optimized blades. Comparison of those figures corroborates previous observations about the laminar separation bubble and turbulent boundary layer. The magnitudes of the Reynolds stresses τ_{ii} and τ_{jj} inside the boundary layer are shown to be smaller for the optimized blade, indicating lower peaks of turbulence intensity being spread out over a longer boundary layer.

Similar to the first optimization, wake properties and turbulent velocity power spectra density were computed at the meanline of each blade's wake, at intervals of $0.25c$. Figure 6.23 presents the comparison of the turbulent kinetic energy magnitude in the wake of each blade. It is observed that the optimized blade generates a higher peak turbulent kinetic energy, which is spread out over a narrower wake. As the total pressure loss coefficient of each blade is similar, it would be expected that the wake generated by both would amount to similar levels of turbulence kinetic energy. Taking the average turbulence kinetic energy along their respective wakes yields a similar value for both blades. Hence, a narrower wake attaining a higher peak turbulence intensity corroborates the similarity in the total pressure loss coefficients and the shape of the boundary layer, which was previously observed to be thinner for the optimized blade. The turbulent velocity power spectra density comparison in Figure 6.24 adds to this conclusion, by revealing that the optimized blade's wake contains higher levels of turbulent energy at

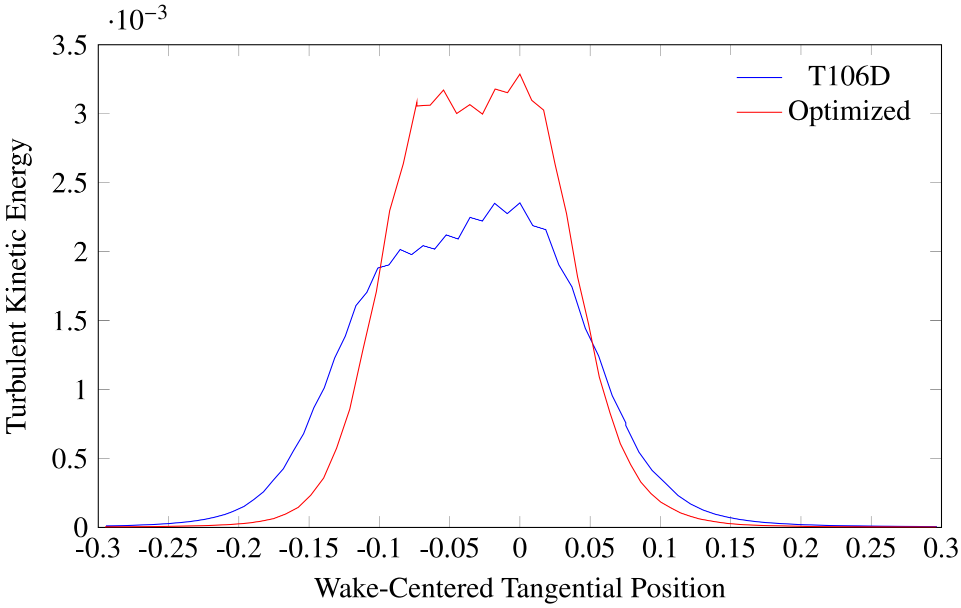


Figure 6.23. Wake profile comparison 0.25c downstream after second 3D optimization

