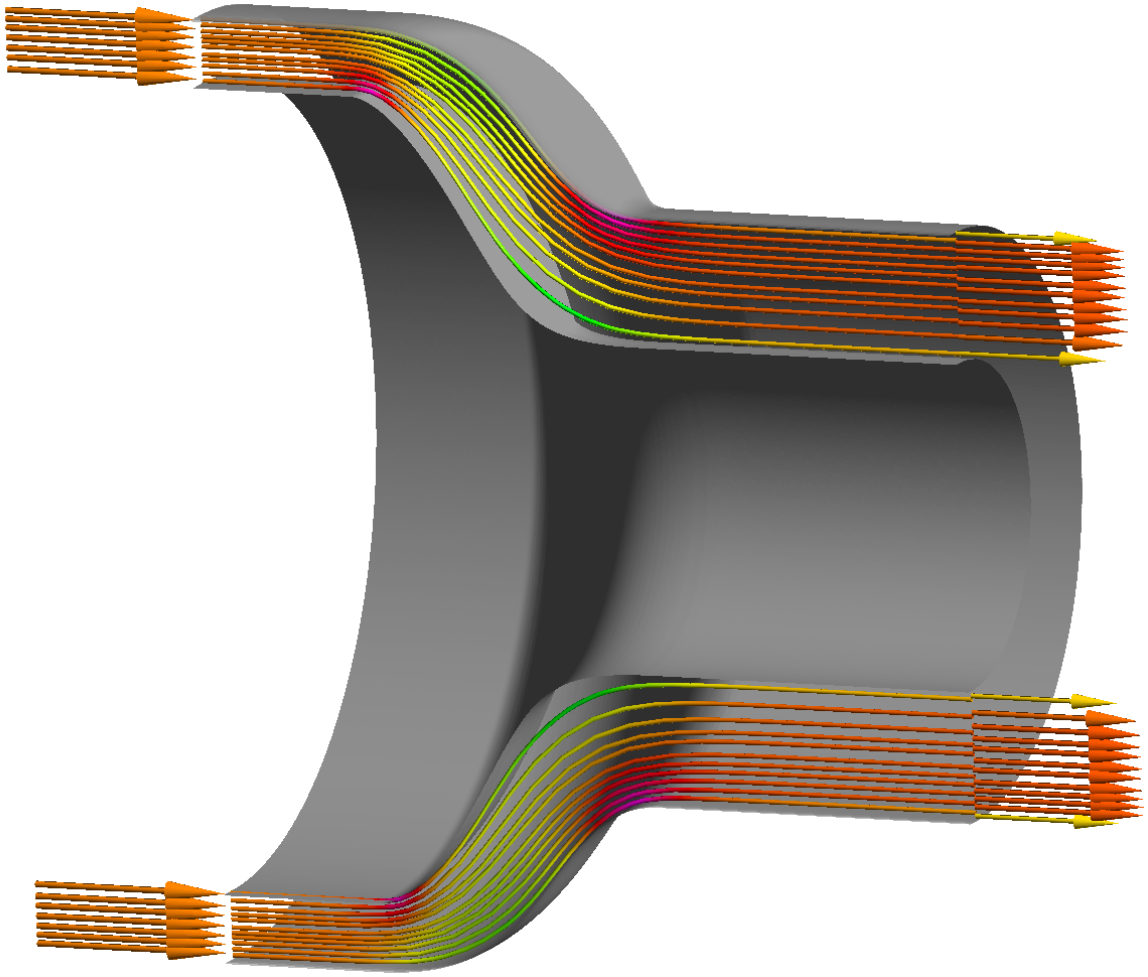


Two Dimensional Axisymmetric Optimization of Intermediate Compressor Duct

ROBERT HESSMAN RANMAN

Department of Mechanics and Maritime Science

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

Bachelor´s Thesis 2021

Two Dimensional Axisymmetric Optimization of Intermediate Compressor Duct

ROBERT HESSMAN RANMAN

CHALMERS

Department of Mechanics and Maritime Science
Division of Fluid Dynamics
Chalmers University of Technology
Gothenburg, Sweden 2021

Two Dimensional Axisymmetric Optimization of
Intermediate Compressor Duct
ROBERT HESSMAN RANMAN

Supervisor: Isak jonsson, Oliver Sjögren, Vinícius Tavares Silva, Applied Mechanics
Examiner: Carlos Xisto, Applied Mechanics

Bachelor´s Thesis 2021
Department of Applied Mechanics
Division of Fluid Dynamics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: ICD flow path.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2021

Two dimensional axisymmetric optimization of intermediate compressor duct
Department of Mechanics and Maritime Science
Robert Hessman Ranman

Chalmers University of Technology

Abstract

The civil aviation industry is a well established option for fast and secure intercontinental transport of people and freight. The continuous use of aerospace propulsion as a modern day means of transportation contributes to the emissions of harmful gases that in excess contributes to the decay of the environment. Engineers are continuously making progress in optimizing the gas turbine engines used for aerospace propulsion in order to reduce the fuel consumption. There has been great progress over the years since the gas turbines was first introduced in aerospace propulsion during the second world war. The first gas turbine engines on aircraft's where very loud and not very efficient in terms of reliability, lifespan and fuel consumption. The initial application for these engines where on military aircraft's as it would take some time before it was adapted by civil aviation, replacing the piston propeller engines. The turbofan engine is currently widely used by civil aviation for its propulsive efficiency with a high bypass and pressure ratio. The progress made over the last two centuries has been greatly affected by the invention of commercial computational fluid dynamics software's. Numeric optimizations can now be coupled with CFD software's that allows for optimization of complete modules and parts of the engine. This method aids engineers in reducing the wight and optimizing the performance on a component basis of the turbofan engine in order to reduce the emissions of harmful greenhouse gases.

Keywords: ICD, Fluent, ICEM, Genetic Algorithm (GA), Python.

Acknowledgements

I would like to start of by giving my appreciations to Isak Jonsson for giving me the opportunity to work on this thesis. Isak works as a Research Engineer at Mechanics and Maritime Sciences, Division of Fluid Dynamics, Chalmers university of Technology. Isak offered great guidance along the duration of this thesis and always contributed with useful knowledge and experience. Isak has been a great source of inspiration and i am thankful for the knowledge he shared.

Oliver Sjögren, PhD student at Mechanics and Maritime Sciences, Division of Fluid Dynamics, Chalmers university of Technology, offered useful knowledge and guidance in optimizing the ICD. Oliver also helped with setting up the geometric components for the ICD with computational considerations for the optimization. I am tankful for the knowledge and guidance he offered along the duration of this thesis.

Carlos Xisto, Senior Researcher at Mechanics and Maritime Sciences, Division of Fluid Dynamics, Chalmers university of Technology, held a weakly meting that helped with all aspects of this thesis. Carlos helped by offering guidance and experience along the duration of this thesis. I am tankful for the help he offered and the experience he shared.

Vinísius Tavares Silva, PhD student at Mechanics and Maritime Sciences, Division of Fluid Dynamics, Chalmers university of Technology. Vinísius helped with the CFD part of this thesis and especially with setting up the ICEM and Fluent scripting for optimization. I am tankful for the help he offered and the material he shared to help along the way.

This thesis and all the surrounding people behind it has been a great and inspiring part of my time at Chalmers University of Technology. The great deal of knowledge i have obtained during this time has prepared me well for my Masters program in Applied Mechanics. Therefore i would like to thank all the people that has been involved in this thesis for sharing their knowledge an experience.

Robert Hessman Ranman, Gothenburg, June 2021

Nomenclature

Roman lower case letters

μ_t	Turbulent viscosity
ω	Specific dissipation
τ	Shear Stress
ε	Dissipation
k	Kinetic Energy
p	pressure
q	Heat Flux
t	time
u, v	Velocity
$y+$	Dimensionless vertical distance (wall variable)

Letter combinations

CR	Crossover Rate
GA	Genetic Algorithm
HPC	High Pressure Compressor
HPT	High Pressure Turbine
ICD	Intermediate Compressor Duct
LPC	Low Pressure Compressor
LPT	Low Pressure Turbine
OGV	Outlet Guiding Vanes
PC	Mutation Rate
RANS	Raynolds Average Navier-Stokes

Roman upper case letters

D	dimensions
$N_{samples}$	number of sampling points
P	Pressure
Q_{level}	quantization level
S	Entropy

S	source term
T	temperature
T_t	time range
U	flow field

Contents

List of Figures

List of Tables

1
Introduction

Air travel is a well established and efficient concept for transporting people and freight all around the world. The increasing global usage of aviation has lead to an increase in emissions of harmful greenhouse gases [1]. For intercontinental travel to keep existing in the future with minimal environmental impact is the flight industry trying to optimize the performance and weight of the turbofan engine. There has been great progress over the years since the gas turbines was first introduced in aerospace propulsion during the second world war [3]. The first gas turbine engines on aircraft's where very loud and not very efficient in terms of reliability, lifespan and fuel consumption. The initial application for these engines where on military aircraft's as it would take some time before it was adapted by civil aviation, replacing the piston propeller engines. In modern times are the turbofans engines widely used by the aviation industry for it's high propulsive efficiency and compact design. Modern turbofan engines work on a high bypass- and pressure-ratio that allows for a reduced specific fuel consumption. Modern tools such as Computational Fluid Dynamics, CFD, has made a significant contribution in aiding engineers to achieve further improvements. CFD allows engineers to obtain complex numerical approximations on most part of the engines, which allows for optimization on individual parts and complete modules.

1.1 Intermediate compressor duct

The Intermediate (interconnecting) compressor duct (ICD) is utilised for guiding the high mass-flow form the large exit radius of the low pressure compressor (LPC), to the smaller inlet radius of the high pressure compressor (HPC) [2]. As losses in the system affects the overall efficiency and specific fuel consumption is the ICD optimized for length and total pressure loss in order to reduce the wight and increase the efficiency.

1.2 Objective

The objective of this thesis is to optimize and investigate the effects of decreasing the axial length of the ICD in regards to total pressure loss. This objective can be divided in four sub-tasks,

1. Optimization framework
2. CFD

3. Optimization
4. Post Evaluations

The first task involves writing the code necessary to construct the ICD in an computation efficient way for the optimization framework. The geometric components is to be expressed so it is efficient for the user and the optimization framework to deal with constraints and easy manipulation of geometric properties.

The second stage of this thesis revolves around creating a reliable CFD environment that is compatible with the optimization framework. This involves scripting and studies to evaluate the overall performance of the work done in this part.

The third stage involves connecting previous work and running the optimization in order to obtain necessary data.