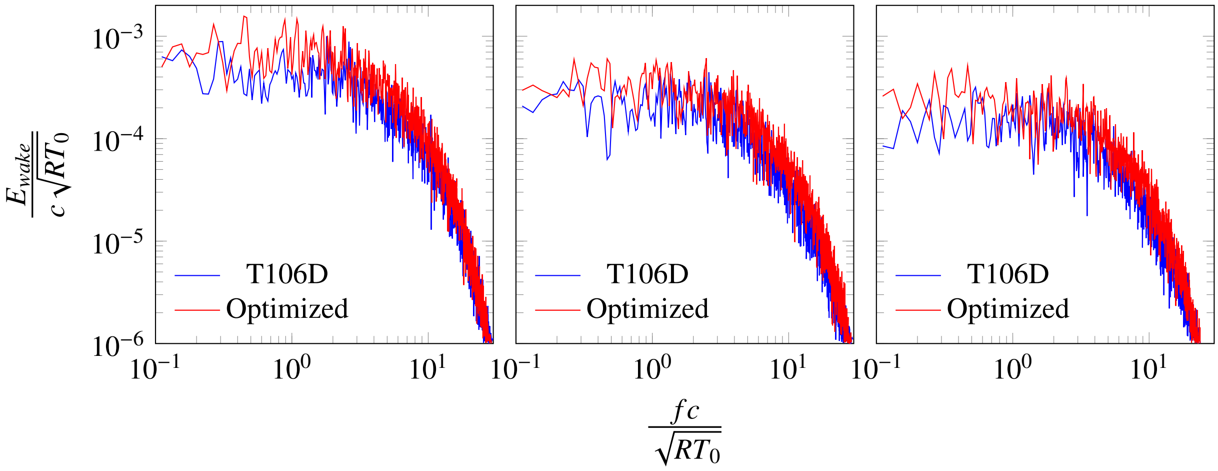


Figure 6.24. Power spectra density comparison at 3 points along the wake for the second 3D optimization

almost all frequencies.

Finally, iso-surfaces of q-criterion presented in Figure 6.25 help summarize the difference between both blades. It shows the absence of a laminar separation bubble, earlier onset of separation, larger turbulent structures in the boundary layer and wake, a narrower wake, and higher velocity magnitudes along the first portion of the SS for the optimized blade.

6.4.5 Discussion

This second optimization using 3D objective function evaluations was successful in finding multiple blades that could drastically improve the initial design. The optimal design achieved its goal of significantly increasing the tangential force, compared to the T106D, while still generating a similar total pressure loss

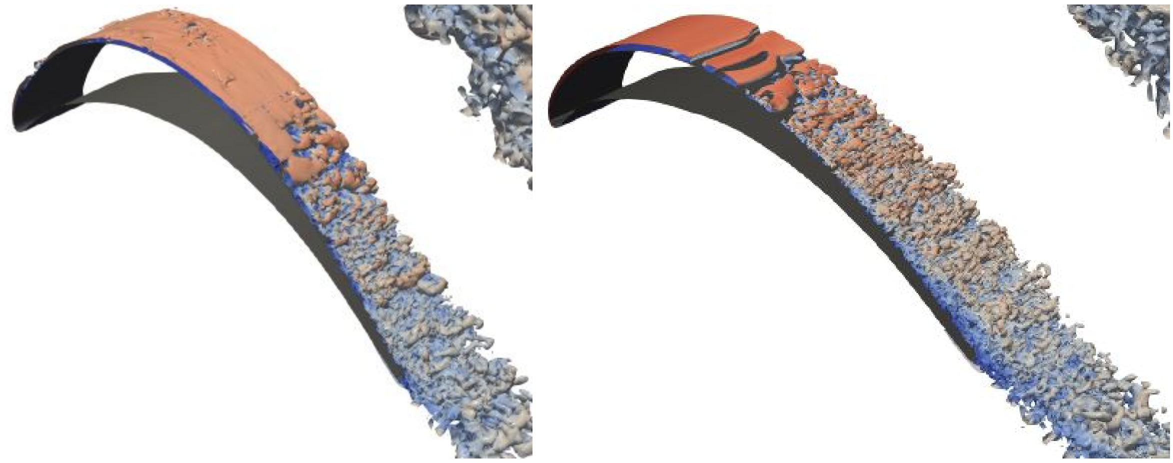


Figure 6.25. Iso-surfaces of q-criterion comparison between T106D (left) and second 3D optimization (right)

coefficient. On Compute Canada's Beluga cluster, running the simulations with polynomial degrees of $\mathbb{P}_0$ and $\mathbb{P}_1$ to generate proper initial conditions inside the domain took roughly 30 minutes combined for each blade. For the simulations with $\mathbb{P}_2$, each iteration took on average roughly 170 minutes to produce satisfactory results. In total, 157 blades were tested, including the T106D, and the combined simulation time was 524 hours. Hence, the computational resources needed for this design amounted to 10,476 GPU-hours, or 1.20 GPU-year.

Since the Beluga cluster is composed of 688 GPUs, multiple blade iterations could be conducted at the same time, in parallel. If all optimization cycles had been performed fully in parallel, the design would have been completed in approximately 3.5 days. On the other hand, if all optimization cycles had been performed in series, as for the 2D case, the design would have been completed in approximately 22 days. Similarly to the first 3D case, it should be noted that the entire cluster is not always available, as hundreds of individuals use it daily, which signifies that there were many moments where no simulations were in progress, and simply waiting in queue. Hence, the total design times provided previously are theoretical, based on the time that each simulation took to converge. Including the queue time and the downtimes caused by cluster updates and nodes crashing, the design took closer to 35 days. Updates installed on the clusters made each simulation slightly faster than for the previous case, and shorter average simulation times enabled more iterations to be completed before reaching the daily limit of the scheduler.