The fourth and final step then involves extracting and analysing the obtained result.

1.3 Literature review

A similar study has previously been done at the university of Cambridge [4] carried out over three defined ICD lengths. The optimization was carried out in regards to minimum stagnation pressure loss with a one-degree of freedom design methodology. The findings was that the gross pressure losses occur as an result of the concave and convex curves of the inner and outer. The curvature was found to create adverse pressure gradients separating the boundary layers from the wall thus resulting in great losses. Reducing the ICD by 26% was found to only have a small impact of the overall net losses, but reducing it by 36% resulted in great losses. An other similar study was conducted at the German aerospace center, by T.Stürzebecher [5]. The ICD was optimized using an meta-model coupled with an Evolutionary algorithm (EA) for multi-objective optimization in regards to length and entropy raise coefficient. The inner and outer wall of the ICD was constructed using parametric curves with control points allowed to be varied in radial direction. In this setup the LPC ad the ICD was quipped with outlet guiding vanes (OGV) and a strut was placed in streamline direction in the ICD. The funding's was that it is feasible to reduce the ICD by 42% from the conventional design but with high risk of fluid flow separation.

1.4 Scope of the work

The aim of this thesis is to build a dynamic optimization framework in Python and Ansys for investigation of generated loss with varying axial length of an S-shaped ICD located in turbofan engines, restricted to steady state axisymmetric 2D simulations.

2

Aerodynamic optimization of compressor interconnecting ducts

2.1 Governing Equations

Fluid mechanics deals with the motion of fluids (liquids and gases), induced by external forces. Fluid flow can be modeled by partial differential equations (PDE), describing the conservation of mass, momentum and energy (commonly Navier-Stokes). In order to efficiently deal with these PDE's is computational fluid dynamics (CFD) utilised. CFD programming discretizise the PDE's and solve them using computers. Several commercial CFD software's exist and they are used in different engineering fields, e.g. aerospace and mechanical. In this chapter the governing equations used when solving turbulent flows are first explained. Last, turbulence modelling is explained.

2.1.1 Navier-Stokes

In order solve fluid flow problems one can treat the fluid as a control volume (Eularian fields) of a region or a point source (Lagrangian displacement field). If the fluid is treated in the Eularian frame of reference the following equations for conservation of mass, momentum and energy, known as the Navier-Stokes equations [6], can be derived on compressible form as,

Continuity,

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho \mathbf{u}) = 0 \tag{2.1}$$

Momentum,

$$\begin{aligned} \tfrac{\partial(\rho u)}{\partial t} + \text{div}(\rho u \mathbf{u}) = & \quad -\tfrac{\partial p}{\partial x} + \text{div}(\mu \text{ grad } u) + S_{Mx} \\ \tfrac{\partial(\rho v)}{\partial t} + \text{div}(\rho v \mathbf{u}) = & \quad -\tfrac{\partial p}{\partial y} + \text{div}(\mu \text{ grad } v) + S_{My} \\ \tfrac{\partial(\rho w)}{\partial t} + \text{div}(\rho w \mathbf{u}) = & \quad -\tfrac{\partial p}{\partial z} + \text{div}(\mu \text{ grad } w) + S_{Mz} \end{aligned} \tag{2.2}$$

Energy,

$$\frac{\partial(pi)}{\partial t} + \text{div}(\rho i \mathbf{u}) = -p \text{ div } \mathbf{u} + \text{div}(k \text{ grad } T) + \Phi + S_i \tag{2.3}$$

Where S_M is the momentum source and Φ the dissipation function of the energy source. These are the second order non-linear PDE's (commonly) used in CFD on compressible form for a Newtonian fluid. The equations are notoriously difficult to solve and are therefore discrtisized throughout the domain and solved with various interpolation techniques for the grid

2.1.1.1 Reynolds average Navier-Stokes

For turbulent flow the equations are harder to approximate, because of the fluctuation. The pressure and velocity at every point is a rapidly varying function of time and space. With the computational resources available for this study, resolving such fluctuation with enough accuracy would not be feasible. From an engineering standpoint the main interest is in mean values of velocity, pressure, shear stress and the high Reynolds number present in turbulent flow. In order to express the mean of the Navier-Stokes equations with respect to time, Reynolds decomposition can be employed. The Reynolds average [7] of a turbulent function $u(x, y, z, t)$, donated $\overline{u}$,

$$\overline{u} = \frac{1}{T_t} \int_0^{T_t} u \, dt \tag{2.4}$$

Where T is an averaging time period longer then the time of the fluctuation. The fluctuation u' is defined as the deviation of u from its average value,

$$u' = u - \overline{u} \tag{2.5}$$

if taking the mean value of a fluctuation, it follows by definition that a fluctuation has a mean value of zero,

$$\overline{u'} = \frac{1}{T_t} \int_0^{T_t} (u - \overline{u}) \, dt = \overline{u} - \overline{u} = 0 \tag{2.6}$$

The intensity of the turbulence are measured by taking the mean square of a fluctuation,

$$\overline{u'^2} = \frac{1}{T_t} \int_0^{T_t} (u')^2 \, dt \neq 0 \tag{2.7}$$

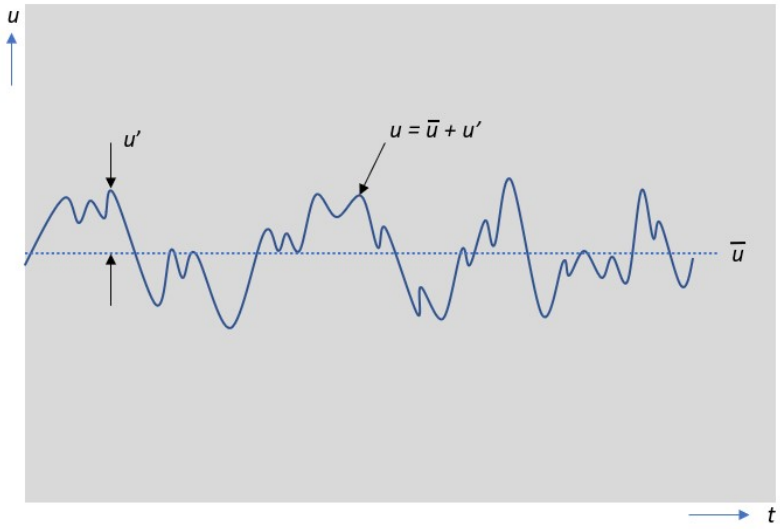


Figure 2.1: Mean and fluctuation variables of Turbulent flow.

Applying these principals to Navier-stokes equation generate the Raynolds time averaged Navier-Stokes equations (RANS). The time average continuity equation and Navier-Stokes equation can be written as, assuming incompressible flow [6],
Continuity,

$$\mathrm{div}\overline{u} = 0 \tag{2.8}$$

Momentum,

$$\begin{aligned}
\frac{\partial \overline{u}}{\partial t} + \mathrm{div}(\overline{u}\overline{\mathbf{u}}) + \mathrm{div}(\overline{u'\mathbf{u}'}) &= \frac{1}{\rho}\frac{\partial P}{\partial x} + v\mathrm{div}(\mathrm{grad}(\overline{u})) \\
\frac{\partial \overline{v}}{\partial t} + \mathrm{div}(\overline{v}\overline{\mathbf{u}}) + \mathrm{div}(\overline{v'\mathbf{u}'}) &= \frac{1}{\rho}\frac{\partial P}{\partial y} + v\mathrm{div}(\mathrm{grad}(\overline{v})) \\
\frac{\partial \overline{w}}{\partial t} + \mathrm{div}(\overline{w}\overline{\mathbf{u}}) + \mathrm{div}(\overline{w'\mathbf{u}'}) &= \frac{1}{\rho}\frac{\partial P}{\partial z} + v\mathrm{div}(\mathrm{grad}(\overline{w}))
\end{aligned} \tag{2.9}$$

Further generating data for mean fields of the turbulent flow regime. This procedure introduce a new unclosed term called Reynolds stresses, thus inducing the need for modelling in order to solve the equations (covered in next section).

2.1.2 Turbulence modelling

Turbulence modelling is utilised in order to solve previously defined RANS equations i.e. modelling of the Reynolds stress tensor. In this section three specific methods will be covered to fully motivate the final choice.

2.1.2.1 standard $k - \varepsilon$ model

The standard $k - \varepsilon$ model [8] solves for the turbulent kinetic energy, k and the dissipation, ε. This model is efficient for solving planar shear layer and recirculating

flow. The main assumption in this model is that the turbulent viscosity is isotropic, i.e. Reynolds stress and mean rate of deformation is equal in every direction. This model is well suited for modeling the free-stream flow for its non-sensitivity in regards to assumed dissipation. The turbulent viscosity otherwise know as eddy viscosity is computed by combining k and ε.

$$\mu_t = \rho \mathbf{C}_\mu \frac{k^2}{\varepsilon} \tag{2.10}$$

The equations for turbulent kinetic energy and dissipation rate then reads,

Turbulent kinetic energy,

$$\rho \frac{\partial k}{\partial t} + \rho \overline{u_j} \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + \tau_{ij} \frac{\partial \overline{u_i}}{\partial x_j} - \rho \varepsilon \tag{2.11}$$

Dissipation,

$$\rho \frac{\partial \varepsilon}{\partial t} + \rho \overline{u_j} \frac{\partial \varepsilon}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(\mu - \frac{\mu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right] + C_{\varepsilon_1} \frac{\varepsilon}{k} \tau_{ij} \frac{\partial \overline{u_i}}{\partial x_j} - C_{\varepsilon_2} \rho \frac{\varepsilon^2}{k} \tag{2.12}$$

The equation is written with tensor (index) notation. The coefficients starting with C and σ are what's called closure coefficients. They are determined in a systematic manner applying a semi-empirical procedure with optimization. They are given by,

- $C_{\varepsilon_1} = 1.44$
- $C_{\varepsilon_2} = 1.92$
- $C_\mu = 0.09$
- $\sigma_k = 1.0$
- $\sigma_\varepsilon = 1.3$

2.1.2.2 standard $k - \omega$ model

The $k - \varepsilon$ model [8] solves for the specific rate of dissipation, rate of which destruction of turbulent kinetic energy (conversion to internal energy) occurs per unit time. The $k - \omega$ model solves for the specific rate at which that dissipation occurs, together with the turbulent kinetic energy k. This method is used for modeling the turbulent viscosity and provides a near-wall treatment for low-Re number without involving complex non-linear damping functions. The eddy viscosity is computed by combining k and ω. The equation reads,

$$\mu_t = \frac{\rho k}{\omega} \tag{2.13}$$

The equations for turbulent kinetic energy k and specific rate of dissipation ω with tensor notation reads,