6.5 Comments

This section first offers a review of the optimization results previously discussed, and compares each design. A discussion on the computational resources and industrial feasibility is then presented.

Looking at the two cases employing the same objective function, it can be observed that the 2D design offered many similarities to the 3D design. Although not identical, both designs converged towards a similar region of the design space, where design variables X_1, X_2, X_3, X_4, and X_5 all ended up around the same values. Even though the flow phenomenon that cause total pressure losses are inherently affected by 3D fluid interactions, the 2D design provided a reasonable estimate of the optimal blade. Its main downfall is the reduced importance given to the PS effects on the total pressure loss coefficient, which lead to the final 2D design having a drastically different PS shape than the 3D design. Overall, the 2D design was able to reliably preview the design space where the 3D design would be located, while only implying a fraction of the computational cost. Another interesting outcome of these designs is the fact that both resulted in a blade with the same flow turning as the T106D. The inlet flow angle remaining unchanged for all blades, flow turning could only be affected by the outlet flow angle produced by each blade iteration. Although this was not explicitly required by the objective function, each optimal design evolved by reducing the outlet metal angle to more closely match the initial outlet flow angle of the T106D. Bringing these two angles closer together meant that the deviation observed for the T106D was greatly reduced for both designs, which helped in the reduction of the total pressure loss coefficient.

While the designs based on efficiency improvements were very similar to typical aerodynamic designs conducted in the aerospace industry, the design aiming for tangential force increase was different. While the main objective was met successfully, a complementary objective was also to observe the ability of the method to accurately resolve separation and transition occurring at vastly different positions and intensity for each iteration. The optimal design shows that the methodology proposed can handle different separation and transition onsets, while also showing how the two main design objectives of this study lead to very different designs. The first 3D design shows how improving efficiency is mainly a function of matching metal angles and flow angles, and creating a smooth transition from blade LE to TE along the SS. The second 3D design shows how increasing tangential force is not achieved by matching metal and flow angles, and is instead achieved by reducing the outlet flow angle and by increasing the LE pressure gradient. Although the flow turning was reduced in the second 3D design, the complimentary objective was observed clearly, and future use of this methodology can simply incorporate a penalty term pertaining to the outlet flow angle.

In terms of computational cost, the 2D design could have been achieved in 1-2 days, if a cluster from Compute Canada had been used. This implies that an estimate of the optimal design space could be obtained relatively cheaply for LPT cascade cases including separation and transition. For the 3D cases, the evaluation of the total pressure loss coefficient ended up as the ultimate bottleneck that prevented further reduction in the necessary computational cost. Even with 5 more MADS cycles needed, the overall and individual wall-clock time decreased significantly in the second 3D case, compared to the first 3D case, but this reduction does not seem sufficient. As the average computational time needed per blade iteration is around 3 hours on 20 GPUs, the application of the current state of the method to industrial problems might not seem appealing. On the other hand, a property such as the tangential force took on average 1 hour on 20 GPUs to converge towards an accurate time-average, a significantly faster convergence. Hence, the use of scale-resolving flow solutions to drive a design should try to stay away from the traditional objectives employed for RANS optimizations, such as the total pressure loss coefficient and isentropic efficiency. Instead, objective functions should ideally be comprised of tangential force, axial force, flow angles, lift, drag, or other simple properties to converge. The need for this constraint constitutes an unexpected downfall of using scale-resolving simulations in an optimization setting, which could be worked around in future studies. Overall, the methodology presented in this study has shown promise in its ability to provide good optimization capabilities at reasonable computational costs for cases where separation and transition are major roadblocks.

Chapter 7

Conclusion

In this research, the FR approach was proposed to perform scale-resolving ILES of LPT cascades. This spatial discretization method provides an effective means to achieve high-order accuracy, while also containing low levels of numerical dissipation. This last property of FR makes it particularly appealing for ILES applications, where the numerical dissipation replaces traditional SGS models needed for LES. Applying this approach yielded a scalable and general methodology to solve for a variety of transitional and separated flows over turbine blades. The application of ILES using FR on modern hardware architectures also provided a cost effective platform for evaluating objective functions in an optimization setting. The selection of LPT cascades as the breakthrough application came from the fact that LPTs operate at relatively low Reynolds number, their design is incrementally moving towards high-lift configurations, and traditional RANS solvers are not reliable enough to capture separation and transition omnipresent in high-lift problems. This combination of factors showed that new tools were needed for this application, and that the methodology proposed could bridge the current gaps.

First, a validation of the methodology was conducted on the challenging T106D case. A comparison with experimental data and between different orders of accuracy on a selected mesh helped to ensure that the methodology could accurately predict and capture complex phenomenon, such as turbulent separation and transition. This validation case set the mesh, boundary conditions, and initial conditions that would be used for the rest of the study.

Afterwards, multiple aerodynamic design processes were conducted, starting from the baseline T106D blade. Optimizing using unsteady flow solutions required the use of a gradient-free method, specifically MADS in this study. The MADS algorithm allowed for fast constrained optimization cycles that were able to efficiently converge towards the global optimum of the objective function. Three separate optimizations were conducted, two focused on total pressure loss coefficient reduction using 2D and 3D simulation results and the other focused on tangential force increase. All were successful in finding different blade profiles and achieving significant reduction of the objective functions, while accurately resolving the major turbulent phenomenon. The 2D optimization achieved a ~15% reduction of total pressure loss coefficient, while the first optimization using 3D simulations achieved a ~16% reduction of total pressure loss coefficient. The second optimization using 3D simulations achieved a ~29% increase of tangential force produced by the blade. Overall, this research provided multiple good benchmark designs, and valuable insights as to how optimization using scale-resolving simulations should be conducted to achieve optimal performance.

7.1 Future Work

This thesis is only the first step in the development of a proper method to perform aerodynamic design using scale-resolving simulations, and can be improved in future studies. Additional work on this topic should further explore the application of MADS and ILES to optimize LPT cascades with more objectives and penalty terms added to the optimization to resemble a more traditional aerodynamic design process. Adding the outlet flow angle and blade cross-sectional area as objectives, and the pitch as an optimization parameter would generate a wider range of designs that would be applicable directly in the original engine.

Further techniques could also be applied to each simulation in order to reduce computational cost of each objective function evaluation, enabling the designer to add more design variables or accelerate the optimization procedure. Techniques like polynomial adaptation could be applied away from the blade surface, where there is no need for high levels of accuracy, to reduce the number of solution points and simulation time. Changing how the efficiency is represented in the objective functions would also help speedup the convergence of each objective function evaluation. Using a mesh that is coarser, but can achieve the same accuracy using a higher polynomial representation within each element would help increase the scalability of the simulations, enabling faster blade iterations on large clusters.

Lastly, a hybrid approach using the methodology proposed in this study and a gradient-based approach in different regions of the design space could also constitute a viable means of reducing computational cost, or achieving a better design. While a suitable gradient-based method to employ with scale-resolving simulations is still an unresolved topic, it could enable a large amount of design variables to be added, permitting more intricate deformations to be applied to the design. Hence, a hybrid approach using a gradient-free method to identify the region of the global optimum, and then switching to a gradient-based method for precise design features would be an ideal process to obtain better designs at reasonable computational costs.

References

[1] J. S. Litt, E. Wong, M. J. Krasowski, and L. C. Greer. Cooperative multi-agent mobile sensor platforms for jet engine inspection - conception and implementation. Technical Report 212708, National Aeronautics and Space Administration, November 2003.

[2] C. B. Allen, D. J. Poole, and T. C. S. Rendall. Wing aerodynamic optimization using efficient mathematically-extracted modal design variables. *Optimization and Engineering*, 19:453–477, 2018.

[3] S. N. Skinner and H. Zare-Behtash. State-of-the-art in aerodynamic shape optimisation methods. *Applied Soft Computing*, 62:933–962, 2018.

[4] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis. CFD Vision 2030 study: a path to revolutionary computational aerosciences. Technical Report 218178, NASA, March 2014.

[5] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. *Mathematics of Computation*, 52(186):411–435, April 1989.

[6] B. Cockburn, S.-Y. Lin, and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems. *Journal of Computational Physics*, 84(1):90–113, September 1989.