Turbulent kinetic energy,

$$\rho \frac{\partial k}{\partial t} + \rho \overline{u_j} \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + \tau_{ij} \frac{\partial \overline{u_i}}{\partial x_j} - \beta^* \rho k \omega \tag{2.14}$$

Dissipation rate,

$$\rho\frac{\partial\omega}{\partial t}+\rho\overline{u_j}\frac{\partial\omega}{\partial x_j}=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_\omega}\right)\frac{\partial\omega}{\partial x_j}\right]+\alpha\frac{\omega}{k}\tau_{ij}\frac{\partial\overline{u_i}}{\partial x_j}-\beta\rho\omega^2 \tag{2.15}$$

The closure coefficients are give by,

- $\alpha = 5/9$
- $\beta = 3/40$
- $\beta^* = 9/100$
- $\sigma_\omega = 1/2$
- $\sigma_k = 1/2$

2.1.2.3 $k-\omega$ Shear Stress Transport (SST) model

This model is a hybrid of the $k-\omega$ and $k-\varepsilon$ models [8]. It utilize a blending function F that activates the $k-\omega$ model near the wall, and the $k-\varepsilon$ in the free stream. The model is better at predicting the onset and amount of flow separation under adverse pressure gradients. The transport equations reds,

Turbulent kinetic energy,

$$\frac{\partial(\rho k)}{\partial t}+\frac{\partial}{\partial x_j}(\rho\overline{u_j}k)=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]+\tau_{ij}\frac{\partial\overline{u_i}}{\partial x_j}-\beta'\rho k\omega \tag{2.16}$$

Dissipation rate,

$$\frac{\partial(\rho\omega)}{\partial t}+\frac{\partial}{\partial x_j}(\rho\overline{u_j}\omega)=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_\omega}\right)\frac{\partial\omega}{\partial x_j}\right]+\alpha\frac{\omega}{k}\tau_{ij}\frac{\partial\overline{u_i}}{\partial x_j}-\beta\rho\omega^2+2(1-F)\frac{\rho\sigma_{\omega_2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j} \tag{2.17}$$

Where F is the blending function.

2.2 Parametric curves

A regular curve in $\mathbb{R}^2$ is often defined by two variables, x and y. A parametric curve has its x and y values expressed in terms of another variable. The Bézier curve is defined by the variable $t \in [0, 1]$ which evaluates the curve in relation to an independent variable. The parametric value t is simply a variable substitution that evaluates each x and y for the curve function. This method allows for redefinition of a curve in desired variable(s).

2.2.1 Bézier curves

A Bézier curve is a parametric curve that utilises the Bernstein polynomial as its base [9]. The curve is represented by a sequence of control points that defines the order (n) of the curve, equal to the amount of control points minus one. This method is widely used in computer graphics and all of its fields to generate easily manipulated smooth curves. Equation for a Bézier curve reads,

$$\mathbf{B}(t) = \sum_{i=0}^{n} \mathbf{B}_i^n(t) \cdot \mathbf{P}_i, \quad t \in [0, 1] \tag{2.18}$$

Where $\mathbf{P}$ is the control points and $\mathbf{B}_i^n(t)$ is the Bernstein polynomial:

$$\mathbf{B}_i^n = \binom{n}{i} t^i (1-t)^{n-i}, \quad \binom{n}{i} = \frac{n!}{i!(n-i)!} \tag{2.19}$$

A Bézier curve is only guaranteed to pass through its first and last control points, P_0 and P_n. The final shape of the curve can thereby be difficult to predict. Understanding the geometrical construction of the curve can help to visualise the final outcome.

A cubic Bézier curve is represented by four control points P_0, P_1, P_2 and P_3 [10]. First step to generate a Bézier curve is to connect each control point in numerical order. This creates three new lines $\overline{P_0P_1}$, $\overline{P_1P_2}$ and $\overline{P_2P_3}$ (see Fig 2.2a). The length of these lines are assigned the dimensionless value one, represented by t i eq. (2.18). In this example t is set to be 0.25. This entails that at 25% of the distance from the lower index control point three new points are created, Q_1, Q_2 and Q_3. These three new points are connected in the same way as the control points resulting in two new lines, $\overline{Q_1Q_2}$ and $\overline{Q_2Q_3}$ (see Fig 2.2b). Equivalent to previous method are two new points created, S_0 and S_1 at a distance t% from the lower index point. These two new points are connected by $\overline{S_0S_1}$ (see Fig 2.2c)) and on this final line at a distance of t% from the lower index point the equation is evaluated at $B(0.25)$ (see Fig 2.2d).

When allowing t to vary in its full span from 0 to 1 each of the corresponding $B(t)$ is evaluated and the entire curve is obtained. As seen in Fig 2.2e, the only control points that the curve passes through are the first and last, the rest of the evaluated points shapes the curve (see in Fig 2.2f).

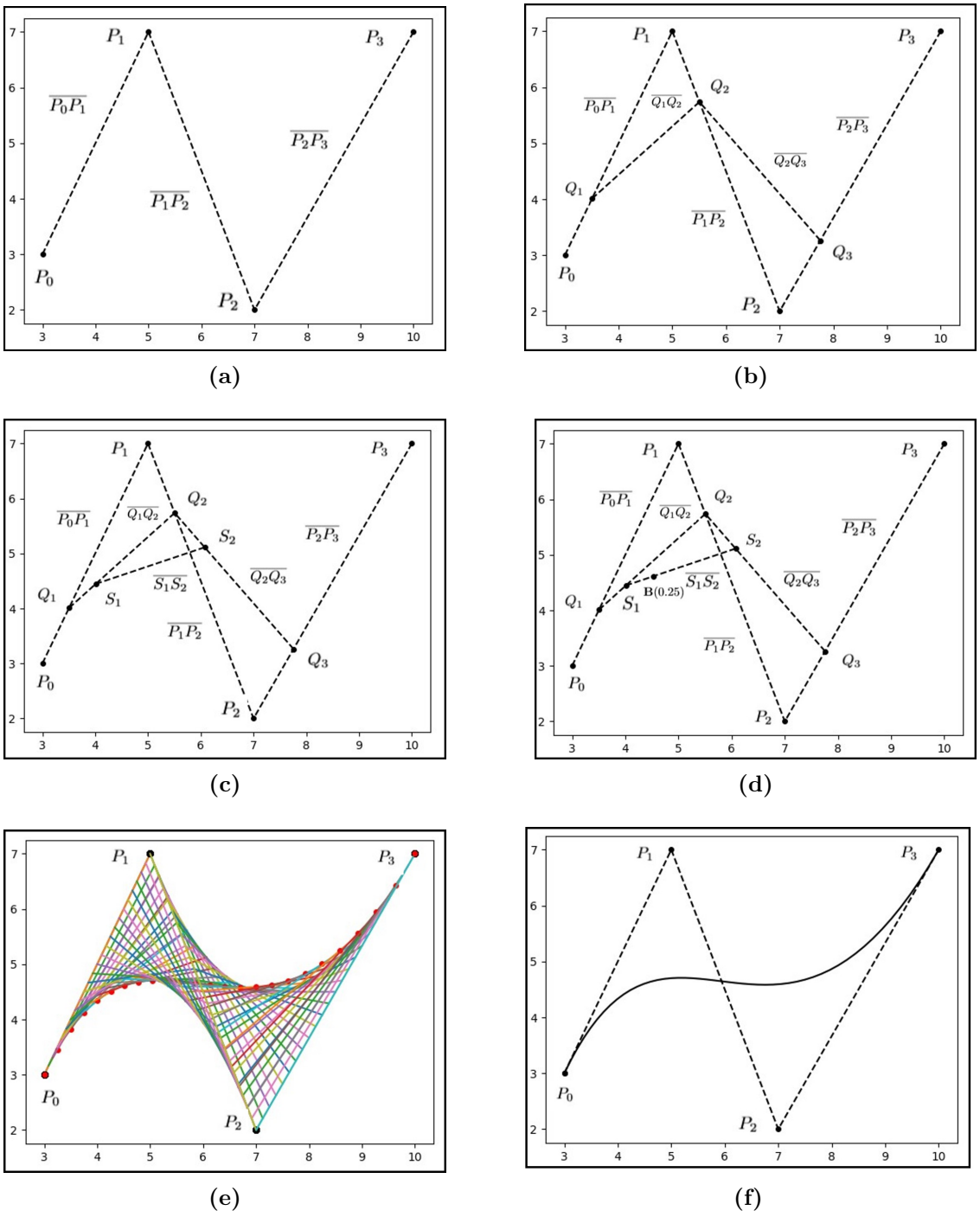


Figure 2.2: Geometric Bézier curve.

The same principal is applied to all Bézier curves regardless of the order. Application of these types of curves makes it easy to manipulate the shape by simply moving the defined control points.

2.3 Stochastic Optimization

In this chapter on Stochastic optimization algorithms (SOA) the basics when dealing with genetic algorithms are explained and covered for the reader and a number of different methods for specific parts are explained.

2.3.1 Genetic Algorithm

Genetic algorithm (GA) mimics natural selection by the means of inheritance to solve an optimization problem [11]. The variables are encoded in strings of ones and zeros (binary encoding) called chromosomes. Each one or zero that make up the chromosomes is known as a Bit or Gene and defines the accuracy, i.e. a higher number of bits is equivalent to a higher floating point accuracy. A set of variable values generated in one iteration is equivalent to an individual that together forms a population. The initial population is randomly chosen to cover the entire parametric space uniformly. Each generation after the initial population is based on its predecessors called parent population i.e, together a population p shapes the next generation $(p+1)$. Each individual in a population is assigned a fitness value that weighs the quality of the objective function. Fitness values are used to determine the best preforming individuals to shape the next generation, i.e. if the objective is to minimize a function value, a low fitness value donate the better performing individuals. This process is repeated until either max generations are achieved or objective function optimum is reached.