[7] B. Cockburn, S. Hou, and C.-W. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. *Mathematics of Computation*, 54(190):545–581, April 1990.

[8] Z. J. Wang. Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation. *Journal of Computational Physics*, 178(1):210–251, May 2002.

[9] Y. Liu, M. Vinokur, and Z. J. Wang. Discontinuous spectral difference method for conservation laws on unstructured grids. *Journal of Computational Physics*, 216, June 2004.

[10] International Coordinating Council of Aerospace Industries Associations (ICCAIA). Advancing technology opportunities to further reduce CO_2 emissions. Technical report, International Civil Aviation Organization, 2019. In ICAO's 2019 environmental report.

[11] PwC. *Tailwinds report: 2018 airline industry trends*, 2018. URL: https://www.pwc.com/us/en/industries/transportation-logistics/tailwinds-airline-industry-trends.html.

[12] B. Lakshminarayana. *Fluid dynamics and heat transfer of turbomachinery*. John Wiley & Sons, Inc., 1996.

[13] M. W. McQuilling. *Design and validation of high-lift low-pressure turbine blade*. PhD thesis, Wright State University, 2007.

[14] E. M. Curtis, H. P. Hodson, M. R. Banieghbal, J. D. Denton, R. J. Howell, and N. W. Harvey. Development of blade profiles for low-pressure turbine applications. *Journal of Turbomachinery*, 119(3):531–538, July 1997.

[15] G. I. Ilieva. *Turbulence modeling approaches - current state, development prospects, applications*, chapter 6. InTechOpen, July 2017.

[16] S. Girgis. Gas turbine design. Course notes at Concordia University, 2019.

[17] H. I. H. Saravanamuttoo, G. F. C. Rogers, H. Cohen, P. V. Straznicky, and A. C. Nix. *Gas turbine theory*. Pearson Education Limited, 7th edition, 2017.

[18] M. Arabnia. *Aerodynamic shape optimization of axial turbines in three dimensional flow*. PhD thesis, Concordia University, 2012.

[19] B. C. Vermeire, F. D. Witherden, and P. E. Vincent. On the utility of GPU accelerated high-order methods for unsteady flow simulations: A comparison with industry-standard tools. *Journal of Computational Physics*, 334:497–521, April 2017.

[20] F. D. Witherden, A. M. Farrington, and P. E. Vincent. PyFR: An open source framework for solving advection-diffusion type problems on streaming architectures using the flux reconstruction approach. *Computer Physics Communications*, 185:3028–3040, July 2014. URL = http://pyfr.org/.

[21] P. Vincent, F. Witherden, B. Vermeire, J. S. Park, and A. Iyer. Towards green aviation with python at petascale. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, number 1 in SC '16. IEEE Press, 2016.

[22] H. T. Huynh. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. In *18th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, June 2007. AIAA 2007-4079.

[23] J. C. Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, Ltd, 3rd edition, 2016.

[24] P. A. Davidson. *Turbulence: an introduction for scientists and engineers*. Oxford University, 2 edition, July 2015.

[25] S. B. Pope. *Turbulent flows*. Cambridge University Press, August 2000.

[26] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Gary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, N. Kroll, G. May, P. O. Persson, B. van Leer, and M. Visbal. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, July 2013.

[27] C. A. Pereira. Analysis of high-order element types for implicit large eddy simulation. Master's thesis, Concordia University, July 2019.

[28] B. C. Vermeire, S. Nadarajah, and P. G. Tucker. Implicit large eddy simulation using the high-order correction procedure via reconstruction scheme. *International Journal for Numerical Methods in Fluids*, 82(5):231–260, October 2016.

[29] B. C. Vermeire and P. E. Vincent. On the properties of energy stable flux reconstruction schemes for implicit large eddy simulation. *Journal of Computational Physics*, 327:368–388, December 2016.

[30] B. C. Vermeire. Recent progress in high-order methods for scale resolving simulations of low pressure turbines. In *Proceedings of Montreal 2018*, volume 122. Global Power and Propulsion Society, May 2018.

[31] P. Stadtmuller, L. Fottner, and A. Fiala. Experimental and numerical investigation of wake-induced transition on a highly-loaded LP turbine at low Reynolds numbers, 2000-GT-0269. In *Proceedings of ASME TURBOEXPO 2000*, May 2000.

[32] D. W. Zingg, M. Nemec, and T. H. Pulliam. A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization. *European Journal of Computational Mechanics*, 17:103–126, 2008.

[33] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(1):97–110, 1974.

[34] A. Jameson and S. Kim. Reduction of the adjoint gradient formula for aerodynamic shape optimization problems. *AIAA Journal*, 41(11):2114–2129, November 2003.

[35] A. Jameson and K. Ou. *Encyclopedia of Aerospace Engineering*, volume 1, chapter Optimization methods in computational fluid dynamics. John Wiley & Sons, Inc., December 2010.

[36] M. Schramm, B. Stoevesandt, and J. Peinke. Optimization of airfoils using the adjoint approach and the influence of adjoint turbulent viscosity. *Computation*, 6(5), January 2018.

[37] D. J. Mavriplis. Discrete adjoint-based approach for optimization problems on three-dimensional unstructured meshes. *AIAA Journal*, 45(4):740–750, April 2007.

[38] J. Luo, J. Xiong, and F. Liu. Aerodynamic design optimization by using a continuous adjoint method. *Science China Physics, Mechanics and Astronomy*, 57(7):1363–1375, July 2014.

[39] J. Luo, J. Xiong, F. Liu, and I. McBean. Three-dimensional aerodynamic design optimization of a turbine blade by using an adjoint method, GT2009-60115. In *Proceedings of ASME Turbo Expo 2009: Power for Land, Sea and Air*, June 2009.

[40] L. Chen and J. Chen. Aerodynamic optimization design for high pressure turbines based on the adjoint approach. *Chinese Journal of Aeronautics*, 28(3):757–769, April 2015.

[41] D. J. Lea, M. R. Allen, and T. W. N. Haine. Sensitivity analysis of the climate of a chaotic system. *Tellus A: Dynamic Meteorology and Oceanography*, 52(5):523–532, 2000.

[42] P. J. Blonigan, P. Fernandez, S. M. Murman, Q. Wang, G. Rigas, and L. Magri. Toward a chaotic adjoint for LES. In *Proceedings of the CTR Summer Program 2016*, pages 385–394, January 2017.

[43] A. Rubino, M. Pini, P. Colonna, T. Albring, S. Nimmagadda, T. Economon, and J. Alonso. Adjoint-based fluid dynamic design optimization in quasi-periodic unsteady flow problems using a harmonic balance method. *Journal of Computational Physics*, 372:220–235, June 2018.

[44] N. Gauger and J. Brezillon. *MEGAFLOW-Numerical flow simulation for aircraft design*, volume 89, chapter 13, pages 181–193. Springer, 2005.

[45] Z. Lyu, G. K. W. Kenway, and J. R. R. A. Martins. Aerodynamic shape optimization investigations of the common research model wing benchmark,. *AIAA Journal*, 53(4):968–985, April 2015.

[46] R. Hassan, B. Cohanim, and O. de Week. A comparison of particle swarm optimization and the genetic algorithm. In *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*. American Institute of Aeronautics and Astronautics, April 2005. AIAA-2005-1897.

[47] C. Grosan, A. Abraham, and M. Nicoara. Search optimization using hybrid particle sub-swarms and evolutionary algorithms. *International Journal of Simulation Systems, Science & Technology*, 6(10-11):60–79, 2005.

[48] C. Audet and J. E. Dennis JR. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006.

[49] A. J. Booker, J. E. Dennis Jr., P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17:1–13, 1999.

[50] I. D. Cooper and C. J. Price. On the convergence of grid-based methods for unconstrained optimization. *SIAM Journal on Optimization*, 11:859–869, 2001.

[51] C. Audet and W. Hare. *Derivative-free and blackbox optimization*. Springer, 1st edition, December 2017.

[52] H. R. Karbasian. *Design in chaos: aerodynamic optimization using high-fidelity computational fluid dynamics*. PhD thesis, Concordia University, August 2021.

[53] H. R. Karbasian and B. C. Vermeire. Gradient-free aerodynamic shape optimization using large eddy simulation. *Journal of Computational Physics*, (in review), 2021.

[54] S. Bahrami. Multi-fidelity design optimization of francis turbine runner blades. 2015.

[55] S.Bahrami, C. Tribes, C. Devals, T. C. Vu, and F. Guibault. Multi-fidelity shape optimization of turbine runner blades using a multi-objective mesh adaptive direct search algorithm. *Applied Mathematical Modelling*, 40(2):1650–1668, January 2016.