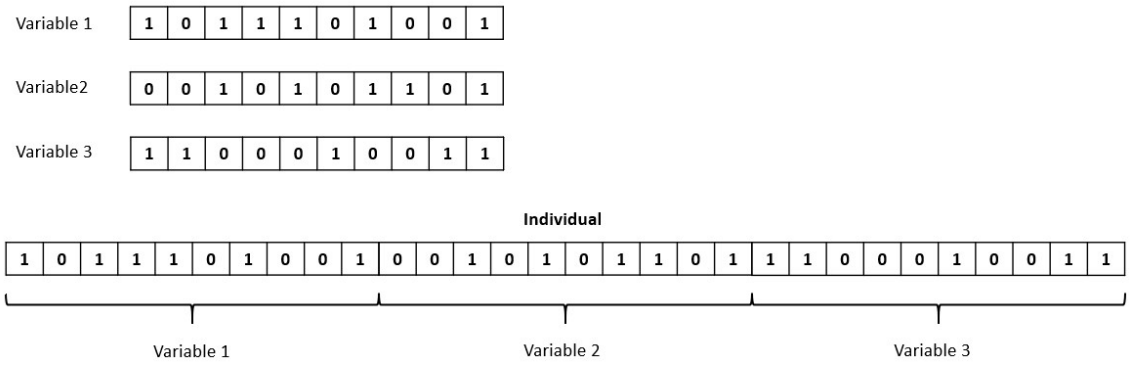


Figure 2.3: Chromosome for three variables, 10 bits.

The algorithm is executed with three main natural processes; selection, crossover and mutation. These are applied on each population to improve performance of population $(p+1)$.

2.3.1.1 Selection

Selection is the process in which the individuals from the previous population is selected for the current generation. There are different selection procedures implemented in GAs, and in this section roulette wheel, tournament, rank and elitism is covered.

Roulette wheel: In roulette wheel selection the probability for an individual to be chosen for breeding to the next generation is proportional to its fitness, i.e. better fitness is equivalent to higher chance for that individual to be chosen. This method can be seen as a roulette wheel with pockets the size of the probability to be chosen. The method does not exclude chosen individuals so the same individual can occur multiple times.

Tournament: Tournament can be seen as a battle between two candid individuals. Two randomly chosen individuals from the population are chosen to "battle" and the winner will be accepted for breading. The chance of winning is proportional to its fitness value. Winners are not excluded so the same individual can occur multiple times.

Rank: Rank works with negative fitness values and are often used in populations close to objective functions optimum. Towards the end of the run the individuals ends up with almost equal fitness values which entails that all individuals have approximately the same probability of breading. The individuals are ranked based on their fitness value which puts their probability for breading in direct proportion to their fitness value.

Elitism: Elitism is no standalone selection method but rather works in parallel with other methods. This method utilise partial reproduction in which a small portion of the best performing individuals are carried over to the next generation without any changes.

2.3.1.2 Crossover

Crossover (CR) is a method that mimics the process of natural reproduction. During this process the genetic material for two individuals are used to reproduce children. The binary encoding for two individuals are combined to resemble offspring. There are different ways of carrying out crossover, in this section one-point crossover is explained. The genetic material for a individual consist of variables encoded by genes (ones and zeros). In crossovers a line at a random position split the parents chromosomes into sections. One section from each parent is then carried over to its children (See Fig 2.4). The resulting offspring is then a mix between the parents binary coding, see Fig 2.4b.

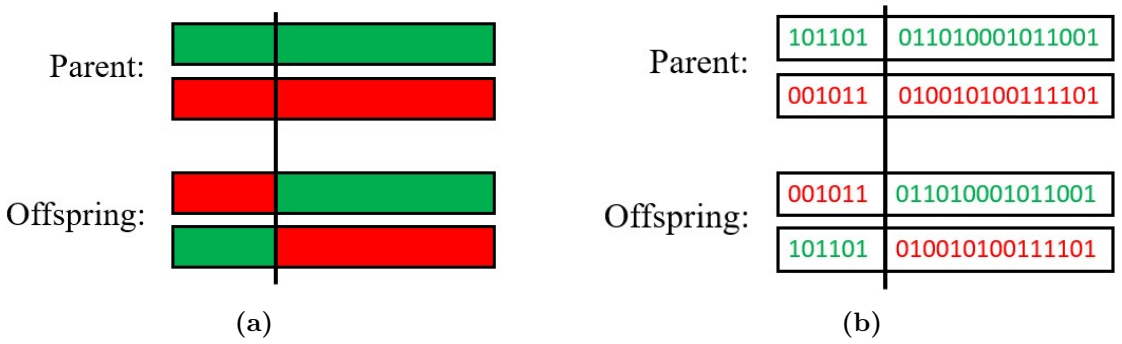


Figure 2.4: Illustration of one point crossover.

The parents are two consecutive individuals chosen from one of the selection methods mentioned in Sec 2.3.1.1. The reproduction takes place with a probability of CR $\in [0, 1]$.

2.3.1.3 Mutation

Mutation is a method that represent the way genes can mutate in nature. This process is mimicked by mutating the binary code for an individual by swapping 1 with 0 or, 0 with 1. The genes in an individual chromosome are checked one by one and the probability of mutation is determined by PC $\in [0, 1]$ (see Fig 2.5). The probability is usually set to $P_{mutation} = 1/m$, where m represents the amount of genes (bits). If the crossover is successful the offspring will undergo mutation, otherwise it is applied on the individuals resulting from the selection process.

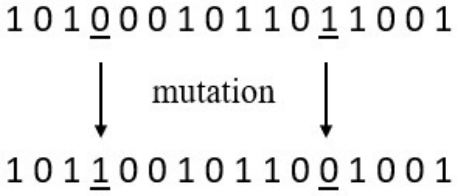


Figure 2.5: Illustration of chromosome mutation.

2.3.1.4 GA flow chart

Flow chart illustrating a genetic algorithm and how it utilizes different natural processes to achieve objective optimum. The algorithm either stops if maximum generations or the best solution is obtained. Elitism is an optional setting and the algorithm will still work without that branch.

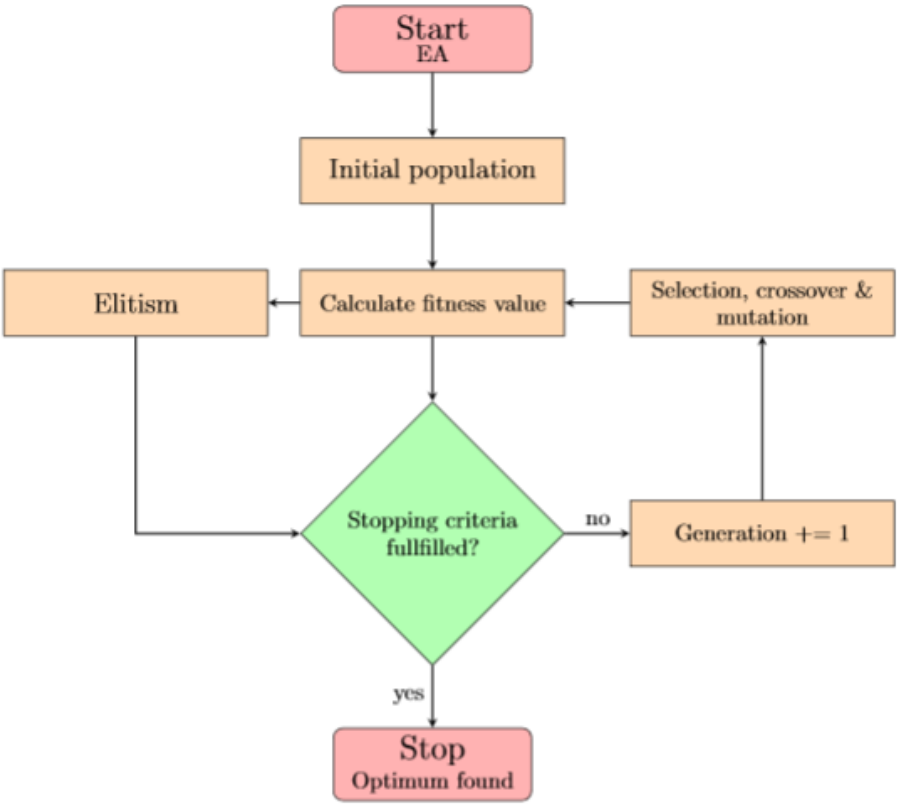


Figure 2.6: Flow chart of genetic algorithm.

2.3.1.5 Initial population

In evolutionary algorithms the initial population is used to define the parametric space made up by individuals. These individuals are made from encoded strings of different combinations of defined design variables. In order to find the global optimum of a function the variables have to be defined on the entire parametric space. If the initial population is too low the algorithm will not get a sufficient amount of combinations (individuals) to find the global optimum, i.e. the population does not cover the entire parametric space. The amount of design variables is equivalent to the amount of dimensions for the problem, donated D. The parametric space can be seen as a space containing a polynomial of second order in a parametric design space of D dimensions. In order to define this polynomial it takes a number of $N_{samples}$ points. This number defines the amount of initial sampling points needed to cover the entire parametric space, as long as the same point only occur once. The equation to obtain this value reads;

$$N_{samples} = \frac{(D+1)(D+2)}{2} \tag{2.20}$$

In optimization problems containing a large amount of dimensions this equation can be used as a guideline for choosing a large enough initial population.

2.3.1.6 Encoding and decoding bits

Each variable in the optimization is encoded in binary vectors translated from a continuous value. In order to encode and decode these variables the algorithm must have a way of translating value to binary strings and vice versa. One way of doing that is by using quantization. This method is mapping a continuous range of values into a finite set of values (samples), thus changing it from infinite precision to finite precision. Then it rounds of the sample to its nearest quantization level. Each quantization level is donated by its own binary string thus defining the variable value in that range in binary code. The amount of quantization levels is defined by the number of bits $Q_{level} = 2^{bits}$, i.e a higher bit count will result in a better finite precision thus getting a more accurate binary code. With a higher bit count the binary vector representing the variable will be more precise but also longer due to the added quantization levels, requiring more binary values to differentiate the levels.

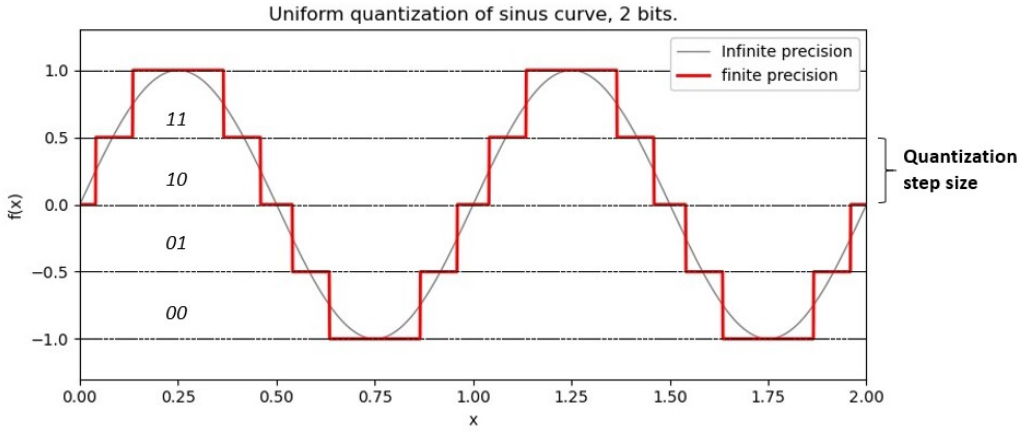


Figure 2.7: Quantization of sinous curve with 2 bits and binary code for each level.

As seen in Fig 2.7 each value for x corresponds to a function value that lies within a quntization level, i.e. discrete continuous intervals. If the number of bits where higher the span for each level would be lower and thereby would each binary code have a smaller function value span, i.e. higher precision equivalent to a lower quantization error. For encoding a variable p_n the equation reads.

$$p_{norm} = \frac{p_n - p_{min}}{p_{max} - p_{min}} \tag{2.21}$$

$$gene[m] = round\left(p_{norm} - 2^{-m} - \sum_{p=1}^{m-1} gene[p]2^{-p}\right) \tag{2.22}$$

for decoding,

$$p_{quant} = \sum_{m=1}^{N_{gene}} gene[m]2^{-m} + 2^{-(m+1)} \tag{2.23}$$

$$q_n = P_{quant}(p_{max} - p_{min}) + p_{min} \tag{2.24}$$

- p_{norm} = normalized variable
- p_{min} = smallest variable value within the quantization level
- p_{max} = highest varibale value within the quantization level
- $gene[m]$ = binary code for variable
- $round\{\cdot\}$ = nearest integer
- p_{quant} = quantized version of P_{norm}
- q_n = quantized version of p_n
- N_{gene} = number of genes

3

Optimization framework

The optimization framework is categorized into five main tasks illustrated in Fig 3.1. Optimization starts with an initial input of design variables and ends either if the maximum generation is achieved or the objective optimum converges to an optimal solution. The Optimization framework is setup with the open-source code openMDAO [14] with a genetic algorithm implementation.

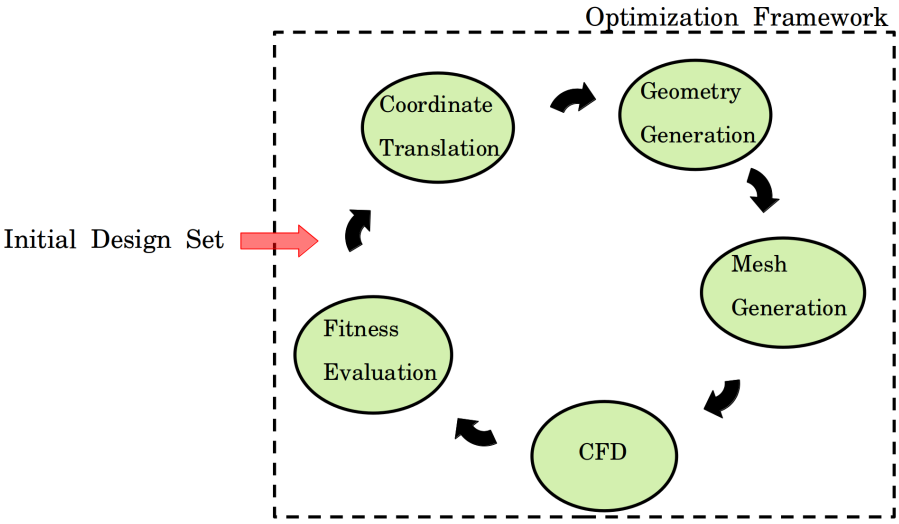


Figure 3.1: Optimization scheme.

The framework is executed in a multi-objective setup utilizing the pressure coefficient (Y_p) and uniformity index (U.I) as the objective functions. The framework is employed in a text based interface (TUI) for optimal performance on a LINUX based operating system. For additional time save the optimization is run in MPI allowing for a number of parallel executions of the algorithm. Further a Pareto front is used for the final evaluation and selection of optimal geometry. For this thesis, three ICD lengths were optimized and evaluated, categorized as conventional, aggressive, and very aggressive design.

ICD design range			
	Very aggressive	Aggressive	Conventional
ΔR/L	0.6-0.9	0.6	0.5

Table 3.1: ICD design range.

3.1 Optimization Parameters

The initial population is calculated to be a minimum of 55 individuals to efficiently search in the parametric space. The value of CR is set to 0.8 and the number of bits used is 28 for each design variable to achieve good accuracy. The mutation rate is set to 0.02 to introduce variation over the generations and to keep the algorithm from converging at a local optimum. Further elitism is employed with one individual carried over, and the optimization framework is set to run for a total of 100 iterations (generations).

3.2 Geometry definition

Geometric constraints of baseline ICD geometry are expressed utilizing parametric values from initial geometry definition. Utilizing the constraints for the area ratio, delta of inlet/outlet radius and the height of the two, the geometry can be defined with the following five expressions.

1. R1
2. L/R1
3. h1/R1
4. R2/R1
5. A2/A1

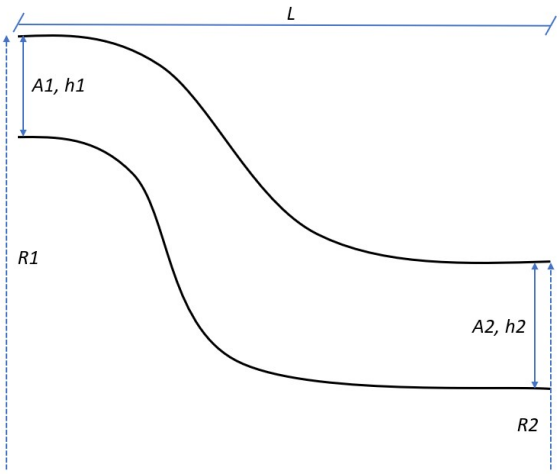


Figure 3.2: Geometric Definition.

For each ICD in the respective design range (table 3.1), these five values are treated as fixed and are utilized as the bounds for the definition of the Bezier control points for the hub and shroud. Each Bezier curve for the hub end shroud is defined by a total of five control points, as five was the minimum number of points found in testing to capture the shape of the ICD. Further additional axial sections are added to the duct at the inlet and outlet to allow the flow to stabilize before entering/exiting the duct. In further sections, subscribe "s" denotes shroud, and "h" denotes hub.

3.2.1 Axial control points

The first and last control points defining the hub and shroud are defined by calculating the length from fraction 2,

$$\frac{L}{R1} = \Gamma \Rightarrow L = \Gamma R1$$

The inlet and outlet axial coordinates can now be obtained as,

$$x_1 = \Gamma R1 - \Gamma R1 = 0$$
$$x_5 = \Gamma R1$$

Further yhe second, third, and fourth control points utilize a design value t that allows them to vary in a defined range constrained by the value of t, e.g. for the shroud,

$$x_2 = t \cdot \Gamma R1, \;\; t \in [0.1, 0.5]$$
$$x_3 = t \cdot \Gamma R1, \;\; t \in [0.2, 0.8]$$
$$x_4 = t \cdot \Gamma R1, \;\; t \in [0.5, 0.9]$$

As seen by the definition of the control points the axial coordinates x_2, x_3 and x_4 is not constrained by each other as this is the method that rendered the most stable algorithm. The definition of hub and shroud control points is equivalent but with a modified range of the variable t.

3.2.2 Radial coordinates

First and last radial position of the control points are defined utilising fraction 1, 3, 4 and 5. The top radius R1 defines the first radial coordinate for the shroud, equivalent coordinate for the hub is obtained by subtracting with the inlet height. Equivalent method is then used for the last hub and shroud control points. With this method the equations for the radial inlet and outlet coordinates reads,

$$y_1^s = R1$$

$$y_1^h = R2$$

$$y_5^s = R1 - h1$$

$$y_5^h = R2 - h2$$

The second and fourth control points radial value is constrained to a constant value. This is a necessary constraint in the algorithm as the gradient for the Bezier curves connecting to the inlet and outlet needs to be zero to minimize the effects of the added axial inlet and outlet sections. How a smooth transition is achieved is further covered in the following section. For the fourth radial coordinates of the control points the definition is independent of the other curve but constrained to its own control points. Equivalent to that of the axial control points a parametric value t

is utilized constrained to a predefined range. The definition of these points is such that it's always the previous point and is a function of ΔR_s and ΔR_h defined as the radial height between the inlet and outlet coordinates.

$$\begin{aligned} y_4^s &= y_3^s - (t \cdot \Delta R_s), \quad t \in [0.1, 0.9] \\ y_4^h &= y_3^h - (t \cdot \Delta R_h), \quad t \in [0.1, 0.9] \end{aligned} \tag{3.1}$$

This method es employed as it allows for easy manipulation of the bounds of the ICD, as the parametric value t defines the range of the coordinates and allows the user to manipulate for example how small the duct diameter is allowed to be.

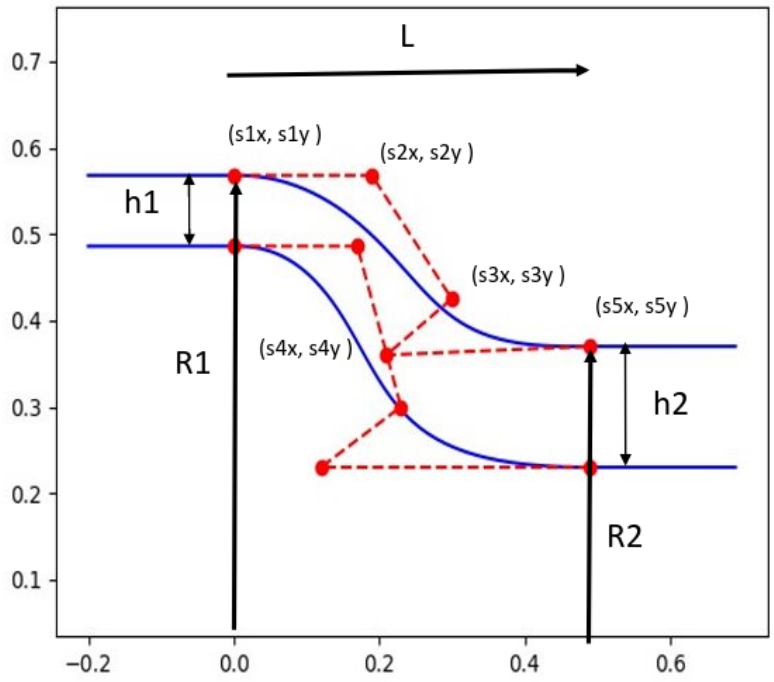


Figure 3.3: Shroud Control Points.