[56] W. Yang. *Surgical design for the Fontan procedure using computational fluid dynamics and derivative-free optimization*. PhD thesis, University of California, San Diego, 2012.

[57] C. Hirsch. *Numerical computation of internal & external flows*, volume 1. John Wiley & Sons, Ltd, 2nd edition, 2007.

[58] H. T. Huynh. A reconstruction approach to high-order schemes including discontinuous Galerkin for diffusion. In *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, January 2009. AIAA 2009-403.

[59] Z. J. Wang and H. Gao. A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids. *Journal of Computational Physics*, 228(21):8161–8186, November 2009.

[60] T. Haga, H. Gao, and Z. J. Wang. A high-order unifying discontinuous formulation for the Navier-Stokes equations on 3D mixed grids. *Mathematical Modelling of Natural Phenomena*, 6(3):28–56, 2011.

[61] J. Watkins, J. Romero, and A. Jameson. Multi-GPU implicit time stepping for high-order methods on unstructured grids. In *46th AIAA Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, June 2016. AIAA 2016-3965.

[62] B. J. Zimmerman and Z. J. Wang. The efficient implementation of correction procedure via reconstruction with GPU computing. In *21st AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, June 2016. AIAA 2013-2692.

[63] Z. J. Wang, Y. Li, F. Jia, G. M. Laskowski, J. Kopriva, U. Paliath, and R. Bhaskaran. Towards industrial large eddy simulation using the FR/CPR method. *Computers and Fluids*, 156:579–589, October 2017.

[64] K. Asthana and A. Jameson. High-order flux reconstruction schemes with minimal dispersion and dissipation. *Journal of Scientific Computing*, 62:913–944, 2015.

[65] C. A. Kennedy, M. H. Carpenter, and R. M. Lewis. Low-storage, explicit Runge-Kutta schemes for the compressible Navier-Stokes equations. Technical Report 209349, NASA, June 1999.

[66] R. D. Stieger. *The effects of wakes on separating boundary layers in low pressure turbines*. PhD thesis, Cambridge University, February 2002.

[67] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009. URL = https://gmsh.info/.

[68] C. Pereira and B. C. Vermeire. Spectral properties of high-order element types for implicit large eddy simulation. *Journal of Scientific Computing*, 85(48), November 2020.

[69] J.-R. Carlson. Inflow/outflow boundary conditions with application to FUN3D. Technical Report 217181, NASA, October 2011.

[70] A. L. Marsden, M. Wang, J. E. Dennis Jr., and P. Moin. Trailing-edge noise reduction using derivative-free optimization and large-eddy simulation. *Journal of Fluid Mechanics*, 572:13–36, July 2006.

[71] C. Audet, G. Savard, and W. Zghal. A mesh adaptive direct search algorithm for multiobjective optimization. *European Journal of Operational Research*, 204:545–556, November 2009.

[72] A. Koshakji. Free form deformation technique for 3D shape optimization problems. Master's thesis, Politecnico Di Milan, 2010.

[73] B. Galvan, D. Greiner, J. Periaux, M. Sefrioui, and G. Winter. Parallel evolutionary computation for solving complex CFD optimization problems : a review and some nozzle applications. In *Parallel Computational Fluid Dynamics 2002*, pages 573–604, 2003.

[74] A. Garai, L. Diosady, S. Murman, and N. Madavan. DNS of flow in a low-pressure turbine cascade using a discontinuous-Galerkin spectral-element method, GT2015-42773. In *Proceedings of ASME Turbo Expo 2015: Turbine Technical Conference and Exposition*, June 2015.

[75] S. L. Dixon and C. A. Hall. *Fluid mechanics, thermodynamics of turbomachinery*. Butterworth-Heinemann College, 7th edition, November 2013.

[76] J. S. Bendat and A. G. Piersol. *Random data : Analysis and Measurement Procedures*. Wiley, fourth edition, 2010.

[77] W. Haase, M. Braza, and A. Revell. *DESider - A European effort on hybrid RANS-LES modelling*, volume 103. Springer, 2009.

[78] Beluga, March 2019. URL = https://docs.computecanada.ca/wiki/Beluga/en.

[79] NVIDIA. *NVIDIA Tesla P100 GPU accelerator data sheet*, October 2016.

[80] NVIDIA. *NVIDIA V100 tensor core GPU data sheet*, January 2020.