Figure 3.3 illustrates the shroud control points and variable declaration of its axial and radial coordinates.

3.3 Objective

The multi-objective optimization setup is evaluated according to two performance-defining parameters, pressure loss over the duct, and flow uniformity at the duct outlet.

3.3.1 Uniformity Index

Uniformity index (U.I) is a normalized value to the mean flow parameter at a selected point of evaluation.

$$U.I = \sqrt{\frac{\sum_{i=1}^{n} A_i (x_i - \overline{x})^2}{\sum_{i=1}^{n} A_i \overline{x}^2}}, \quad x = V, p_0 \tag{3.2}$$

Nodal data is normalized to the mean flow value of parameter along selected evaluation line or surface. The value can be used to determine disturbance in the flow due to upstream conditions in the flow regime.

3.3.2 Pressure loss, Y_p

Parameter defining normalized pressure loss over geometry. Found by calculating the pressure loss over any control volume and further normalizing to the entry condition.

$$Y_p = \frac{P_{0,\text{ inlet}} - P_{0,\text{ outlet}}}{P_{0,\text{ inlet}}} \tag{3.3}$$

Data obtained by averaging procedure of choice, e.g. area/mass average.

3.4 Mesh generation

Ansys ICEM is utilised for the meshing routine with its integrate scripting tool for further implementation in the optimization algorithm. A hexahedral mesh is generated in the domain with a mesh refinement close to the walls. The mesh is scaled to a resolution to capture a non-dimensional wall distance of $y+ \leq 1$ and satisfy convergence in the nomalized pressure loss parameter Y_p.

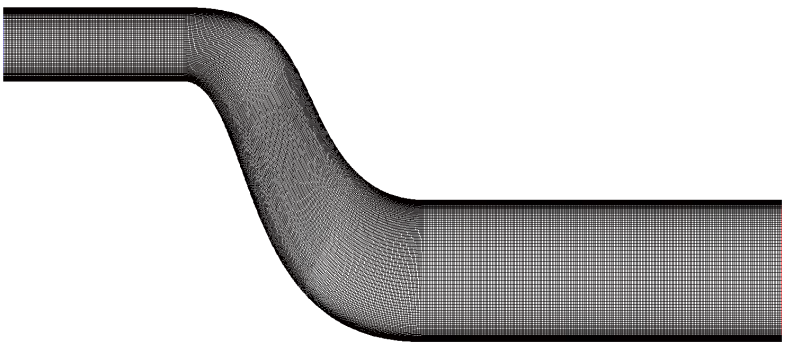


Figure 3.4: Mesh.

Figure 3.4 illustrates the final iteration of the mesh for a very aggressive ICD.

3.5 CFD

Numeric simulations of the ICD is carried out in Ansys Fluent coupled with the external python script that handles the optimization framework. This section covers the numerical methods used for simulation each ICD design and applied boundary conditions to reach a stable simulation procedure.

3.5.1 Numerical models

The simulation is carried out in an axisymmetric 2D spacial formulation of the governing equation with a steady temporal formulation. A pressure based formulation of the governing equations is used for the solver where the velocity field is obtained by solving the momentum equations for the pressure. This is carried out with a coupled algorithm for the pressure-velocity coupling, solving for the pressure and velocity fields in one matrix i.e no need for pressure correction in obtaining the velocity fields. A coupled solver extends the memory usage requirements of the system but decreases the convergence time as there is no need for pressure correction. The decision to use a coupled scheme is therefore weighted by a large pool of memory available on the system at hand and the need for time reduction as there is a large number of simulations to run. The spatial discretization used in the interpolation is carried out with a second order upwind scheme for density, momentum, and energy while the gradients are computed with least square cell based assuming linear variation of the field properties. Further, the k-omega SST turbulence model is applied to model the Reynolds stress due to turbulence and close the momentum equation.

3.5.2 Boundary conditions

The simulation setup is executed in the absolute frame of reference imposed by a reference pressure and temperature of zero. For the hub and shroud wall a no-slip condition is utilised generating zero velocity at the location of the wall reducing the need for frictional modelling at the wall. The inlet and outlet condition utilised is a pressure boundary, i.e. inducing movement in the fluid by a pressure gradient. In obtaining the correct delta in pressure the known value of inlet velocity and outlet pressure is utilised. Firstly assuming uniform velocity profile at the inlet the Mach number can be obtained as,

$$M = \frac{u}{a} = \frac{u}{\sqrt{\gamma RT}} \approx \frac{212}{398} = 0.53$$

Knowing the inlet Mach number the inlet pressure is adjusted until the correct Mach number is present at the inlet.

3.6 Scripting

Scripting is a text based file system that saves information in written command lines. The Ansys ICEM framework offers the ability for replay control in which each action done when initially meshing is saved in a text based format. The information can read,

```
ic_set_global geo_cad 0 toptol_userset
ic_set_global geo_cad 0.0005 toler
```

Figure 3.5: ICEM scripting command lines

The text-based interface is non-dynamic itself but makes for very easy manipulation by simply changing the information in one line. This method is utilized to render a new mesh for each generated geometry in the optimization, as only the generated ICD coordinates are to be replaced in the script file and executed by the algorithm to render a new mesh.

For CFD simulations in Ansys Fluent scripting as also used for communications between programs. Here manual scripting is used with Fluent internal commands, found in the Fluent user manual. The commands are first tested in the GUI and later copied to a text file saved in the journal format (*.jou). The information Fluent receives from the journal can read,

```
/define/models axisymmetric yes
/define/models/viscous/kw-sst yes
/define/boundary-conditions/velocity-inlet/inlet no no yes yes no 48 no 0 no no yes 5 10
/solve/monitors/residual/convergence-criteria 0.00001 0.00001 0.00001 0.00001 0.00001
```

Figure 3.6: Fluent scripting command lines

4
Results

This chapter will first present data from the grid Independence study on the ICD mesh, and lastly data from the optimization studies.

4.1 Grid Independence study

Grid Independence study is carried out on initial geometry for a conventional ICD design. The mesh is required to satisfy numerical convergence in the normalized pressure loss coefficient Y_p and render a Y^+ value close to or below 1 to render accurate results following the recommended requirements for the k-ω SST turbulence model.

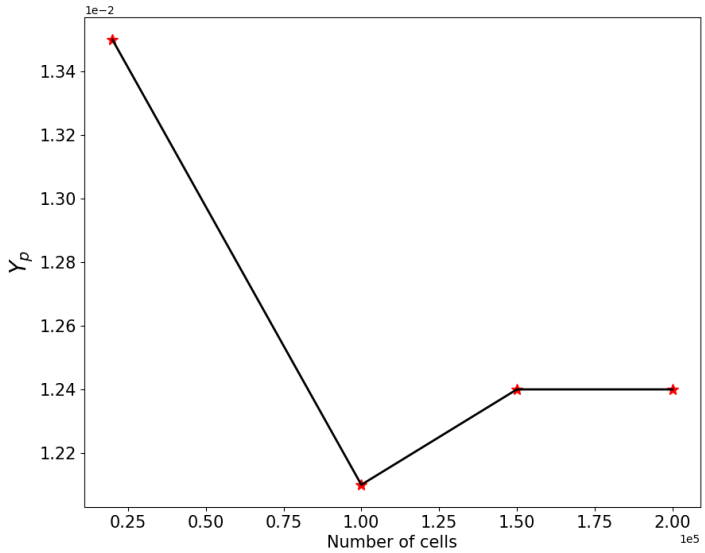


Figure 4.1: Cell count evaluation.

Figure 4.1 illustrates the normalized pressure loss coefficient as a function of the discretization of the ICD domain. The data is collected for equivalent geometry and boundary conditions with the varying factor limited to cell count. Convergence can be seen reached in a cell range of 150.000 - 200.000, indicating that the final mesh should lie within this range. With the cell count confined to this range, the Y^+ value can be evaluated and further minor adjustments to the mesh can be implemented at the walls to reach the desired Y^+ value.

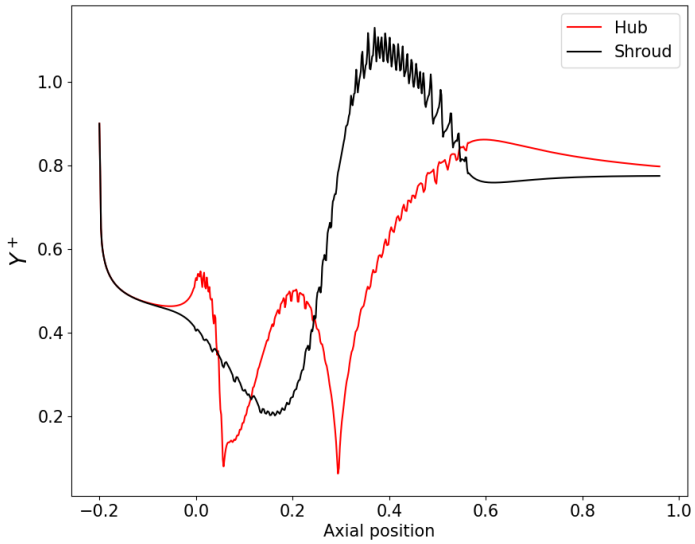
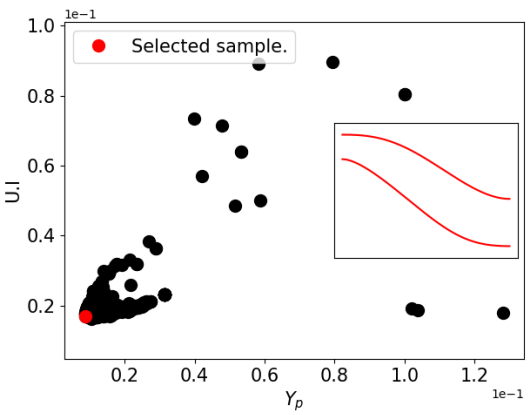


Figure 4.2: y^+ validation.

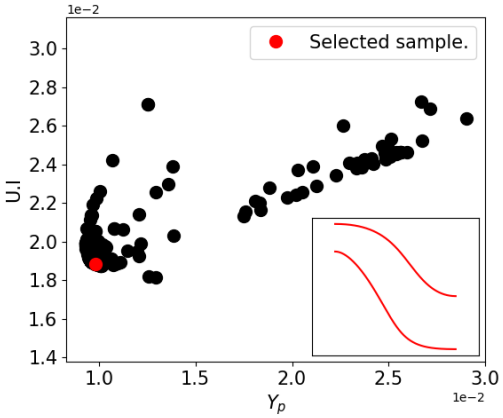
Figure 4.2 illustrates raw data of the non-dimensional parameter y^+ at the hub and shroud wall for the final iteration of the mesh. A local $y^+ \leq 1$ can be observed for greater parts of the walls, while at the shroud a local peak can be observed with a $y^+ > 1$.

4.2 Optimization studies

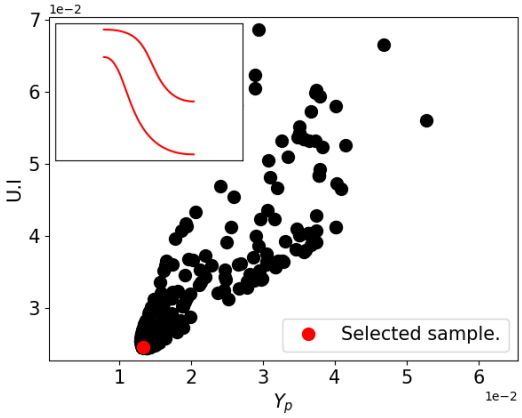
For the three ICD subcategories defined in table 3.1 one duct from each range is optimized in which one design is selected based on its shortest distance to the origin of a Pareto front.



(a) Conventional.



(b) Aggressive.



(c) Vary aggressive.

Figure 4.3: Pareto fronts.

Figure 4.3 illustrates the final generations of optimized ICD designs, with designs selected for further evaluation marked as red. Reducing the axial length of the ICD is expected to induce a more aggressive radial turning of the fluid thus increasing the overall losses in the system. This can be seen illustrated in the plots as both the optimized value for pressure loss coefficient and uniformity index increases.

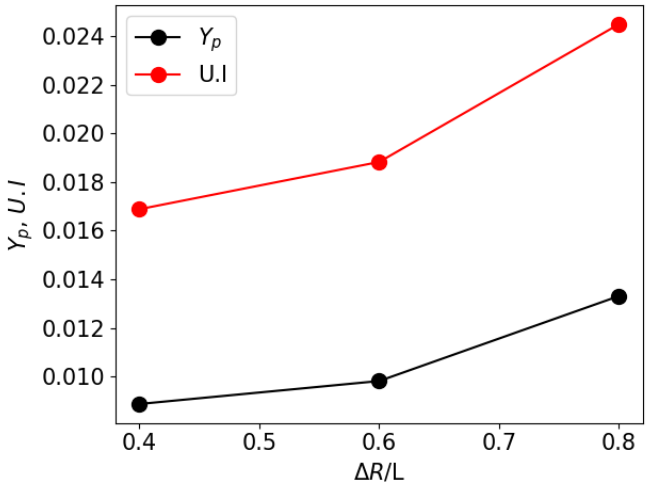


Figure 4.4: Sample evaluation.

Figure 4.4 illustrates the pressure loss coefficient and U.I as a function of ICD length. The data illustrated is that of the selected samples from figure 4.3. The gradient of the U.I line can be seen to be higher indicating that increased radial turning of the fluid in the more aggressive ducts induces a greater loss in flow uniformity compared to pressure loss. With the cases illustrated in fig.4.4, the static pressure data for each case along hub and shroud line is exported from fluent and further used to calculate the pressure coefficient, C_p, along each wall. The C_p values for each case are plotted on a normalized scale where the equivalent x-coordinate along hub and shroud line is divided by its equivalent ICD length, (x/L).

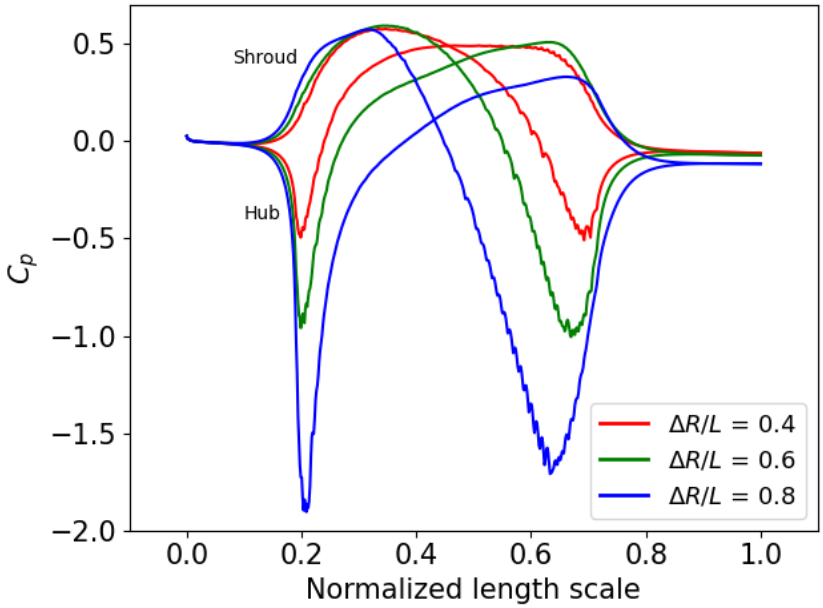
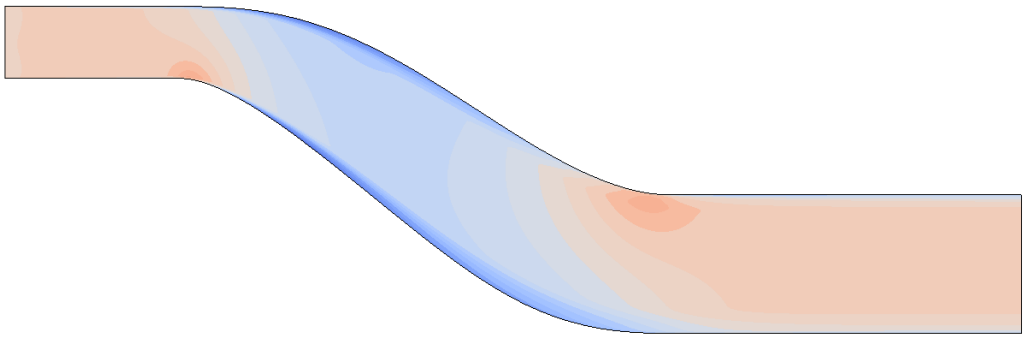


Figure 4.5: Pressure distribution.

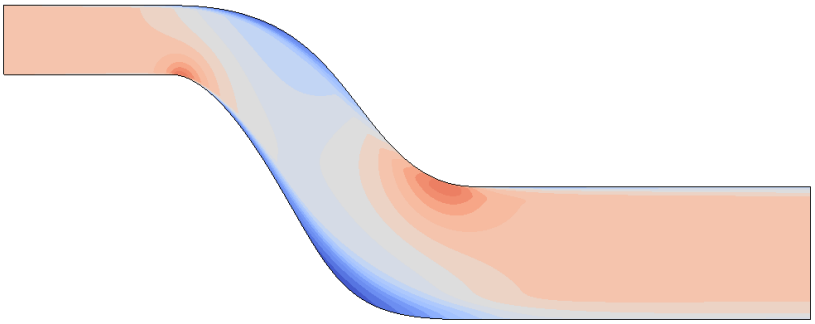
Figure 4.5 illustrates the pressure coefficient distribution at hub and shroud line for

optimized ICD geometries. A pressure coefficient equivalent to one would indicate a stagnation point as the pressure can never exceed the stagnation pressure, i.e. a point in the flow where the fluid becomes stagnant (stationary). In the figure, no stagnation points can be seen for either of the designs in the hub or shroud line.

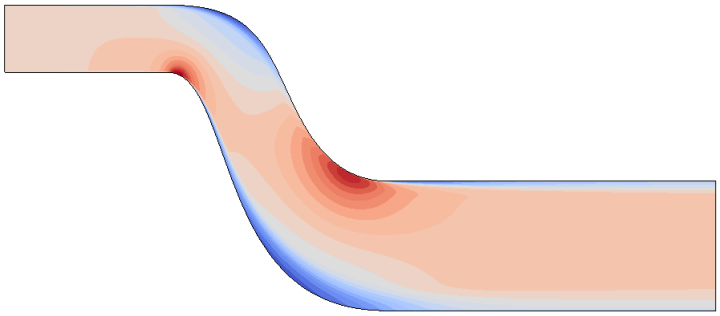
Following the hub lines, studying the pressure coefficient in figure 4.5, it can be seen that the flow is accelerated over the initial radial turn of the S-shaped duct, increasing in magnitude (absolute value of C_p) while the axial length is reduced. Further observing the static pressure recovery following the negative peak, both the conventional and aggressive designs can be seen to approach close to equivalent C_p (function of free stream static pressure), with the conventional design doing so further upstream. The axial reduction of the very-aggressive design can be observed to not allow for equivalent pressure recovery. Further studying the shroud lines, the flow can at the initial turn be observed to decelerate facing the wall, and further downstream accelerated over the second turn of the duct. The overall gradient approaching peak velocity over the second turn (maximum negative pressure coefficient) can be observed as lesser compared to hub lines, potentially due to upstream losses, increasing in magnitude with the reduction in axial length. Both negative peaks following the hub and shroud lines of the pressure coefficient indicate sensitive locations regarding flow separation of the wall adjacent flow, with an increasingly higher probability with a reduction of the axial length indicated by the plot.



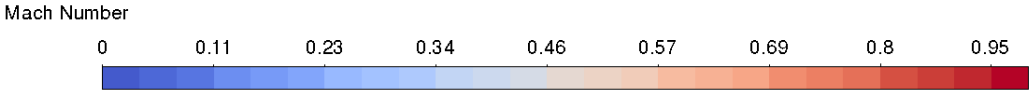
(a) Conventional design.



(b) Aggressive design.



(c) Very-aggressive design.

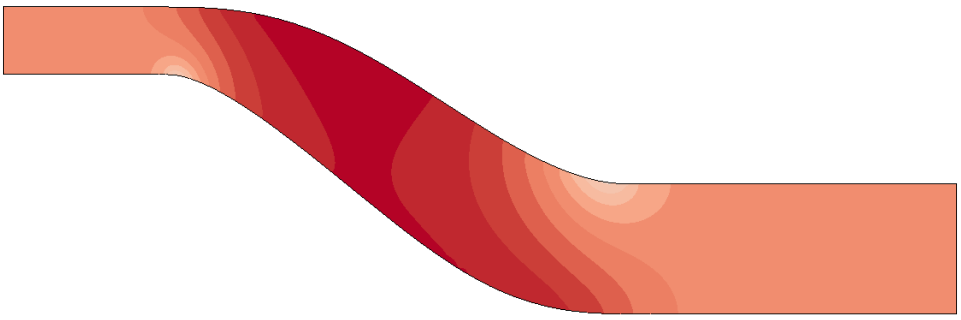


(d) Mach number color bar.

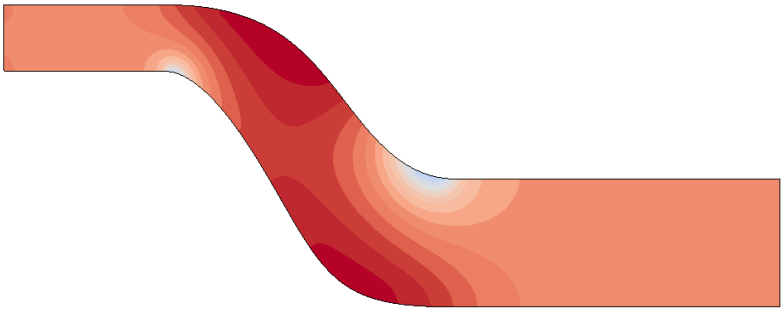
Figure 4.6: Mach number distribution for conventional, aggressive, and very-aggressive ICD design.

Figure 4.6 illustrates mach number distribution for conventional, aggressive, and very-aggressive ICD design. Convex turns downstream of the inlet and upstream of the outlet can be seen to induce acceleration of the wall adjacent flow and indicate separation form the hub and shroud. The relative reattachment axial position of the mean flow following the hub can be observed to occur after the concave turn for all designs. Progressively reducing the axial extent of the duct can be observed to induce a greater wall normal extent of wake downstream of the separation points as

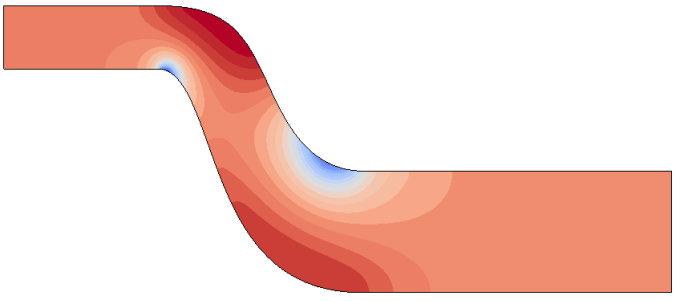
the required radial turning of the flow increases. A thicker wake downstream of the separation at the outlet induces a greater contraction of the mean flow, indicating that additional design considerations is needed at the inlet to the HPC for the aggressive designs. The very-aggressive design can further be observed to induce local mach numbers in the transonic regions, indicating that shocks may have occurred for other samples.



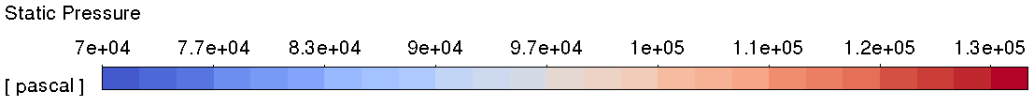
(a) Conventional design.



(b) Aggressive design.



(c) Very-aggressive design.



(d) Static pressure color bar.

Figure 4.7: Static pressure distribution for conventional, aggressive, and very-aggressive ICD design.

Figure 4.7 illustrates the static pressure distribution for conventional, aggressive, and very-aggressive ICD design. Progressively reducing the axial length can be seen

to induce greater low pressure zones at the convex turns, induced by the acceleration of the flow. Further the pressure recovery at the the hub downstream of the indicated separation point can be seen progressively delayed as the duct gets more aggressive. High pressure zones can further be observed at concave turns as the flow is retarded when facing the walls, and redirected by the duct.

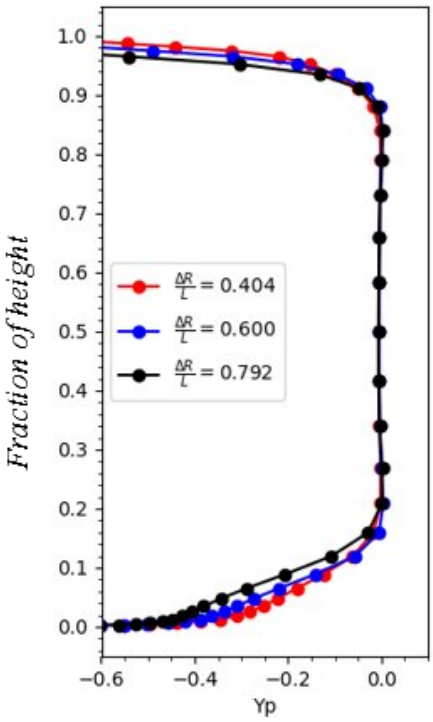


Figure 4.8: Y_p for conventional, aggressive and very aggressive ICD design.

Figure 4.8 illustrates Y_p measured at half the outlet length for three optimized duct designs in the range conventional, aggressive and very aggressive design range. The plot represent the pressure profile subjected to the fluid as a fraction of height where the hub line is located at 0.0, and the shroud line at 1.0. When the Y_p value reaches zero the pressure is equivalent to the stagnation free stream pressure at the inlet. It can be seen in the figure that decreasing the ICD length results in an increase in boundary layer thickness thus indicating greater losses as the ICD design gets more aggresive.

5
Conclusion

5.1 Optimization studies

From the studies, it has been shown that a reduction of the axial length from conventional to aggressive ICD design induces moderate pressure losses over the S-shaped duct with proportional effects regarding uniformity in flow profiles at the duct outlet. Further reducing the axial length of the ICD to the very-aggressive design range was shown to induce greater pressure losses compared to the jump from conventional to aggressive design, with an indicated further decrease in flow uniformity at the outlet. The source of the losses has been argued to be related to the separation occurring downstream of the convex radial turns of the S-shaped duct, with the pressure recovery in the wall adjacent flow becoming increasingly limited by the reduction of the axial length. Further peak Mach number at current flow conditions has been shown to approach 1, creeping closer with reduced axial length, indicating that a shock could occur, potentially further reducing the performance of the duct, a phenomena that was observed in some designs.

The effects of these pressure losses and flow uniformity regarding overall performance in a full engine integration are difficult to gauge in terms of the effect on downstream compressor and engine weight, but each design can be considered feasible for engine integration with the data presented in this thesis.

5.2 Future work

This specific study is limited to 2D axisymmetric simulations, that in this case is considered a good approximation, but a full 3D simulation could potentially introduce new design constraints due to the development of 3D flow structures in the flow field. Future work could therefore potentially include a full-scale 3D optimization of the duct, if manageable regarding computer resources, otherwise possibly reduced to 3D simulations of the generated designs in this thesis. Further, potentially the effects of guide vanes integrated at the ICD inlet could be investigated to see if it could better guide the flow filed over the first radial turn, potentially reducing the separation of the wall adjacent over the first radial turn, further reducing the separation-related losses.

Regarding the computational load of the current code generated for this thesis, it runs slow, requiring days to finish on a computer equipped with a 12 core CPU,

future work could therefore entail further development of the optimization code to run more efficiently in parallel. Additionally, the code could be improved regarding time by adding a surrogate model to the optimization algorithm, only running a selective amount of CFD simulations on estimates generated by the surrogate model.

Bibliography

[1] Reducing emissions from aviation, Climate Action, https://ec.europa.eu/clima/eu-action/transport-emissions/reducing-emissions-aviation_sv

[2] M.V.S Elías, "Aerodynamics of an Aeroengine Intermediate Compressor Duct: Effects from an Integrated Bleed System", *Chalmers University of Technology, Division of Fluid Dynamics, Department of Mechanics and Maritime Science, 2020.*

[3] H.I.H. Saravanamuttoo, G.F.C. Rogers, H.Cohen, P.V. Straznicky, A.C. Nix. *Gas Turbine Theory*, Seventh edition, Pearson Education Limited, Edinburgh Gate, United Kingdom, 2017.

[4] C.Ortiz Dueñas, R.J. Miller, H.P.Hodson, J.P.Longley. "EFFECT OF LENGTH ON COMPRESSOR INTER-STAGE DUCT PERFORMANCE", **GT2007-27752**.

[5] T. Stürzebecher, G. Goinis, C. Voss, H. Sahota, P. Groth, S. Hammer. "AUTOMATED AERODYNAMIC OPTIMIZATION OF AN AGGRESSIVE S-SHAPED INTERMEDIATE COMPRESSOR DUCT", **GT2018-75184**.

[6] L.Davidson. "Fluid Mechanics, turbulent flow and turbulence modeling", *Chalmers University of Technology, Division of Fluid Dynamics, Department of Mechanics and Maritime Science, 2021.*

[7] Frank M. White, Rhim Yoon Chul, "Fluid Mechanics", Eight Edition, McGraw-Hill Education 2016, New York. *Chapter 6.*

[8] I. B. Celik, Introductory Turbulence Modeling, West Virginia University Morgantown, Mechanical and Aerospace Engineering Dept, 1999.

[9] Towards Data Science, Bézier Curve, https://towardsdatascience.com/b%C3%A9zier-curve-bfffdadea212

[10] Gonzalo Montero Villar, "Arodynamic Optimization of Heigh Speed Propeller", Chalmers University of Technology, Depertment of Applied Mechanics, Gothenburg, Sweden, 2016.

[11] R. L. Haupt, S. E. Haupt, PRACTICAL GENETIC ALGORITHMS, seconds edition, Hoboken, N.J. Wiley, cop. 2004, [Online] Available: https://web.b.ebscohost.com/ehost/pdfviewer/pdfviewer?vid=0&sid=d360d4e3-748e-4879-893d-a5faf18bab16%40sessionmgr102

[12] R. M. Storn, "On the Usage of Differential Evolution for Function Optimization", Fuzzy Information Processing Society, 1996. NAFIPS. 1996 Biennial Conference of the North American, [Online] Available: https://www.researchgate.net/publication/3639093_On_the_usage_of_differential_evolution_for_function_optimization

[13] A.P Engelbrecht, *Computational Intelligence, An Introduction*, Second Edition, John Wiley & Sons, Ltd, 2007.

[14] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor, "OpenMDAO: An Open-Source Framework for Multidisciplinary Design, Analysis, and Optimization," Structural and Multidisciplinary Optimization, 2019. https://openmdao.org/

[15] Gustaver, M. (2020) A Chalmers University of Technology Master's thesis template for LaTeX. Unpublished.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS