



Delft University of Technology

Delft, The Netherlands

Faculty of Mechanical, Maritime and Materials Engineering

Process & Energy Department

Energy Technology Section

Master of Science Thesis

Optimization of the Preliminary Design of a Radial Compressor

Supervisor:
Prof. Ir. J.P. van Buijtenen

Mentor:
Mattia Olivero M.Sc.

By Jairsinho Pietersz
Report number 2428

Summary

Optimization is the mathematical process through which the set of conditions (satisfying the constraints) that produces the optimum of a specified function is obtained.

An optimization problem consists of design variables, objective function and constraints. Design variables are the quantities whose numerical values will be determined in the course of obtaining the optimal solution. Moreover the objective function is a mathematical equation that embodies the design variables to be minimized or maximized and constraints are conditions that must be satisfied to achieve the optimum.

Besides the design space and optimality of the problem must be clear to define the problem. The design space is the total region, or domain, defined by the design variables in the objective function.

When the above mentioned parameters are understood and applied, the problem can be formulated and a solution approach can be chosen to solve the optimization problem.

During the past, several optimization algorithms have been developed by several authors, but at first optimization algorithms can be globally distinguished as deterministic or stochastic algorithms.

Deterministic algorithms are algorithms which behave predictably. Given a particular input, it will always produce the same output, while stochastic algorithm is the opposite. Stochastic algorithms are the synonym of random search.

Moreover, deterministic algorithms can be divided in gradient based and derivative free algorithms, while stochastic algorithms can be divided in simulated annealing and evolutionary algorithms.

In this report several optimization methods have been studied and a one-dimensional model to determine the preliminary geometry of the radial compressor has been presented.

Moreover, the most suitable optimization algorithm for this assignment has been chosen. GA's has been chosen based on some criteria and advantages, which are shown in the report.

The optimization of the compressor design led to the choice of impeller hub radius $r_{lh} = 2.4 \text{ mm}$. The impeller tip radius is $r_{lt} = 4.0 \text{ mm}$. The relative flow angle is $\beta_1 = -59.76^\circ$, the relative tip Mach number is $M_{lt} = 0.86$ and the meridional velocity is $C_{m1} = 147.24 \text{ m/s}$.

The final configuration selected shows a blade number $Zr=9$ at a backsweep angle $\beta_{B2} = -45^\circ$. The correlated impeller exit radius r_2 and blade height b_2 are equal to 6.7 mm and 0.78 mm respectively, while the stage efficiency $\eta_{stage} = 0.5839$.

The absolute and relative flow angles at the impeller exit are $\beta_2 = -62.93^\circ$ and $\alpha_2 = 73.33^\circ$. The Mach number at the impeller exit is $M_2 = 0.7190$ and the pressure ratio is equal to 2, which is the chosen design value.

Acknowledgements

This thesis represents not only my work, but also a milestone in my life till now.

Prof. Ir. Jos P. van Buijtenen was my supervisor and M.Sc Mattia Olivero my mentor.

I would like to thank Prof. Ir. Jos P. van Buijtenen for giving me the opportunity of working on this final assignment and for his supervision.

I am very grateful to M.Sc Mattia Olivero for his helpful advices and the continuous assistance throughout the work. He consistently allowed this thesis to be my own work, but steered me in the right direction whenever he thought I needed it.

I thank my friends and all other people who helped me and encourage me.

Special thanks go to my girlfriend for the patience, encouragement and support during this work.

Finally, I would like to thank my family for encouragement and support.

Introduction

In the past years several optimization algorithms have been developed and tested for several purposes. Moreover these algorithms determine the way the optimization will be accomplished.

This report represents an effort for optimizing a radial compressor, which will be installed in a micro turbine.

The objective of this work was to study several optimization algorithms.

Furthermore a suitable optimization technique had to be chosen to perform calculations for the preliminary design of the radial compressor.

In chapter 1 a brief explanation has been given about several components necessary to perform optimization. When these components were clear, the second step was to study some optimization techniques.

In chapter 2 some deterministic methods (algorithms) has been explained. Deterministic methods are algorithms which behave predictably. Given a particular input, it will always produce the same output. They can be further divided in derivative-free and gradient based methods.

The following derivative free methods have been explained in this chapter:

- The Hooke and Jeeves method [1]
- The Hooke and Jeeves pattern search method modified by Onwubiko and Park [1] followed by Escape Algorithm for Hooke and Jeeves Pattern [1] search;
- Simplex Algorithm followed by Two-Phase Method and Dual Approach.

Moreover the following Gradient based algorithms have been explained in this chapter:

- The Lagrange's Method;
- Quadratic programming followed by Sequential Quadratic programming.

In chapter 3, stochastic methods have been explained. Stochastic methods are the synonym of random search. Besides these methods can be divided in simulated annealing and evolutionary algorithms.

In chapter 4, the station numbering and Euler turbomachinery equation have been presented followed by a one-dimensional model to determine the preliminary geometry of the radial compressor.

In chapter 5 the most suitable optimization technique to determine the preliminary geometry of the radial compressor have been chosen, based on some criteria and advantages. Subsequently, the components necessary for optimization have been applied to the 1 D model followed by results and effects of impeller outlet on performance.

Then in chapter 6 the final design has been presented after validation of several designs.

Finally, in chapter 7 the conclusions are shown and recommendations for future works are given.

Table of contents

Summary.....	i
Acknowledgements.....	ii
Introduction.....	iii
Table of contents.....	iv
List of figures.....	vi
List of tables.....	viii
List of symbols.....	x
Chapter 1 Optimization	
1.1. Introduction to optimization.....	1
1.2. Design Variables.....	2
1.3. Objective Function.....	3
1.4. Constraints.....	3
1.5. Design space.....	4
1.6. Optimality.....	5
1.7. Classification of Optimization Problems.....	6
1.8. Problem Formulation.....	7
1.9. Solution approaches in Optimization.....	8
Chapter 2 Deterministic Methods	
2.1 Derivative-free Methods.....	10
2.2 Gradient Based Methods.....	15
Chapter 3 Stochastic Methods	
3.1 Simulated Annealing.....	24
3.2 Evolutionary algorithms.....	27
Chapter 4 Preliminary geometry of the compressor	
4.1 Stations numbering.....	38
4.2 The Euler turbomachinery equation.....	39
4.3 Model for determining the geometry of the compressor.....	40
4.4 Impeller inlet design calculations.....	40
4.5 Impeller exit design calculations.....	43

Chapter 5 Problem statement and results

5.1 Chosen optimization Algorithm.....	48
5.2 Design Parameters.....	51
5.3 Design variables	52
5.4 Design objectives.....	53
5.5 Design Constraints.....	55
5.6 Design Space.....	56
5.7 Design Optimality	57
5.8 Inlet Design.....	57
5.9 Inlet and outlet optimization results and effects of impeller outlet on performance.....	63

Chapter 6 Final design

6.1 Optimized design for $N=600,000$ rpm.....	87
6.2 Optimized design for $N=600,000$ rpm for several exit blade angles and number of blades.....	89
6.3 Final overall design	94

Chapter 7 Conclusions and recommendations for future work

7.1 Conclusions.....	97
7.2 Recommendations.....	98

Appendix A. Assignment..... 99

Appendix B. Software Used.....101

Appendix C. Optimization Method Examples.....102

Appendix D. Specific speed and specific diameter.....126

Appendix E. Matlab Files..... 128

References..... 136

List of figures

Figure 1-1.	Definition of a design space.....	5
Figure 1-2.	Local and global maxima.....	6
Figure 1-3.	An ill-defined problem	8
Figure 2-1.	Local exploration for the Hooke and Jeeves Method	11
Figure 2-2.	Premature termination of pattern search and the escape mechanism.....	13
Figure 2-3.	A function of two variables with different function values.....	18
Figure 2-4.	The relationship between ∇f and ∇g at the point of tangency.....	18
Figure 3-1.	Principle work flow of an evolutionary algorithm.....	28
Figure 3-2.	The structure of an evolution program.....	31
Figure 3-3.	Roulette Wheel Parent Selection Procedure.....	33
Figure 4-1.	Meridional view of a radial compressor.....	38
Figure 4-2.	The Euler turbomachinery equation.....	39
Figure 4-3.	Sign convention for velocity triangles.....	40
Figure 4-4.	Impeller inlet velocity triangle.....	41
Figure 4-5.	Inlet velocity triangles at nonzero incidents.....	41
Figure 4-6.	Impeller exit velocity triangle.....	44
Figure 5-1.	Meridional view of a radial compressor.....	51
Figure 5-2.	Tip relative Mach number as function of meridional velocity for $N=600,000$ rpm and $B_I=0.02$	54
Figure 5-3.	Tip relative Mach number as function of meridional velocity for $N=600,000$ rpm and $B_I=0.03$	54
Figure 5-4.	Tip relative Mach number as function of meridional velocity for $N=600,000$ rpm and $B_I=0.04$	55
Figure 5-5.	Tip relative Mach number as function of meridional velocity for $N=600,000$ rpm and $r1h=1, 2, 3$ mm and $B_I=0.02$	61
Figure 5-6.	Tip relative Mach number as function of meridional velocity for $N=600,000$ rpm and $r1h=1, 2, 3$ mm and $B_I=0.03$	61
Figure 5-7.	Tip relative Mach number as function of meridional velocity for $N=600,000$ rpm and $r1h=1, 2, 3$ mm and $B_I=0.04$	62
Figure 5-8.	η_{stage} as function of DR for $N=600,000$ rpm and $B_I=0.02$	66
Figure 5-9.	η_{stage} as function of DR for $N=600,000$ rpm and $B_I=0.03$	68
Figure 5-10.	η_{stage} as function of DR for $N=600,000$ rpm and $B_I=0.04$	70
Figure 5-11.	Effect of β_{B2} on η_{stage} for $N=600,000$ rpm and $B_I=0.02$	72
Figure 5-12.	Effect of β_{B2} on M_{It} for $N=600,000$ rpm and $B_I=0.02$	73

Figure 5-13.	η_{stage} as function of M_{It} for $N=600,000$ rpm and $B_I=0.02$	73
Figure 5-14.	Effect of β_{B2} on η_{stage} for $N=600,000$ rpm and $B_I=0.03$	75
Figure 5-15.	Effect of β_{B2} on M_{It} for $N=600,000$ rpm and $B_I=0.03$	75
Figure 5-16.	η_{stage} as function of M_{It} for $N=600,000$ rpm and $B_I=0.03$	76
Figure 5-17.	Effect of β_{B2} on η_{stage} for $N=600,000$ rpm and $B_I=0.04$	78
Figure 5-18.	Effect of β_{B2} on M_{It} for $N=600,000$ rpm and $B_I=0.04$	78
Figure 5-19.	η_{stage} as function of M_{It} for $N=600,000$ rpm and $B_I=0.04$	79
Figure 5-20.	η_{stage} as function of Z_r for $N=600,00$ rpm and $B_I=0.02$	80
Figure 5-21.	M_{It} as function of Z_r for $N=600,000$ rpm and $B_I=0.02$	81
Figure 5-22.	η_{stage} as function of M_{It} for $N=600,000$ rpm and $B_I=0.02$	81
Figure 5-23.	η_{stage} as function of Z_r for $N=600,000$ rpm and $B_I=0.03$	83
Figure 5-24.	M_{It} as function of Z_r for $N=600,000$ rpm and $B_I=0.03$	83
Figure 5-25.	η_{stage} as function of M_{It} for $N=600,000$ rpm and $B_I=0.03$	83
Figure 5-26.	η_{stage} as function of Z_r for $N=600,000$ rpm and $B_I=0.04$	85
Figure 5-27.	M_{It} as function of Z_r for $N=600,000$ rpm and $B_I=0.04$	85
Figure 5-28.	η_{stage} as function of M_{It} for $N=600,000$ rpm and $B_I=0.04$	85
Figure 6-1.	Gas turbine generator layout	89
Figure 6-2.	Pump side elevations.....	90
Figure C.V-1.	Graphical solution to the example.....	110
Figure C.VIII-1.	function $f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$	113
Figure C.IX-1.	Function plot.....	125

List of tables

Table III-1.	Relationship between Physical and Simulated Annealing.....	25
Table III-2.	Example of mutation process.....	33
Table V-1.	Operating condition.....	52
Table V-2.	Design variables.....	53
Table V-3.	Hub radii chosen for the design.....	53
Table V-4.	Design Constraints.....	56
Table V-5.	Exit blade angle reference values.....	57
Table V-6.	Blade numbers reference values.....	57
Table V-7.	Inducer design results at $N=600,000$ rpm for $r_{lh}=1, 2, 3, 4, 5, 6$ mm and $B_I=0.02$	58
Table V-8.	Inducer design results at $N=600,000$ rpm for $r_{lh}=9, 12, 15, 18, 21, 24$ mm and $B_I=0.02$	58
Table V-9.	Inducer design results at $N=600,000$ rpm for $r_{lh}=1, 2, 3, 4, 5, 6$ mm and $B_I=0.03$	59
Table V-10.	Inducer design results at $N=600,000$ rpm for $r_{lh}=9, 12, 15, 18, 21, 24$ mm and $B_I=0.03$	59
Table V-11.	Inducer design results at $N=600,000$ rpm for $r_{lh}=1, 2, 3, 4, 5, 6$ mm and $B_I=0.04$	59
Table V-12.	Inducer design results at $N=600,000$ rpm for $r_{lh}=9, 12, 15, 18, 21, 24$ mm and $B_I=0.04$	60
Table V-13.	Inducer design results at $N=600,000$ rpm for $r_{lh}=1, 2, 3$ mm and $B_I=0.02$	62
Table V-14.	Inducer design results at $N=600,000$ rpm for $r_{lh}=1, 2, 3$ mm and $B_I=0.03$	63
Table V-15.	Inducer design results at $N=600,000$ rpm for $r_{lh}=1, 2, 3$ mm and $B_I=0.04$	63
Table V-16.	Optimization results with diffusion ratio as equality constraint for $N=600,000$ rpm and $B_I=0.02$	64
Table V-17.	Optimization results after converting Zr into integer for $N=600,000$ rpm and $B_I=0.02$	65
Table V-18.	Optimization results with diffusion ratio as equality constraint for $N=600,000$ rpm and $B_I=0.03$	66
Table V-19.	Optimization results after converting Zr into integer for $N=600,000$ rpm and $B_I=0.03$	67
Table V-20.	Optimization results with diffusion ratio as equality constraint for $N=600,000$ rpm and $B_I=0.04$	68
Table V-21.	Optimization results after converting Zr into integer for $N=600,000$ rpm and $B_I=0.04$	69

Table V-22.	Optimization results: effects of β_{B_2} for $N=600,000 \text{ rpm}$ and $B_I=0.02$	71
Table V-23.	Optimization results: effects of β_{B_2} for $N=600,000 \text{ rpm}$ and $B_I=0.03$	74
Table V-24.	Optimization results: effects of β_{B_2} for $N=600,000 \text{ rpm}$ and $B_I=0.04$	77
Table V-25.	Optimization results: effects of Zr for $N=600,000 \text{ rpm}$ and $B_I=0.02$	80
Table V-26.	Optimization results: effects of Zr for $N=600,000 \text{ rpm}$ and $B_I=0.03$	82
Table V-27.	Optimization results: effects of Zr for $N=600,000 \text{ rpm}$ and $B_I=0.04$	84
Table VI-1.	Optimization results for $B_I=0.02, 0.03$ and 0.04 without considering the influence β_{B_2} and Zr	88
Table VI-2.	Summarized optimization results for $N=600,000 \text{ rpm}$ and $B_I=0.02$	91
Table VI-3.	Summarized optimization results for $N=600,000 \text{ rpm}$ and $B_I=0.03$	92
Table VI-4.	Summarized optimization results for $N=600,000 \text{ rpm}$ and $B_I=0.04$	93
Table VI-5.	Summarized optimization results for $B_I = 0.02, 0.03$ and 0.04	94
Table VI-6.	Final optimization results for $N=600,000 \text{ rpm}$	95
Table C.III-1.	Initial tableau (phase 1).....	107
Table C.III-2.	Iteration 1 (phase 1).....	107
Table C.III-3.	Iteration 2 (phase 1).....	107
Table C.III-4.	Iteration 1 (phase 2).....	108
Table C.III-5.	Final Iteration (phase 2).....	108
Table C.VII-1.	Iteration results.....	112
Table C.VIII-1.	Bit position and random number.....	120
Table C.VIII-2.	Bit position, chromosome number and bit number within chromosome.....	121

List of symbols

Symbol	Unit	Meaning
A	m ²	Area
a	m/s	Speed of sound
B	-	Boundary layer blockage factor
b	m	Blade height
C	m/s	Absolute velocity
C _m	m/s	Meridional component of absolute velocity
C _{slip}	m/s	Slip velocity
C _u =C _θ	m/s	Tangential component of absolute velocity
c _p	J/(kg·K)	Specific heat at constant pressure
c _v	J/(kg·K)	Specific heat at constant volume
$\hat{c}_v$	-	Constant equal to 3/2 for monatomic gases and to 5/2 for diatomic gases
D	m	Diameter
DR	-	Diffusion ratio of velocity
E	J	Energy
F	N	Force
g	m/s ²	Gravitational acceleration
H _{ad}	ft	Adiabatic head
h	J/kg	Specific enthalpy
Δh _{0, is}	J/kg	Specific enthalpy rise
IGV		Inlet Guide Vane
i _l	°	Incidence angle at impeller inlet
k	J/K	Boltzmann constant
L	m	Impeller axial length
M	-	Mach number
MTT		Micro Turbine Technology
m	kg	Mass
$\dot{m}$	kg/s	Mass flow rate
N	rpm; -	Rotational speed; number of
N _s	rpm·ft ^{3/4} /s ^{1/2}	Specific speed
n _s	-	Dimensionless specific speed
p	Pa; %	Pressure; probability
p _r	-	Pressure ratio
Q ₀₀	ft ³ /s	Volume flow rate
R	J/(kg·K)	Gas constant
r	m	Radius
T	K	Temperature
U	m/s; J	Blade speed; internal energy
V	m ³ ; m/s	Volume; relative velocity at the leading edge of the MTT microturbine blade

$\dot{V}$	m^3/s	Volume flow rate (SI)
W	m/s	Relative velocity
$W_u=W_\theta$	m/s	Tangential component of relative velocity
W_x	J/kg	Amount of work per unit mass flow
Z_r	-	Number of blades

Greek

α	$^\circ$	Absolute flow angle
β	$^\circ$	Relative flow angle
β_B	$^\circ$	Blade angle
γ	-	Ratio of specific heats
∂	-	Step size
Δ	-	Difference
η	-	Efficiency
λ	-; -	Swirl parameter; Lagrange multiplier
μ	-	Work coefficient
ρ	kg/m^3	Density
σ	-; Pa	Slip factor; stress
ω	$1/\text{s}$	Rotational speed
Φ	-	Flow coefficient

Subscripts

0	Stagnation or total state; ambient condition
1	Impeller inlet
2	Impeller exit
3	Diffuser Inlet
4	Diffuser exit
ad	Adiabatic
B	Blade property
c	cycles; crossover
∞	Refers to absence of slip
f	Flow
h	Hub
is	Isentropic
m	Meridional velocity component; mutation
r	Radial velocity component
rel	Relative to rotating coordinates
rms	Root mean square
rotor	Refers to compressor rotor
S	Static
s	selection
slip	Refers to slip condition
stage	Refers to compressor stage
T	Time
t	Tip
ts	Total-to-static
$u = \theta$	Tangential velocity component

Chapter 1. Optimization

In this chapter a brief explanation about optimization will be given. The chapter starts with an introduction to optimization, which gives some design optimization examples and the definition of optimization.

Moreover an optimization problem consists of design variables, objective function and constraints.

Furthermore, the design space and optimality of the problem must be clear to define the problem.

The following section, classification of optimization problems gives some sections in which optimization problems can be classified.

Finally, after previous sections is understood and applied, the problem can be formulated and a solution approach chosen to solve the optimization problem.

1.1. Introduction to optimization

Before the definition of optimization, an essential concept to understand optimization is considered.

Although design methodology is often poorly understood, there are generally accepted design steps given in introductory design texts. If designers follow these steps, solutions can generally be found.

However, two questions still need to be answered before the design can be considered complete:

- Is it adequate?
- Can it be improved?

Adequate design can be defined as the selection of the sizes of materials that satisfy the functional requirements of the design while keeping the costs and undesirable effects within tolerable limits.

Adequate design is usually based on the engineering information available in equations and graphs (in hand books) and on the experience of the designer. Often, because of cost considerations, adequate sizes are discarded in favour of a standard size. Although this approach may be satisfactory, other conditions may preclude its use. For example, in the design of a machine element such as a shaft, the size specified may really be the best solution, given the constraints of the original design. In other cases, it is necessary to modify the design, regardless of how good it is, in order to achieve other objectives.

The fact that designer frequently modify existing designs is in effect an admission that adequate designs are not always the best possible designs, given all the constraints imposed on them.

To fully consider how best to improve a design, the concept of optimum design must be understood, which is the best design among the alternatives that meet a specified objective. In many books optimization is defined as follows:

- Optimization is the mathematical process through which the set of conditions (satisfying the constraints) is obtained that produces the maximum or minimum value of a specified function.

Ideally, one would like to obtain the perfect solution the design situation. But in the real world the only strive is to achieve the best solution possible within the present constraints of time and funds available.

1.2. Design Variables

An engineering optimization model consists of parameters and decision variables. Design parameters are data that define the problem.

Design variables are the quantities whose numerical values will be determined in the course of obtaining the optimal solution. These decision variables are called design variables. They include such things as size or weight, geometry or the number of teeth in a gear, the number of coils in a spring, the number of blades in a hub, or the number of tubes in a heat exchanger. In short, they represent the variables that are required to quantify or completely describe an engineering system. The number of variables depends on the type of design involved, and as this number increases, so does the complexity of the solution.

Design variables may be one of three types. A design variable is continuous if it is free to assume any value. However, when a design variable can only assume a fixed value, it is discrete [1]. This would be the case when, for example, rod diameter can only be selected from a set of finite standard sizes. In some situations, variables can only assume integer values. These design variables are known as integer variables. Typical examples include the number of teeth in a gear, the number of threads, the number of blades in a hub, or the number of rivets required to assemble a structure.

It is important in design optimization to clearly identify the design variables and the design parameters because confusing them will result in an improper optimization model. Consider the simple case where the appropriate short rod for carrying a specified load P is to be selected using the simple model $\sigma = P/A$. For this case, stress (σ) and load are the design parameters and area A is the design variable.

It would be incorrect to include stress as a design variable here because it is dependent on the area. However, should the design call for material selection, stress would then become a design variable and the other two would become design parameters. The point is that design variables are determined by the nature of the problem. Thus, a design variable in one situation may very well be a design parameter in another. So, always identify the design variables and design parameters carefully. It is also a good practice to select design variables such that they are as independent of each other as possible.

1.3. Objective Function

The process of selecting the "best solution" from various possible solutions must be based on a prescribed criterion, which is the objective function. For the purpose of optimization, the objective function is defined as a mathematical equation that embodies the design variables to be minimized or maximized [1]. It can be given in the form:

$$D = f(x), \quad (1.1)$$

where D stands for a desired effect that is to be achieved. The vector $x = (x_1, x_2, \dots, x_n)$ represents the design variables.

The objective function may be the equation that maximizes the life and efficiency of a mechanical element such as a gear set, or it may be the equation that reduces an undesirable condition, such as the weight of a machine element. The choice of objective function is crucial, because different objective functions produce different optima.

Therefore, choosing which objective function to optimize is one of the most important steps in the process.

In cases where a company's policy is clear, it may be an easy and obvious decision. However, in instances where no policy exists, it may not be obvious at all. Suppose it is decided to set up a CAD/CAM laboratory from scratch.

At least two objectives—minimizing costs and maximizing the number of work stations—have to be pursued. These two objectives are, at first glance, in opposition to each other; every one leads to a different solution. Which one must be chosen? Selecting the objective function requires a great deal of care, insight, and experience.

It is possible to have only one objective. This situation is referred to as single criterion optimization. Situations where two or more objectives are desired are referred to as multi criteria or multi goal optimization. Sometimes objectives conflict with each other. Suppose a car manufacturer sets up four objectives: minimize cost, minimize weight, maximize structural integrity, and maximize vehicle size. Obviously, it would be very difficult to maximize the size and minimize the weight at the same time. Therefore, in the case of multigoals, the designer must establish priorities and assign weighted values to each set of design criteria.

1.4. Constraints

In this section constraints will be explained and some examples concerning constraints will be mentioned.

Designers are rarely free to design as they please. In thermal system design, which requires high—temperature conditions, designers are limited because few materials withstand high temperatures, and, for very high temperatures, suitable materials do not yet exist. A restriction which almost every time is taken into account is, of course, cost. Such restrictions are called restriction constraints [1].

Optimization constraints are therefore numerical values of identified conditions that must be satisfied to achieve a feasible solution to a given problem.

Generally, there are two types of constraints, internal and external.

External constraints are those uncontrollable restrictions or specifications imposed on a system by an outside agency and therefore not under the direct control of the designer. Typical examples of external constraints include laws and regulations set by governmental agencies, such as allowable materials for house construction. Availability of raw materials can also be an external constraint.

Internal constraints are those imposed by the designer. They require a keen understanding of the physical system or an engineering background. Such constraints arise from the fundamental laws of conservation of mass, momentum, and energy. Other constraints arise from stress and geometrical considerations. For example, the design of a helical spring may require a restrictive expression relating the volume, the number of coils N , the mean spring diameter D and the wire diameter d as:

$$V = \frac{\pi^2 d^2 D N}{4}.$$

Additional constraints may be placed on the spring index (ratio D/d) so that it is within a prescribed range.

Mathematically, constraints can be written either in the form of an equality such as $h(x) = 0$ or in the form of an inequality $g(x) \geq 0$.

One major problem in optimization is how to handle constraints. It is often of practical value to analyze how a constraint affects the outcome of an optimization problem because this activity may help in problem simplification. Suppose a constraint is removed from the problem; how does this affect the optimization. If this approach does not affect the optimum value, the constraint is inactive; this implies that at a point x^* , $g(x^*) > 0$. However, if the optimum is affected, then the constraint is active (binding). Alternatively, an inequality constraint at some x^* is now satisfied strictly in form of an equality of the form $g(x^*) = 0$.

1.5. Design space

The total region, or domain, defined by the design variables in the objective function is called the design space. This is normally limited by constraints (Figure 1-1); otherwise, there is an unbounded design space for which no acceptable solution exists.

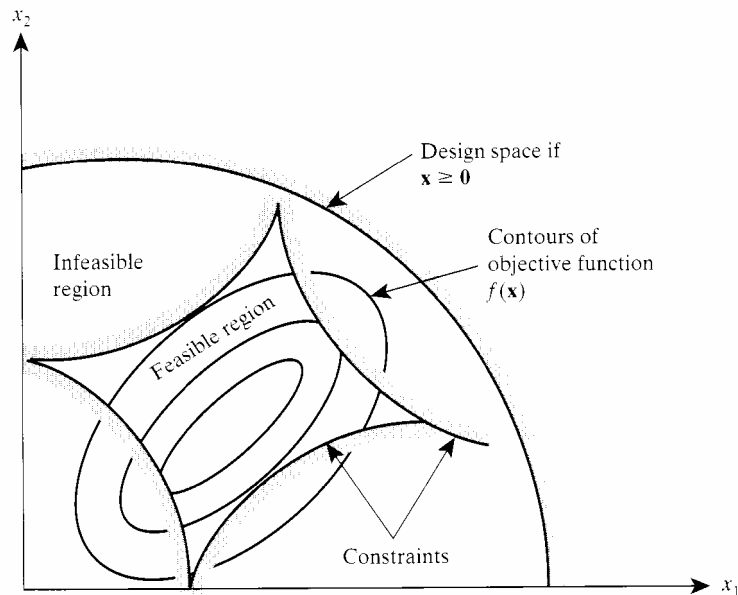


Figure 1-1. Definition of a design space [1]

Therefore, the use of constraints is especially important in restricting the region in which the search for optimum values of the design variables will be done.

The set of all points that satisfy both the equality and the inequality constraints is known as the feasible region of the objective function $f(x)$. The vector x in the feasible region is called a feasible point. A feasible point can be either in the interior of the feasible region or at its boundary. Optimization involves selecting the feasible point that results in the best improvement of the objective function. Of course, if the problem is constrained, the optimum point must not only result in the best improvement of the objective function but must also satisfy the stated constraints.

1.6. Optimality

In most optimization problems, the objective is to find a minimum [1]. A maximization problem can be converted to minimization by changing the sign of the objective function such that, maximize $f(x) = -\text{minimize } (-f(x))$. The maximizing or minimizing of a function in a design situation requires the definition of certain terms.

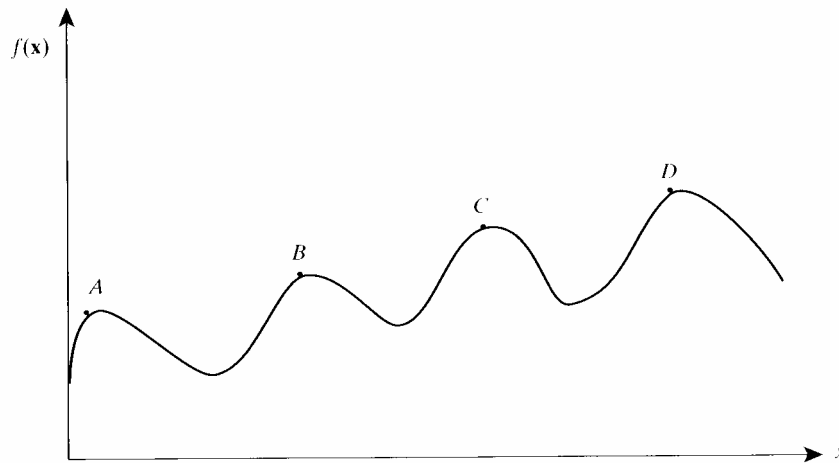


Figure 1-2. Local and global maxima [1]

Throughout the definition of the optimality, the term optimum to refer to the solution that best meets the design objective. That is, the optimum can be associated with either a minimum or a maximum solution.

Consider the function shown in Figure 1-2; it is obvious that:

$$f(D) > f(C) > f(B) > f(A).$$

Point A is the highest point in its immediate vicinity, yet it is the least of the four points. This illustrates the concept of local and global maxima. The point in the design space that is higher than all other points within its immediate vicinity is therefore the local maximum, whereas the highest of all local maxima is called the global maximum. Conversely, the global minimum is the smallest of all the local minima.

1.7. Classification of Optimization Problems

Before discussing some techniques used to find numerical solutions to optimization problems, various types of problems that will be encountered in optimization must be known.

There are at least six categories [1] of optimization problems:

- Category 1 (Constraint) If the problem is stated with some constraints, this is a constrained optimization. If, however, the problem is stated without constraints, then this is an unconstrained optimization;
- Category 2 (Variable) If the objective function is a function of one variable, this is a single-variable (or univariate) optimization. On the other hand, if the objective function consists of two or more variables, the problem is known as multivariable (or multivariate) optimization. If the design variables can only assume discrete values, this is a discrete optimization. Similarly, if the variables must assume integer values, this is integer optimization;

- Category 3 (Objective) An optimization problem with a single design objective is known as single-criterion optimization, but if there are two or more objectives, the problem is known as multicriteria optimization.
- Category 4 (Linearity) If the objective function and its constraints consist of linear functions, the problem is known as a linear programming (LP) problem.
However, if either the objective function or the constraints consist of nonlinear functions, the problem is known as a nonlinear programming (NP) problem.
- Category 5 (Time) If the optimization problem is time-dependent, it is known as dynamic optimization. If, the problem is not time-dependent, this is static optimization.
- Category 6 (Data) This category depends on the nature of the data available. If the data are known with certainty, the problem is said to be deterministic. If the data are not known with certainty, the problem is said to be stochastic optimization.

1.8. Problem Formulation

Two difficulties confront both the novice and the experienced designer in the field of optimization. The first is how to solve an optimization problem and often the most discouraging to the novice, is how to formulate optimization problems.

The first step in design optimization is to identify the objective. As in every design situation, it is often difficult to decide what should be the controlling objective, and good designers do not attempt optimization processes until their objective is defined. The next step is to identify the design variables that can be mathematically related to the objective. If the design objective cannot be modelled in terms of the design variables, the methods presented in this report are not appropriate, and an experimentation procedure may be necessary.

After obtaining a mathematical model, which describes the system or process, the next step is to decide what constraints, if any, should be used. One classic source of frustration during the execution of an optimization process is an ill-defined design space.

Therefore, great care must be exercised in defining the constraints. Even when it's almost certain that the objective function and constraints have been properly defined, design-space problems may still exist. This is particularly the case when the objective function is a rational polynomial function (a function whose numerator and denominator are polynomials). When such an objective function exists, it may be necessary to restrict the denominator so that the objective function does not become unbounded. An illustration of an ill defined problem is given in Figure 1-3.

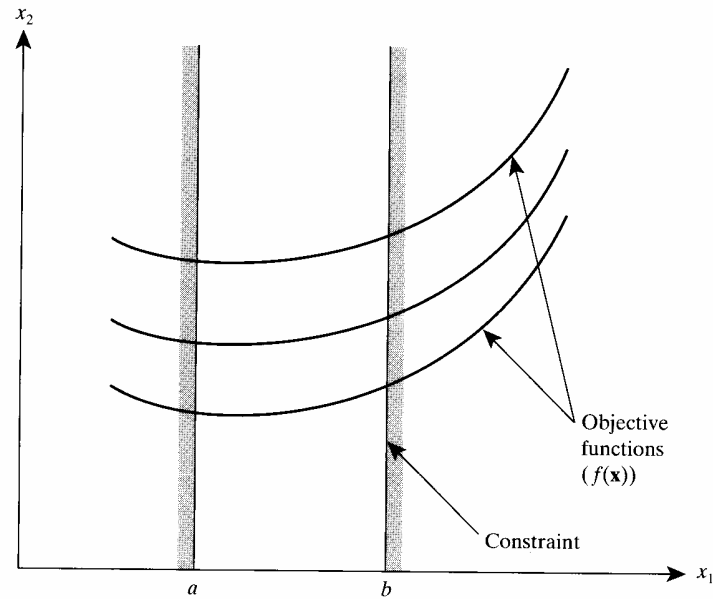


Figure 1-3. An ill-defined problem (The objective function is bounded on x_1 but not on x_2) [1]

1.9. Solution approaches in Optimization

The goal of a good optimization model is to obtain useful numerical values.

Once the problem has been formulated, there are many ways to obtain a solution.

These methods can be summarized as follows:

- Graphical Method;
- Analytical Technique ;
- Numerical Techniques;
- Experimental Technique.

Each of the above mentioned methods will be briefly explained in the following sub-paragraphs.

1.9.1. Graphical Method

In this method the objective function is plotted in terms of the decision variables. This method is limited to two-dimensional problems that are problems with no more than two design variables. Once considered tedious and time-consuming, software packages have made this a much more attractive approach.

1.9.2. Analytical Technique

The cornerstone of this technique is differential calculus. The objective function is differentiated once and set to zero. The values of the variables at that instant are considered optimum. A second derivative test is then applied to determine whether the optimum is a maximum or a minimum. If the problem is a constrained optimization, the classical method of Lagrange multipliers can be used. The drawback in

this technique, as well as with many of the other techniques, is that the objective functions must be described

as mathematical models. Also, this technique cannot be used if the objective function is not differentiable or if the derivatives are discontinuous.

1.9.3. Numerical Techniques

Most numerical techniques are iterative and come under the classification of search methods. One approach uses gradients. The other approach does not require knowledge of gradients. However, both approaches make use of past information during the iterative process.

1.9.4. Experimental Technique

This technique does not require a mathematical model of the physical system because the actual process is used. An experiment is performed on the process and the result is compared to that of the preceding experiment, in order to decide where to locate the next one. This procedure is continued until the optimum is achieved.

Chapter 2. Deterministic Methods

Deterministic methods are defined as methods whose resulting behaviour is entirely determined by its initial state and inputs, and which is not random or stochastic.

These methods can be divided in derivative free and gradient based methods.

This chapter starts with derivative free methods and ends with gradient based methods.

The first section of this chapter starts with the Hooke and Jeeves pattern search method [1] continued by Hooke and Jeeves pattern search method modified by Onwubiko and Park [1]. Furthermore an explanation about the Escape Algorithm for Hooke and Jeeves Pattern search [1] will be given.

Then the simplex Method will be explained continued by the two-phase method.

Finally, the last method that will be explained in this chapter is the dual approach, which applies the simplex method after transformation of the objective function.

2.1. Derivative-free Methods

Derivative-free methods are convenient because they do not require the determination of the derivatives of the objective function or the constraints and are thus generally applicable because they can be used for functions whose differentiation is difficult and even for functions whose derivatives are discontinuous.

Anyway, there are some disadvantages concerning these methods. They are not rooted in any mathematical basis, because most of them are based on function comparisons. Consequently, since no optimality conditions can be applied to them, solutions obtained by these methods cannot be considered optimum. No convergence criteria can be established and hence the search may be terminated prematurely.

Anyway, when other techniques fail the only action that remains is to use these techniques.

2.1.1. Hooke and Jeeves Method

One of the most widely used direct-search techniques is the method developed by Hooke and Jeeves [1]. This method, known as the pattern search, is based on a sequence of exploratory and pattern moves, starting at an initial base point. A step size for each variable is selected. A local exploration begins with the evaluation of the objective function at an initial base point, $x_0 = \{x_1, x_2, \dots, x_n\}$ and two other points removed from it by a predefined step size. If one of the points results in a decrease in the objective function (for the case of minimization), a success (S) is said to result and the particular point that produced the success is called a temporary base point, x'_0 . If neither of the two points produces a success, the step size for that variable is reduced by half and the exploration is repeated. Figure 2-1 illustrates the local exploration for the two-dimensional case.

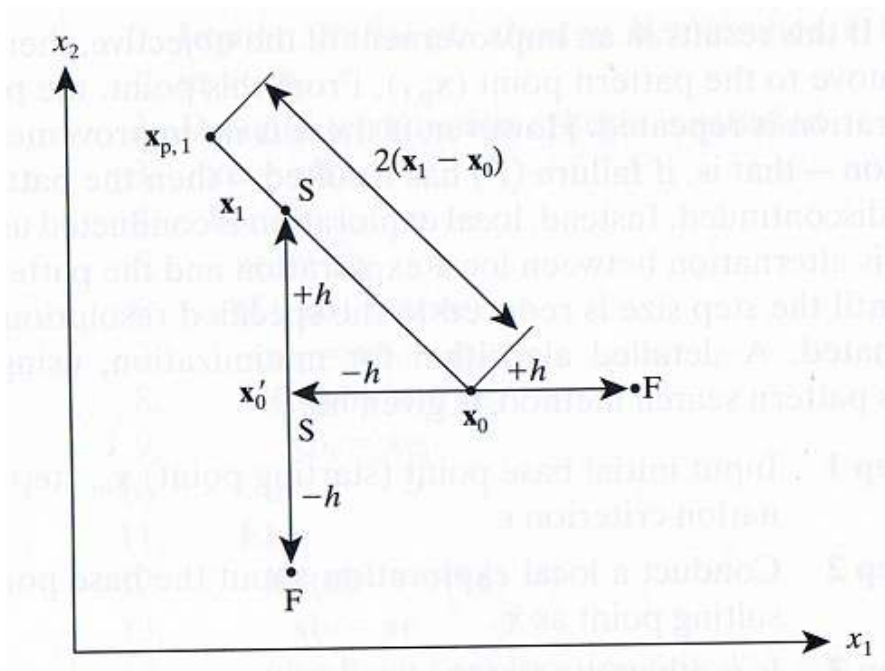


Figure 2-1. Local exploration for the Hooke and Jeeves Method [1]

The search begins at the base point x_0 with a step size of $\pm h$. An exploration is carried out in the x_1 - dimension. Suppose that the point x_1-h produces a success (S); this point becomes the temporary base. An exploration search is then performed in x_2 using x_0' as the base point. If a success results, then a new temporary base point is established and identified as x_1 . With the establishment of this new base point, the exploration is stopped and a pattern move is carried out.

The pattern move accelerates the search process by using a greater step size. The original and the most recently established base points create a pattern that is used to locate the first pattern point designated $x_{p,1}$. The direction of the pattern move is represented by a line passing through the original point (x_0) and the newest base point (x_1). The pattern base point ($x_{p,1}$) is obtained by doubling the distance from x_0 to x_1 , which is the same as subtracting the original base point from twice the newest base point. This is expressed as follows:

$$x_{p,1} = x_0 + 2(x_1 - x_0);$$

or

$$x_{p,1} = 2x_1 - x_0.$$

To move to this new pattern point, it must be assured that such a move will result in a success. The objective function is evaluated at the pattern point ($x_{p,1}$). If this results in an improvement of the objective, then the pattern is made to move to the pattern point ($x_{p,1}$). From this point, the process of the local exploration is repeated. However, if there is no improvement in the objective function and if failure (F) has resulted, then the pattern move to point $x_{p,1}$ is discontinued. Instead, local exploration is conducted using the base point x_1 . This alternation between local exploration and the pattern move is continued until the step size is reduced to the specified resolution and the search is terminated. A detailed algorithm for minimization, using the Hooke and Jeeves pattern search method is given here:

Step 1 Input initial base point (starting point) x_b , step size h , and termination criterion ϵ .

Step 2 Conduct a local exploration about the base point; denote the resulting point as x_t .

Step 3 Is x_t a better point than x_b ?

- (a) Yes: Go to step 5.
- (b) No: Go to step 4.

Step 4 Is termination criterion satisfied?

- (a) Yes: Stop; x_t approximates the optimal solution.
- (b) No: Reduce step size by half and return to step 2.

Step 5 Accelerate search by making a pattern move to the point x_p defined by:

$$x_p = 2x_t - x_b.$$

Step 6 Conduct a local exploration using the point x_p and designate the resultant new point of the exploration as x'_p .

Step 7 Is x'_p a better point than x_t ?

- (a) Yes: Set $x_b = x_t$ and $x_t = x'_p$. Go to step 5.
- (b) No: Set $x_b = x_t$. Go to step 4.

An example of the Hooke and Jeeves method will be given in Appendix C.1.

2.1.2. The Hooke and Jeeves pattern search method modified by Onwubiko and Park

The Hooke and Jeeves pattern search method [1] is based on the principle that during local exploratory moves the constraints are evaluated and, if any of them are violated, the step size is reduced. If there is no constraint violation, then a pattern move starts.

Constraints are first checked to see whether the pattern move will result in constraint violation. If so, the step size is reduced until there is no constraint violation and the pattern move is then carried out. This procedure sometimes prematurely terminates the search.

Consider Figure 2-2 and suppose that the following search direction, d_k brings the search to the point x_k , which is obviously not the minimum. Any local exploration in either direction of the two variables will result in constraint violation because it involves taking steps on either side of the base point along the direction of the coordinate. If the step size is continuously reduced, the termination criterion is reached without reaching the minimum.

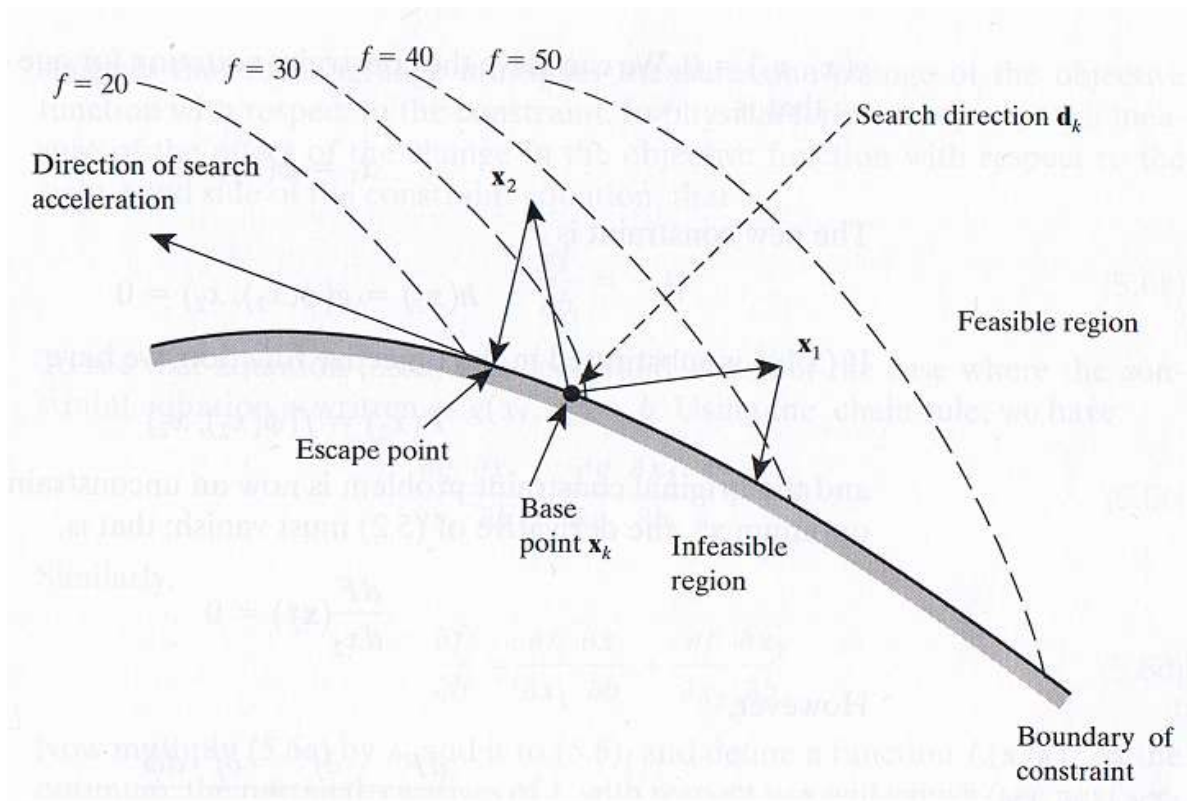


Figure 2-2. Premature termination of pattern search and the escape mechanism by [1]

2.1.3. Escape Algorithm for Hooke and Jeeves Pattern search

To avoid prematurely terminated search the following escape algorithm has been proposed [1].

- | | |
|---------------|---|
| Step 1 | Set $\partial = 0.001$ which is the step size and $i=1$ |
| Step 2 | Initiate an escape by perturbing on each variable x_i using ∂ . Evaluate the constraints. |
| Step 3 | If the constraints are violated, go to step 7; otherwise go to step 4. |
| Step 4 | Conduct a local exploration using a step size of $\frac{\partial}{100}$. |
| Step 5 | If the new point is better than the base point, go to step 6. If $i = n$ (number of variables) terminate search. The base point is the optimum. Otherwise, set $i = i + 1$ and return to step 2. |
| Step 6 | Accelerate search in the direction going from the base point through the escape point found in step 5. |
| Step 7 | Set $\partial = \frac{\partial}{2}$. If $\partial \leq \epsilon$ a specified tolerance then terminate the search because the base point is the optimum otherwise return to step 2 for more perturbation in the search direction. |

The escape algorithm works well as a subroutine in the pattern search. However it is best used at the end of the search to prevent premature search termination.

2.1.4. Simplex Algorithm

The simplex method [1] developed by George Dantzig in 1947, is an efficient and systematic method for obtaining optimums if they exist for linear programming problems. In mathematics, linear programming (LP) problems involve the optimization of a linear objective function, subject to linear equality and inequality constraints.

Anyway the principle behind the simplex method is elegant. The objective function is moved from one feasible basic solution to another. Only the feasible basic solutions increase (maximize) or decrease (minimize) the objective function considered. The simplex algorithm which is sometimes known as the primal simplex procedure is used to maximize the objective function of a linear programming problem. The steps which compose the algorithm are as follows:

- | | |
|---------------|---|
| Step 1 | Convert all inequalities to equalities using slack and artificial variables. |
| Step 2 | Set up an initial simplex tableau. The simplex tableau presents the system of equations in tabular form. |
| Step 3 | Identify the most negative entry that is the one with the largest magnitude in the objective row. The column containing this element is known as the pivot column denoted by the j^{th} column. If ties exist, choose anyone of the tied columns arbitrarily. |
| Step 4 | Divide each positive solution value by the corresponding positive element of the pivot column in the same row as the positive solution value. Store the value of each ratio and select the smallest one. The row, say the i^{th} that corresponds to the smallest ratio is the pivot row. The a_{ij} entry is the pivot element. Note that if there is a tie one of the tied rows is chosen arbitrarily, |
| Step 5 | Create a second simple tableau from the first using row operations. To begin make the value of the pivot element unity. Using the pivot element and row operations, all elements above and below the pivot element that are in the same pivot column must be cleared. |
| Step 6 | Check the objective function row, see whether any of its entries are still negative. If so return to step 3 and repeat the process. Otherwise the optimum has been achieved. |

2.1.5. The Two-Phase Method

For constraints with inequality sign of the form " $\leq$ " it was easy to obtain a subidentity matrix (i.e., initial feasible basis) by the addition of slack variables. However if, the constraints are either

inequality constraints of the form $\geq$ or equality constraints, it becomes difficult to find an initial feasible solution or a subidentity matrix. To solve this difficulty the two-phase method is used.

The method consists of two phases. The objective of phase 1 is to find a feasible basic solution to use in phase 2. This is done by driving the artificial variables to zero. Once the feasible basic solution has been found, phase 2 simply consists of the application of simplex method.

The first step is to begin by replacing the objective function of the original problem with the sum of the artificial variables while continuing to use the original constraints. The simplex algorithm is applied until an optimum is obtained. If the objective function has a value greater than zero the problem has no solution because the artificial variables are not part of the original problem. If the phase 1 has a solution, proceed to phase 2.

2.1.6. The Dual Approach

The simplex algorithm outlined in section 2.1.3 is for maximization problem. Naturally there arises the question of how to deal with minimization problems.

An important property of linear programming (LP) is that every LP problem is associated with another LP problem. The initial problem is called primal and the associated problem is called dual. The implementation of duality is that a minimization problem can be converted into a maximization problem.

If the primal problem is to minimize the following equation:

$$CX;$$

which is subjected to the following constraints:

$$AX \geq B;$$

$$X \geq 0.$$

When the dual approach is used the previous equation is transformed in the maximization of the following equation:

$$B^T Z;$$

which is subjected to the following constraints:

$$A^T Z \leq C^T;$$

$$Z \geq 0.$$

2.2. Gradient Based Methods

Gradient methods differ from other search methods in the amount of computational effort involved. In other search methods, only the evaluation of the objective function is required, whereas gradient methods may also require the evaluation of the derivatives of the objective function. This drawback notwithstanding, the overall computational efficiency of certain gradient methods is higher than that of the search methods, because the gradient methods are able to converge in fewer steps.

Gradient methods are classified as either first-order or second-order methods, depending on whether the first partial derivatives or the second partial derivatives of the function are required.

Quadratic Programming, Sequential Quadratic Programming, Steepest descent and Conjugate gradient method are some first-order methods and Newton's Method is one of second-order methods.

Lagrange's Method is a first-order method which can be modified to a second-order method.

In section 2.2 the Lagrange's Method concerning first-order method will be described and an example is given in the appendix.

Further on, Quadratic programming is considered. Also for this method an example is given in the appendix.

Finally Sequential Quadratic Programming will be described and an example is given in the appendix

2.2.1. Lagrange's Method

Suppose the optimization of a function of two variables $f(x_1, x_2)$ subjected to the constraint $g(x_1, x_2) = 0$. The constraint equation can be solved for one of the variables, say x_1 resulting in:

$$x_1 = \phi(x_2). \quad (2.1a)$$

The new constraint is:

$$h(x_2) = g(\phi(x_2), x_2) = 0. \quad (2.1b)$$

If the Variables in Equation 2.1a are substituted in the objective function:

$$F(x_2) = f(\phi(x_2), x_2), \quad (2.2)$$

the original constraint problem is now an unconstraint problem.

At the optimum x^* the derivative of Equation 2.2 must disappear resulting in:

$$\frac{dF}{dx_2}(x_2^*) = 0. \quad (2.3)$$

However:

$$\frac{dF}{dx_2} = \frac{\partial f}{\partial x_2} + \frac{\partial f}{\partial x_1} \frac{\partial \phi}{\partial x_2}; \quad (2.4a)$$

and:

$$\frac{dh}{dx_2} = \frac{\partial g}{\partial x_2} + \frac{\partial g}{\partial x_1} \frac{\partial \phi}{\partial x_2} = 0. \quad (2.4b)$$

From Equation 2.4b:

$$\frac{d\phi}{dx_2} = - \left(\frac{\partial g / \partial x_2}{\partial g / \partial x_2} \right) \quad (2.5a)$$

then Equation 2.3 becomes:

$$\frac{\partial f}{\partial x_2} - \left(\frac{\partial f / \partial x_1}{\partial g / \partial x_1} \right) \frac{\partial g}{\partial x_2} = 0 . \quad (2.5b)$$

The following identity can be written:

$$\frac{\partial f}{\partial x_1} - \left(\frac{\partial f / \partial x_1}{\partial g / \partial x_1} \right) \frac{\partial g}{\partial x_1} = 0 , \quad (2.5c)$$

adding Equation 2.5b and 2.5c and simplifying leads to:

$$\nabla f - \lambda \nabla g = 0 , \quad (2.5d)$$

where:

$$\lambda = \left(\frac{\partial f / \partial x_1}{\partial g / \partial x_1} \right) . \quad (2.6a)$$

In Equation 2.6a, λ is known as the Lagrange multiplier [1].

Equation 2.6a suggests that Lagrange multiplier measures the change of the objective function with respect to the constraint. In physical terms it represents a measure of the effect of the change in the objective function with respect to the right hand side of the constraint equation resulting in:

$$\frac{\partial f}{\partial b_i} = -\lambda_i . \quad (2.6b)$$

Geometric Interpretation of a Lagrange Multiplier

Consider the problem of optimizing a function of two variables $f(x_1, x_2)$ subjected to constraint $g(x_1, x_2) = c$. Several levels of curve result from $f(x_1, x_2) = k$. The graph of typical level curves of f for different k values are presented in Figure 2-3.

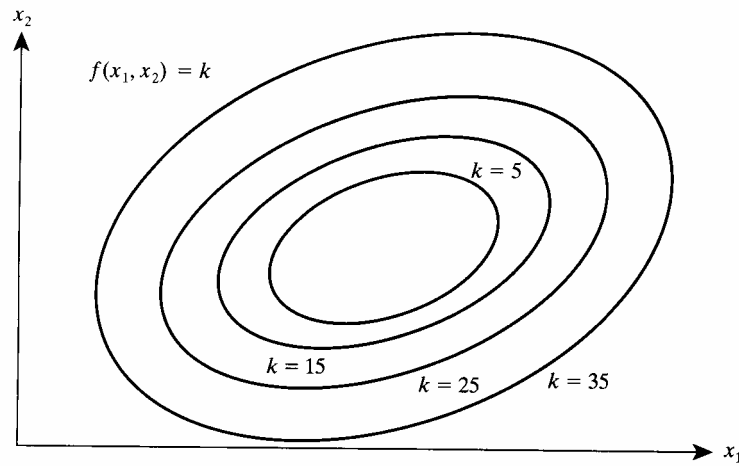


Figure 2-3. A function of two variables with different function values [1]

To optimize the function subject to the given constraint, the optimal value of k must be determined such that a constraint intersects an objective function curve. The optimal point is generally at the point where the appropriate level curve is tangent to the constraint curve, provided the tangent exists at that point. The exact location of this point is given with the help of the theorem given by Purcell in [1], which says that the gradient of a function f at x_0 is perpendicular to the level curve of f that passes through x_0 .

It follows that if the level curve of f is tangent to the constraint curve g , they must both have a common tangent and a common normal.

These normals are in directions of $\nabla f(x_0)$ and $\nabla g(x_0)$ and are parallel (Figure 2-4).

This means:

$$\nabla f(x) = \lambda \nabla g(x_0). \quad (2.7)$$

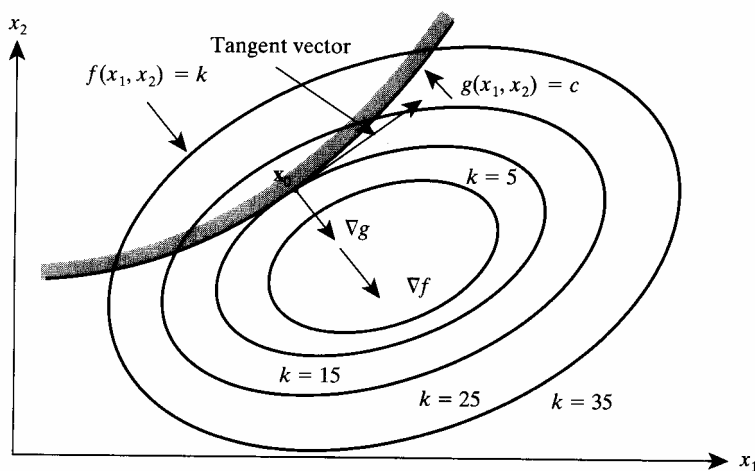


Figure 2-4. The relationship between ∇f and ∇g at the point of tangency [1]

In addition:

$$\begin{aligned} g(x_0) - c &= 0; \\ \lambda &\neq 0. \end{aligned} \tag{2.8}$$

Equations 2.7 and 2.8 can be solved for x_0 and λ . Point x_0 is a critical point for the constrained problem and must be an optimum, at least in local sense.

The constrained problem can be extended to handle the problems with inequality constraints by introducing surplus or slack variables that convert the inequality constraint to an equality constraint.

In general the object is to minimize the equation in the following form:

$$f(x), \quad x \in R^n;$$

which is subjected to the following constraints:

$$\begin{aligned} h_i(x) &= 0 & i=1,2,\dots,k; \\ g_i(x) &\geq 0 & i=k+1,k+2,\dots,m; \end{aligned} \tag{2.9}$$

then the Lagrange problem is stated as minimize the function $L(x, \lambda)$, where L is given by

$$L(x, \lambda) = f(x) + \sum_{i=1}^k \lambda_i (h_i(x)) + \sum_{i=k+1}^m (\lambda_i g_i(x) - x_{n+i}^2) \tag{2.10}$$

Equation 2.10 referred to as the Lagrange function, where x_{n+i} are artificial variables useful in stating the optimality condition for constrained optimization. The most general way in stating the optimality condition is by the use of Karush-Kuhn-Tucker conditions, which are explained in the following section.

Karush-Kuhn-Tucker Optimality conditions

Search methods for constrained optimization were not able to utilize an optimality condition. Other techniques rely on the optimality condition to qualify the optimum obtained, as at least a local optimum. If the objective function $f(x)$ and the constraints $g(x)$ and $h(x)$ are once differentiable then the point x^* is a local minimum if the following equation:

$$\nabla f(x^*) + \sum_{i=1}^k \lambda_i \nabla h_i(x^*) - \sum_{i=k+1}^m \lambda_i \nabla g_i(x^*) = 0; \tag{2.11}$$

which is subjected to the following constraints:

$$h_i(x^*) = 0, \quad i = 1, 2, \dots, k; \tag{2.12}$$

$$g_i(x^*) \geq 0, \quad i = k+1, k+2, \dots, m; \tag{2.13}$$

$$\lambda_i g_i(x^*) = 0, \quad i = k+1, k+2, \dots, m; \tag{2.14}$$

$$\lambda_i \geq 0, \quad i = k+1, k+2, \dots, m. \tag{2.15}$$

The vector (x^*, λ^*) that satisfies Equations 2.11 to 2.15 is known as the Karush-Kuhn-Tucker (KKT) point. KKT conditions are conditions necessary and sufficient for optimality based on the first order conditions. If the gradients of the objective and constrained functions are utilized in the optimization process, it may become necessary to consider the effect of the curvature of the given functions. In this situation, the first order method is inadequate and a second order method is required.

The optimality conditions which apply to twice differentiable functions have been given in [1]. In what follows it is assumed that both the objective functions and the constraints are twice differentiable at the point x^* .

Second-Order Necessary Conditions

The necessary conditions for the point x^* to be local minimum to the problem given in equation 2.9 are:

1. There exists a KKT point (x^*, λ^*) .
2. For every nonzero vector u satisfying:

$$u^T \nabla h_i(x^*) = 0 \quad i = 1, 2, \dots, k; \quad (2.16)$$

$$u^T \nabla g_i(x^*) = 0 \quad i = k + 1, k + 2, \dots, m; \quad (2.17)$$

then

$$u^T \nabla^2 L(x^*, \lambda^*) u \geq 0. \quad (2.18)$$

2.2.2. General quadratic programming

The general quadratic programming problem may be written as, the maximization of the following equation:

$$f(x) = x^T Qx + c^T x;$$

which is subjected to the following constraints:

$$\begin{aligned} Ax &\leq b; \\ x &\geq 0. \end{aligned} \quad (2.19)$$

It is assumed that the quadratic form $x^T Qx$ is either negative-definite or negative-semi-definite and $f(x)$ is strictly concave. Also, Q and A are real matrices of orders $(n \times n)$ and $(m \times n)$ respectively, with n being the number of variables and m the number of constraints. Note that c is an $(n \times 1)$ vector.

The solution may be obtained by application of previous explained Kuhn-Tucker necessary conditions. If the constraint is in equality form, the problem reduces to a system of linear equation because application of the Kuhn-Tucker conditions to quadratic functions results in linear functions. To continue the case of inequality shall be explained.

For the constraints of 2.19 to be compatible with Equation 2.9 the following must be true:

$$Ax - b \leq 0 ; \quad (2.20)$$

$$-x \leq 0 . \quad (2.21)$$

The Lagrangian function for Equation 2.19 is:

$$L(x, \lambda) = x^T Qx + c^T x + \lambda^T (b - Ax) + \mu^T x . \quad (2.22)$$

Application of Kuhn-Tucker condition to Equation 2.22 gives:

$$\begin{aligned} 2Qx^* + c - A^T \lambda^T + \mu^T &= 0 ; \\ \lambda(Ax^* - b) &\leq 0 ; \\ Ax^* - b &\leq 0 ; \\ \mu x^* &\leq 0 ; \\ -x^* &\leq 0 ; \\ \lambda^* &\geq 0 ; \\ \mu^* &\geq 0 . \end{aligned} \quad (2.23)$$

Note that $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ and $\mu = (\mu_1, \mu_2, \dots, \mu_n)$ are the Lagrange multipliers for Equation 2.20 and 2.21 respectively. Moreover note that the subscript * is for the condition at the optimum. If it is dropped and slack variables $u = (u_1, u_2, \dots, u_n)$ are introduced, the KKT conditions can be written as:

$$2Qx - A^T \lambda + \mu = 0 ; \quad (2.24)$$

$$Ax - b + u = 0 ; \quad (2.25)$$

$$\mu x = 0 ; \quad (2.26)$$

$$\lambda u = 0 . \quad (2.27)$$

The optimum for Equation 2.19 can be obtained by finding x, u, λ and μ that satisfy Equations 2.24 to 2.27. All these variables except λ are restricted to either zero or positive values. The sign of an element of λ can be ≥ 0 , ≤ 0 or unrestricted depending on whether the sign on the constraint in Equation 2.19 is $\geq, \leq$ or $=$.

Note that the solution obtained by solving Equations 2.24 and 2.25 is a global optimum for Equation 2.19 because $f(x)$ is strictly concave and the feasible region is convex.

Moreover, note that Equations 2.24 and 2.25 are in a linear programming form and hence the solution can be obtained by the use of phase 1 of the two-phase simplex method. If the phase 1 approach is used, special precaution must be taken to satisfy Equations 2.26 and 2.27. If an element of μ , for example μ_i , is a basic variable, then the corresponding element of x , for example x_i must not be allowed to become a basic variable, otherwise Equation 2.26 cannot be satisfied. If the pivoting rule in

the simplex suggests that x_i should become a basic variable when μ_i is already in the basis, then the next best variable should be selected. The same reasoning applies to Equation 2.27.

Hessian matrix of $f(x)$

If $f(x)$ is twice continuously differentiable then at the point x there exists a matrix of second order partial derivatives or Hessian matrix:

$$H(x) = \left\{ \frac{\partial^2 f}{\partial x_i \partial x_j}(x) \right\} = \nabla^2 f(x)$$

$$= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \dots \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & & \\ \vdots & & \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \dots & \frac{\partial^2 f}{\partial x_n^2}(x) \end{bmatrix} \quad (2.28)$$

2.2.3. Sequential Quadratic Programming

The sequential quadratic programming (SQP) method is considered one of the most promising techniques for problems dealing with nonlinear constraint optimization. Implemented in the 1970s [2] and [3], the basic idea behind the method was given [44]. Many variations of this technique have been presented by several authors [4], [5] and [6]. The method has become popular, because it has the property of finding the optimum from an arbitrary starting design point. In addition, it requires fewer function and gradient evaluations compared to other methods for constrained optimization.

The method consists of approximating the Equation 2.9 with a quadratic programming subproblem and solving the subproblem successively until convergence has been achieved on the original problem. Thus for, Equation 2.9 with the binding constraint modified for the quadratic subproblem, $P(d)$ may be represented as the minimization of the following equation:

$$P = \nabla f_x^T d + 0.5d^T d, \quad (2.29)$$

which is subjected to the following constraints:

$$\nabla h^T d + h(x) = 0; \quad (2.30)$$

$$\nabla g_x^T d + g(x) \leq 0. \quad (2.31)$$

After the optimum of the problem defined by Equations 2.29 to 2.31 is obtained, it is then used to obtain the new value of the original variable of the original problem at the k^{th} iteration as:

$$x^{k+1} = x^k + d^k . \quad (2.32)$$

Using Equation 2.32, the problem is linearized through the new value of x . The processes are continued until a specified tolerance.

Chapter 3. Stochastic Methods

Stochastic methods are the synonym of random search. In essence, it simply consists in picking up random potential solutions and evaluating them. The best solution over a number of samples is the result of random search.

These methods can be divided in Simulated Annealing (SA) and evolutionary algorithms.

More over evolutionary algorithms can be divided in Genetic Algorithms (GA's) and Evolutionary Strategies (ES).

3.1. Simulated Annealing

Simulated Annealing (SA) [7] is motivated by an analogy to annealing in solids. The idea of SA comes from a paper published by Metropolis in 1953. The algorithm in this paper simulated the cooling of material in a heat bath. This is a process known as annealing.

If you heat a solid past melting point and then cool it, the structural properties of the solid depend on the rate of cooling. If the liquid is cooled slowly enough, large crystals will be formed. However, if the liquid is cooled quickly (quenched) the crystals will contain imperfections.

Metropolis's algorithm simulated the material as a system of particles. The algorithm simulates the cooling process by gradually lowering the temperature of the system until it converges to a steady, frozen state.

In 1982, Kirkpatrick took the idea of the Metropolis algorithm and applied it to optimisation problems. The idea is to use simulated annealing to search for feasible solutions and converge to an optimal solution.

3.1.1. Acceptance Criteria

The law of thermodynamics state that at temperature, t , the probability of an increase in energy of magnitude, ΔE , is given by

$$p(\Delta E) = e^{\frac{-\Delta E}{kT}}, \quad (3.1)$$

where T is the temperature of the body and k is the Boltzmann's constant.

Metropolis observed that the probability of higher energy is larger at higher temperatures and there is some chance of a high energy as the temperature drops.

Energy in the annealing process sometimes increases even while the trend is a net decrease.

This property applied to optimization problems is referred to as the Metropolis algorithm. In optimization applications the temperature is initialized at a high level. Boltzmann's constant may be

chosen as equal to 1. The change Δf in the function value is accepted whenever it represents a decrease if a minimization is applied. When it is an increase, it is accepted with a probability:

$$p(\Delta E) = e^{\frac{-\Delta f}{T}}. \quad (3.2)$$

This is accomplished by generating a random number r in the range 0 to 1 and accepting the new value when $r \leq p$. Metropolis algorithm has been extended, coded and further explored [8].

3.1.2. Relationship between Physical Annealing and Simulated Annealing

Table III-1 [51] shows how physical annealing can be mapped to simulated annealing.

Thermodynamic Simulation	Optimization
System States	Feasible Solutions
Energy	objective
Change of State	Neighbouring Solutions
Temperature	Control Parameter
Frozen State	Heuristic Solution

Table III-1. Relationship between Physical and (SA) [51]

Using these mappings any optimization problem can be converted into an annealing algorithm [9] and [10] by sampling the neighbourhood randomly and accepting worse solutions using Equation 3.2.

Furthermore, heuristic is defined as a method of problem solving using exploration and trial and error methods [52].

3.1.2. Implementation of Simulated Annealing in a problem

The objective of the algorithm is to minimize the following function:

$$f(x),$$

which is subject to the following constraints:

$$l_i \leq x_i \leq u_i \quad i=1 \text{ to } n.$$

In this problem, each variable has lower and upper bounds. The search for the minimum is initiated at a feasible starting point x . Function value f is evaluated at x while $x_{\min} = x$, and $f_{\min} = f$ is set. A vector s of step sizes with step size s_i along the coordinate direction e_i is chosen. Initially each s_i may be set equal to a step size s^T ($=1$) and a step reduction parameter r_s chosen. A vector a of acceptance ratios with each element equal to 1 is defined. A starting temperature T and a temperature reduction factor r_T are chosen. Let say r_T and r_s have been chosen as 0.9 and 0.5 [11], other values

may be tried. The main loop is the temperature loop where the temperature is set as $r_T T$ and the step size set as $r_s r_T$ at the end of a temperature step.

At each temperature, $N_T (= 5)$ [11] iterations are performed. Each iteration consists of N_c cycles. A cycle involves taking a random step in each of the n directions successively. A step in a direction i is taken in the following manner. A random number r in the range -1 to 1 is generated and a new point x_s is evaluated using:

$$x_s = x + r s_i e_i. \quad (3.3)$$

If this point is outside the bounds, the i th component of x_s is adjusted to be a random point in the interval l_i to u_i . The function value f_s is then evaluated. If $f_s \leq f$ then the point is accepted by setting $x = x_s$. If $f_s \leq f_{\min}$ then $f_{\min}$ and $x_{\min}$ are updated. If $f_s > f$ then it is accepted with a probability of $p = e^{\frac{f-f_s}{T}}$.

A random number r is generated and f_s is accepted if $r < p$. This is referred to as the Metropolis criterion. Whenever a rejection takes place, the acceptance ratio a_i , which is the ratio of the number of acceptances to the total number of trials for direction i is updated as:

$$a_i = a_i - \frac{1}{N_c}. \quad (3.4)$$

At the end of N_c cycles, the value of the acceptance ratio a_i is used to update the step size for the direction. A low value implies that there are more rejections, suggesting that the step size is to be reduced. A high rate indicates more acceptances which may be due to small step size. In this case step size is to be increased. If $a_i = 0.5$, the current step size is adequate with the number of acceptances at the same level as that of rejections. Once again drawing from the work of Metropolis on Monte Carlo simulations of fluids, the idea is to adjust the steps to achieve the ratio of acceptances to rejections equal to 1. A step multiplication factor, $g(a_i)$, is introduced as:

$$s_i^{new} = g(a_i) s_i^{old}. \quad (3.5)$$

3.1.4. Start Temperature (start control parameter)

The starting temperature must be hot enough to allow a move to almost any neighbourhood state. If this is not done then the ending solution will be the same (or very close) to the starting solution.

However, if the temperature starts at value, which is too high, then the search can move to any neighbour and thus transform the search (at least in the early stages) into a random search.

The problem is finding the correct starting temperature. At present, there is no known method for finding a suitable starting temperature for a whole range of problems. Therefore, we need to consider other ways.

If the maximum distance (objective function difference) between one neighbour and another is known, then this information can be used to calculate a starting temperature.

Another method [12], is to start with a very high temperature and cool it rapidly until about 60% of worst solutions are being accepted. This forms the real starting temperature and it can now be cooled more slowly.

A similar idea [13], is to rapidly heat the system until a certain proportion of worse solutions are accepted and then slow cooling can start. This can be seen to be similar to how physical annealing works in that the material is heated until it is liquid and then cooling begins (i.e. once the material is a liquid it is pointless carrying on heating it).

3.1.5. Final Temperature (final control parameter)

It is usual to let the temperature decrease until it reaches zero. However, this can make the algorithm run for a lot longer, especially when a geometric cooling schedule is being used (see below).

In practise, it is not necessary to let the temperature reach zero because as it approaches zero the chances of accepting a worse move are almost the same as the temperature being equal to zero.

Therefore, the stopping criteria can either be a suitably low temperature or when the system is “frozen” at the current temperature (i.e. no better or worse moves are being accepted).

3.2. (EAs) Evolutionary algorithms

In general these algorithms are inspired by the principles of natural evolution to find an optimal solution to a problem. Natural evolution is driven by the principles of selection, recombination and mutation of genetic information. Individuals in a population which are well adapted to their environment have a higher probability to survive in the nature, known as '*survival of the fittest*'. These individuals are declared with a higher fitness value and are chosen in order to become parents (selection) which produce offsprings for the following generation. The genetic information of the offspring is either a direct copy of the genes of just one single parent, which differs from the natural evolution, or results from the mating process of multiple parents (recombination or crossover). In the latter case the gene of the offspring is arranged from gene sequences of both parents. Additionally, a randomly generated mutation can modify the genetic information of the offspring's (mutation) and the best solutions are selected in the selection process.

Figure 3-1 shows the principle work flow of an evolutionary algorithm consisting of crossover, mutation and selection.

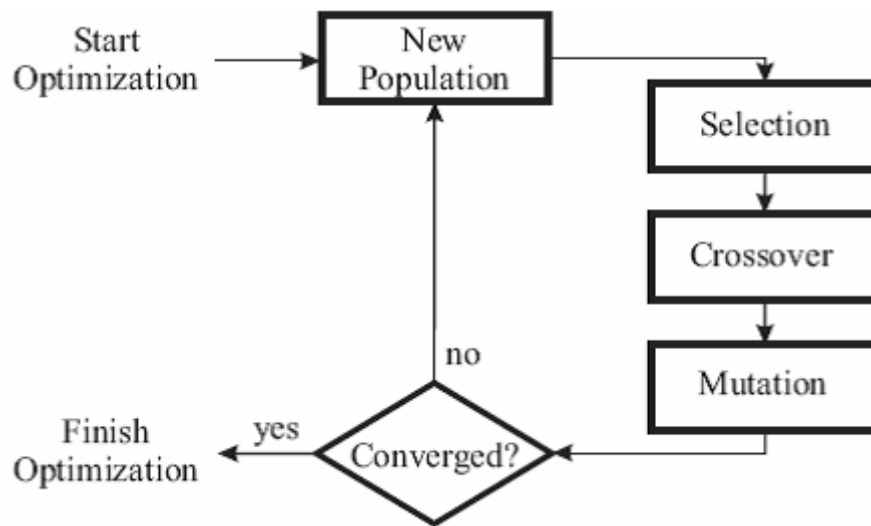


Figure 3-1. Principle work flow of an evolutionary algorithm

Applied to engineering design problems, the genetic information corresponds to the design variables which specify the properties of a solution to the engineering optimization problem and the fitness of a solution is either determined directly by the objective function or by a combination between the objective function value and the constraints.

It should be mentioned that (EAs) are building an upper class of algorithms containing the subgroups of Genetic Algorithms (GAs) and Evolutionary Strategies (ES). GAs were firstly proposed and applied [14] while Rechenberg [15] developed Evolutionary Strategies independently and applied them to some engineering design problems. Although based on a similar idea, both approaches were different in some aspects at the beginning of their development.

In GAs an individual is represented by a string of bits and the evolutionary process is based on selection, recombination, and mutation techniques. In contrast to this, in ES each individual of the population is represented by a vector of real design variables and the evolutionary process is characterized by selection and mutation techniques only.

Furthermore, because both approaches are based on a similar idea and the popularity of GAs in engineering, only GAs will be explained in this section.

3.2.1. Introduction to Genetic Algorithms (GAs)

A large class of interesting problems for which no reasonably fast algorithms have been developed exists. Many of these problems are optimization problems that arise frequently in applications. Given such a hard optimization problem, it is often possible to find an efficient algorithm whose solution is approximately optimal. For some hard optimization problems, probabilistic algorithms can be used, but these algorithms do not guarantee the optimum value; anyway by randomly choosing sufficiently many “witnesses”, the probability of error may be reduced as much as possible.

For small spaces, classical exhaustive methods usually suffice; for larger spaces, special artificial intelligence techniques must be employed. GAs are among such techniques. They are adaptive

heuristic search algorithms premised on the evolutionary ideas of natural selection and genetic. The basic concept of GAs is to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. The metaphor underlying genetic algorithms is that of natural evolution. In evolution, the problem each species faces is searching for beneficial adaptations to a complicated and changing environment. The 'knowledge' that each species has gained is embodied in the makeup of the chromosomes of its members. Thus, they represent an intelligent exploitation of a random search within a defined search space to solve a problem. First pioneered by John Holland in the 60s, GAs have been widely studied, experimented and applied in many fields in the engineering world. Not only do GAs provide alternative methods to solving problem, but they consistently outperform other traditional methods in most of the problems link. Many of the real world problems involved finding optimal parameters, which might prove difficult for traditional methods but ideal for GAs.

The idea behind genetic algorithms is to do what nature does. Take the example [16] of rabbits "at any given time there is a population of rabbits".

Some of them are faster and smarter than other rabbits. These faster, smarter rabbits are less likely to be eaten by foxes and therefore many of them survive to breed. Of course, some of the slower, dumber rabbits survive just because of luck. This surviving population of rabbits starts breeding. The breeding results in a good mixture of rabbit genetic material.

Some slow rabbits breed with fast rabbits, some fast with fast, some smart rabbits with dumb rabbits and so forth.

On the top of that, nature throws in a 'wild hare' every once in a while by mutating some of the rabbit genetic material. The resulting baby rabbits will (on average) be faster and smarter than these in the original population because faster, smarter parents survived the foxes. It has to be noticed that a good thing that the foxes are undergoing similar process otherwise the rabbits might become too fast and smart for the foxes to catch any of them.

A genetic algorithm follows a step-by-step procedure that closely matches the story of the rabbits.

The fundamental principle of natural selection as the main evolutionary principle has been formulated by C. Darwin long before the discovery of genetic mechanisms.

Genetic algorithms use a vocabulary borrowed from natural genetics. In genetic algorithms individuals (or genotypes or structures) are used in a population. Quite often these individuals are called also strings or chromosomes. This might be misleading, because each celi of every organism of a given species carries a certain number of chromosomes. However, in GAs only one chromosome individuals are taken into account. Each single chromosome represents a potential solution to a problem. An evolution process run on a population of chromosomes corresponds to a search through a space of potential solutions. Such a search requires balancing two (apparently conflicting) objectives: exploiting the best solutions and exploring the search space.

GAs are a class of general purpose (domain independent) search methods which strike a remarkable balance between exploration and exploitation of the search space.

GAs have been quite successfully applied to optimization problems like wire routing, scheduling, adaptive control, game playing, cognitive modelling, transportation problems, travelling salesman problems, optimal control problems, database query optimization any many others.

During the last decade, the significance of optimization has grown even further, because many important large-scale combinatorial optimization problems and highly constrained engineering problems can only be solved approximately on present day computers.

Genetic algorithms aim at such complex problems. They belong to the class of probabilistic algorithms, yet they are very different from random algorithms as they combine elements of directed and stochastic search, since GAs are also more robust than existing directed search methods. Another important property of such genetic based search methods is that they maintain a population of potential solutions, while all other methods process a single point of the search space.

As mentioned earlier, a GA performs a multi-directional search by maintaining a population of potential solutions and encourages information formation and exchange between these directions. The population undergoes a simulated evolution at each generation. The relatively good solutions reproduce, while the relatively bad solutions die.

The structure of a simple genetic algorithm is the same as the structure of any evolution program (Figure 3-2). During iteration (t), a genetic algorithm maintains a population of potential solutions (chromosomes, vectors, variables), $P(t) = \{\mathbf{x}'_1, \dots, \mathbf{x}'_n\}$. Each solution $\mathbf{x}'_i$ is evaluated to give some measure of its “fitness”. Most of the effort is spent on the reproduction operators: selection, crossover and mutation. A new population (iteration $t + 1$) is formed by selecting the most fit individuals. Some members of this new population undergo alterations by means of crossover and mutation, to form new solutions.

Crossover combines the features of two parent chromosomes to form two similar offspring by swapping corresponding segments of the parents.

Mutation arbitrarily alters one or more genes of a selected chromosome, by a random change with a probability equal to the mutation rate. The intuition behind the mutation operator is the introduction of some extra variability into the population.

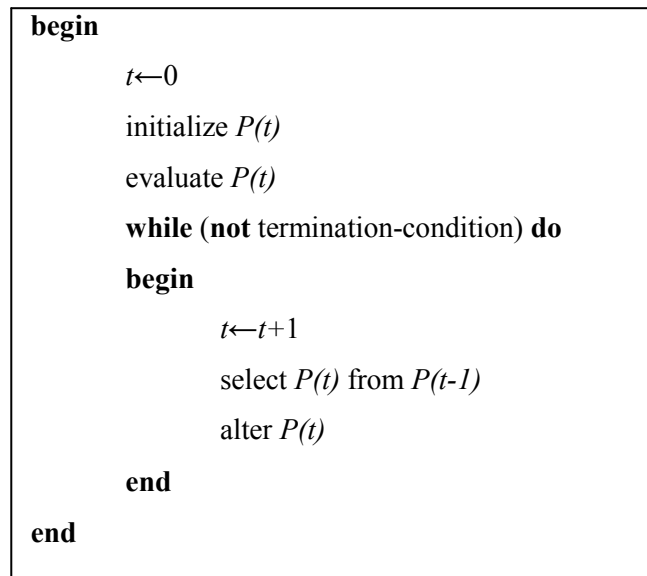


Figure 3-2. The structure of an evolution program [16]

GAs (as any evolution program) for a particular problem must have the following five components:

1. a genetic representation for potential solutions to the problem,
2. a way to create an initial population of potential solutions,
3. an evaluation function that plays the role of the environment, rating solutions in terms of their “fitness”,
4. genetic operators that alter the composition of children,
5. values for various parameters that the GAs uses (population size, probabilities of applying genetic operators and so forth).

3.2.2. Selection Operator

The reproduction process, like in biology, requires the selection of mates. In genetic algorithms, two chromosomes called parents are randomly selected to reproduce two new offspring. Selection basically involves four steps. First, the values of the objective function are determined for each chromosome in the current population. Second, a fitness function for each chromosome is determined. Third, the probability of each chromosome being selected is determined. Finally, the actual selection is made.

The fitness value is determined from the value of the objective function.

This value is often referred to as the raw fitness value. In some instances, raw fitness values are used as fitness values only when they are positive. If they are negative, then constants are added to the raw fitness values to transform them into positive fitness functions. This modification works particularly well with the popular selection method known as the proportional selection algorithm.

The fitness function $U(x)$ [17], for a minimization problem is as follows:

$$U(x) = \begin{cases} C_{\max} - f(x) & \text{if } f(x) < C_{\max} \\ 0 & \text{otherwise} \end{cases},$$

where $C_{\max}$ is a constant that can be chosen in a number of ways. In one method, the largest value of $f(x)$ in the current population is used. If the problem is one of maximization, the transformation is carried out as:

$$U(x) = \begin{cases} f(x) + C_{\min} & \text{if } f(x) + C_{\min} > 0 \\ 0 & \text{otherwise} \end{cases},$$

where $C_{\min}$ is a constant. A good value is the absolute value of the worst raw fitness value in the current population. However, in the Boltzmann selection procedure [18], the raw fitness value is transformed into a fitness function by the following equation:

$$U(x) = e^{f(x)/T},$$

where T is a tolerance value that is adjusted during the succeeding generations.

A good initial starting value reported in literature is 4 [1].

Many selection methods such as linear rank selection have been presented [19].

Another method is the proportional selection method. In this method, first the probability P_i of a particular chromosome being selected is determined using the following equation:

$$P_i = \frac{U_i(x)}{\sum_{k=1}^n U_k(x)},$$

where n is the size of the population. Once the probability of selecting a particular chromosome is determined, the actual selection is carried out.

A more popular selection procedure at this juncture is the roulette wheel [1]. The roulette wheel is divided into n slots weighted in proportion to the fitness

values of the chromosomes (or population member). The wheel is "spun" and the selected member is the slot corresponding to the final position of the spinner. Clearly, the members with higher fitness values have higher chances of being selected. The outcome of the spin of the roulette wheel thus leads to parents being randomly selected. The underlying principle is quite simple. A random number generator yields a number, $y \in [0, 1]$. The value of y is multiplied by the sum of the total fitness values in the population. Then a process of adding the fitness values begins with the first member of the current population. The first member at which the current value of the partial sum of the fitness values exceeds or equals the product of y and the total sum of the fitness values is the parent selected. This procedure is shown in Figure 3-3.

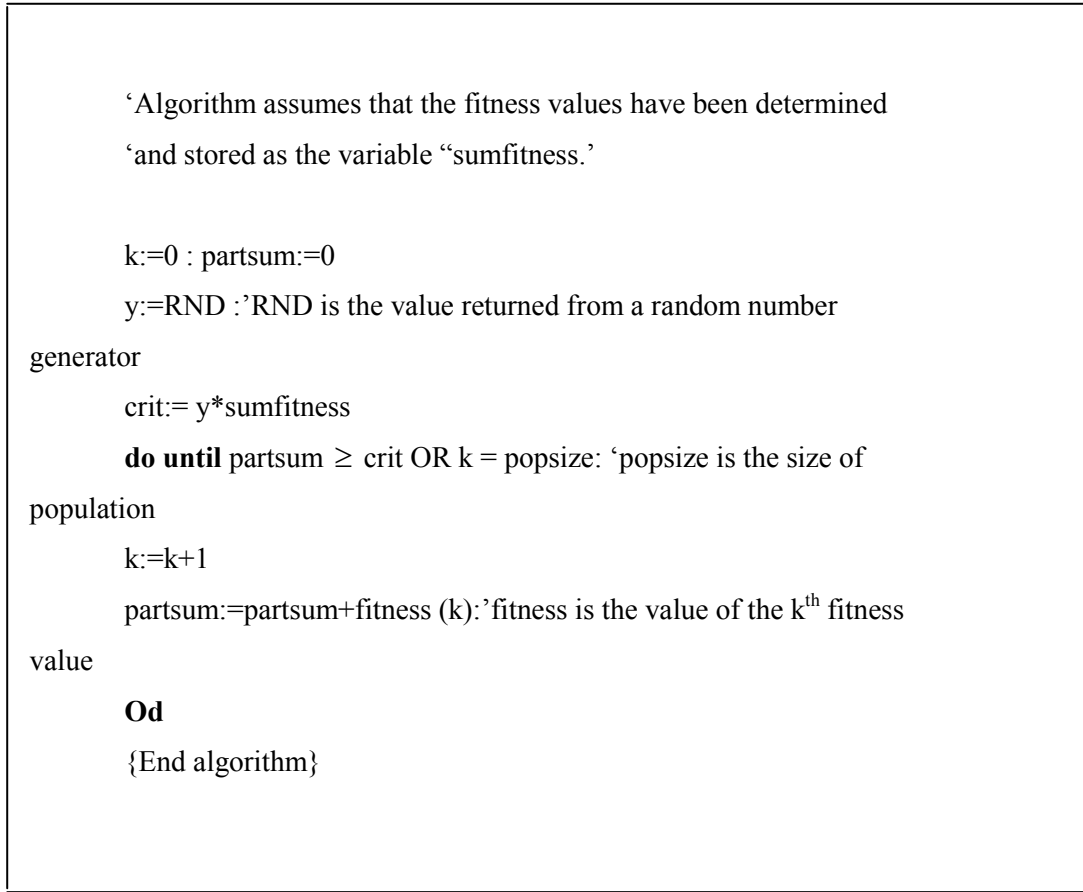


Figure 3-3. Roulette Wheel Parent Selection Procedure [16]

3.2.3. Mutation Operator

The mutation process is a "flipping" operation that results in each bit in the binary string being changed from 0 to 1, or vice versa according to a random process. A bit is changed if the random number associated with it is less than or equal to a preset probability value, $p(=0.2)$ [1], which is often set quite low. The process is very simple, and is illustrated in table III-2

Chromosome before Mutation	Random Numbers	Chromosome after Mutation
111000	0.0048 0.8459 0.4483 0.9517 0.4475 0.1295	011001
111001	0.0130 0.2616 0.7553 0.7434 0.8090 0.2099	011001
010111	0.1519 0.0793 0.4744 0.4443 0.0065 0.0920	010101
100011	0.3877 0.5035 0.5535 0.8802 0.3959 0.5825	100011

Table III-2 Example of mutation process

Several observations can be made from Table III-2. Since the chromosome has length 6, the mutation process is carried out six times for each bit in the string. Then, six random numbers are generated for each chromosome.

At last, the bit is flipped only if the random number is less than the specified probability. Thus, bit 1 for random number 0.0048 is flipped to 0 since $0.0048 < 0.2$, but bit 1 for random number 0.8459 does not flip.

3.2.4. The Crossover Operator

The mutation and crossover operators are essential in assuring that the new chromosomes are different than their parents. The crossover process involves a swap of bits in specified locations of two parent chromosomes. A popular method of crossover is the one-point crossover [1]. Two-point crossovers and multipoint crossovers are also used [20]. A one-point crossover occurs when parts of two parent chromosomes are swapped at a specified position. In this operation, an integer number k is randomly selected between 1 and $l - 1$ according to a uniform distribution, where l is the length of the binary string. The number k is referred to as the crossover site. All bits between the positions $k + 1$ and l are swapped to create two new binary strings. Let's illustrate this method with an example.

Consider two chromosomes given by:

$$\begin{aligned} Y_1 &:= 10110101 ; \\ Y_2 &:= 00001110 . \end{aligned}$$

The crossover site is specified as 5. Then with $l = 8$, all characters between 6 and 8, are swapped resulting in:

$$\begin{aligned} Y'_1 &:= 10110110 ; \\ Y'_2 &:= 00001101 . \end{aligned}$$

Two factors are fundamental in the crossover process. First, it is possible that after a crossover the new chromosomes will be radically different from their parent chromosomes. Second, if the bits have the same values at the same location on the chromosomes, no difference results from the crossover. This is the case in the example because both original chromosomes had values equal to 1 in the sixth position. Note that if the parent chromosomes are identical from the crossover site to the end of the string, the new chromosomes will be identical to each other and their generating parents. Since crossover is a random process, a probability for crossover is generally specified. Typical good values are generally between 0.6 and 0.8. As stated earlier, a crossover takes place if the random number generated is less or equal to the specified probability of crossover.

This method is the original method [14] but recent efforts have attempted to improve the crossover operators by making them relevant to the objective function [45].

3.2.5. How do genetic algorithms work

In this section, an explanation will be given about how GAs work.

Suppose the intention is to maximize a function $f(x_1, \dots, x_k)$ of k variables, Suppose further that each variable x_i can take values from a domain $D_i = [a_i, b_i] \subseteq R$ and $f(x_1, \dots, x_k) > 0$ for all $x_i \in D_i$ and the function must be optimized with some precision: suppose six decimal places for the variables values is desirable.

It is clear that to achieve such precision each domain D_i should be cut into $(b_i - a_i)10^6$ equal size ranges. The smallest integer is denoted by m_i such that $(b_i - a_i)10^6 \leq 2^{m_i} - 1$. Then a representation having each variable x_i coded as a binary string of length m_i clearly satisfies the precision requirement. Additionally the following formula interprets each such string:

$$x_i = a_i + \text{decimal}(1001 \dots 001_2) \frac{b_i - a_i}{2^{m_i} - 1},$$

where *decimal* (*string*₂) represents the decimal value of that binary string.

Now, each chromosome (as a potential solution) is represented by a binary string of length $m = \sum_{i=1}^k m_i$; the first m_1 bits map into a value from the range $[a_1, b_1]$, the next group of m_2 bits map into a value from the range $[a_2, b_2]$, and so on; the last group of m_k bits map into a value from the range $[a_k, b_k]$.

To initialize a population, some *pop_size* number of chromosomes can be set randomly in a bitwise fashion. However, if some knowledge about the distribution of potential optima is available, such information can be used to arrange the set of initial (potential) solutions.

The rest of the algorithm is straightforward: in each generation each chromosome is evaluated (using the function f on the decoded sequences of variables), new population is selected with respect to the probability distribution based on fitness values and the chromosomes are altered in the new population by mutation and crossover operators. After some number of generations, when no further improvement is observed, the best chromosome represents an optimal solution (possibly the global).

The algorithm often stops after a fixed number of iterations, depending on speed and resource criteria.

For the selection process (selection of a new population with respect to the probability distribution based on fitness values), a roulette wheel with slots sized according to fitness is used.

A roulette wheel is constructed as follows:

- Calculate the fitness value *eval* (v_i) for each chromosome v_i ($i = 1, \dots, \text{popsize}$);
- Find the total fitness of the population $F = \sum_{i=1}^{\text{pop_size}} \text{eval}(v_i)$;
- Calculate the probability of a selection p_i for each chromosome v_i ($i = 1, \dots, \text{popsize}$):
 - $p_i = \text{eval}(v_i)/F$.

- Calculate a cumulative probability q_i for each chromosome v_i ($i = 1, \dots, popsize$):

$$q_i = \sum_{j=1}^i p_j.$$

The selection process is based on spinning the roulette wheel pop_size times;

each time a single chromosome is selected for a new population in the following way:

- Generate a random (float) number r from the range $[0,1]$;
- If $r < q_1$ then select the first chromosome (v_1); otherwise select the i -th chromosome v_i ($2 \leq i \leq pop_size$) such that $q_{i-1} < r \leq q_i$.

Obviously, some chromosomes would be selected more than once.

Now it's time to apply the recombination operator, crossover, to the individuals in the new population.

As mentioned earlier, one of the parameters of a genetic system is probability of crossover p_c . This probability gives the expected number $p_c \cdot pop_size$ of chromosomes which undergo the crossover operation. The procedure is continued in the following way:

- For each chromosome in the (new) population:
 - Generate a random (float) number r from the range $[0,1]$;
 - if $r < p_c$, select given chromosome for crossover.

Now the selected chromosomes are mated randomly: for each pair of coupled chromosomes a random integer number pos from the range $[1, m-1]$

(m is the total length —number of bits— in a chromosome) is generated.

The number pos indicates the position of the crossing point. Two chromosomes:

$$(b_1 b_2 \dots b_{pos} b_{pos+1} \dots b_m)$$

and

$$(c_1 c_2 \dots c_{pos} c_{pos+1} \dots c_m),$$

are replaced by a pair of their offspring:

$$(b_1 b_2 \dots b_{pos} c_{pos+1} \dots c_m)$$

and

$$(c_1 c_2 \dots c_{pos} b_{pos+1} \dots b_m).$$

The next operator, mutation, is performed on a bit-by-bit basis. Another parameter of the genetic system, probability of mutation p_m , gives the expected number of mutated bits $p_m \cdot m \cdot pop_size$. Every bit (in all chromosomes from the whole population) has an equal chance to undergo mutation (i.e., change from 0 to 1 or vice versa). So the procedure continues in the following way, for each chromosome in the current (i.e., after crossover) population and for each bit within the chromosome:

- Generate a random (float) number r from the range $[0,1]$;
- If $r < p_m$, mutate the bit.

Following selection, crossover and mutation, the new population is ready for its next evaluation. This evaluation is used to build the probability distribution, (for the next selection process), i.e., for a

construction of a roulette wheel with slots sized according to current fitness values. The rest of the evolution is just cyclic repetition of the above steps (Figure 3-2).

An application of how a GA works will be given in Appendix C.

Chapter 4. Preliminary geometry of the compressor

In this chapter a one-dimensional model to determine the preliminary geometry of the radial compressor will be discussed. But first the attention will be focused on the station numbering and the Euler turbomachinery equation.

Further on the conditions needed to apply one-dimensional model will be discussed in the section of model for determining the geometry of the compressor.

The model is carried out according to the one-dimensional calculations, which will be presented in the following sections.

These one-dimensional calculations consist of impeller inlet design calculations and impeller exit design calculations.

4.1. Stations numbering

Before a detailed investigation of the quantities involved in the analysis of a radial compressor, the attention is focused on the numbering of the stations inside the compressor.

Figure 4-1 shows the definition of this numbering and several other important geometric parameters, which will be used throughout the report with the same notation.

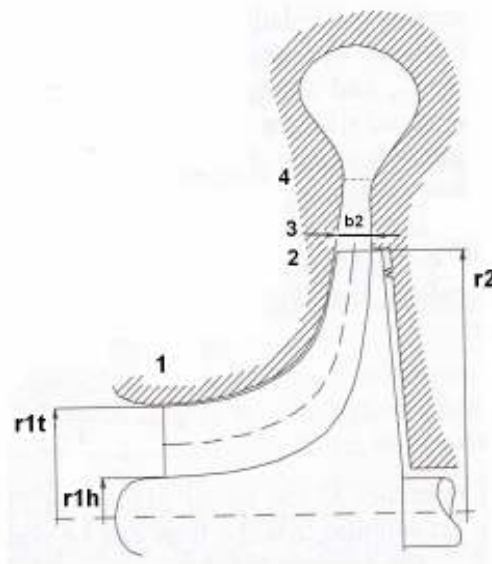


Figure 4-1. Meridional view of a radial compressor [21]

The impeller inlet is marked as station number 1, while station number 2 is located at the impeller exit. Station 3 is the diffuser inlet and station number 4 is the diffuser exit. In Figure 4-1 one can also notice the radii at the impeller inlet hub and tip, respectively r_{1h} and r_{1t} , the radius at the impeller exit r_2 and the blade height b_2 .

4.2. The Euler turbomachinery equation

This equation is of fundamental importance for turbomachinery. It stems directly from the energy and momentum equations applied to a blade row. The Newton Second Law of Motion applied to a rotating system must be initially considered. The torque developed is equal to the rate of change of angular momentum:

$$\tau = \frac{\Delta(mrC_u)}{\Delta t}, \quad (4.1)$$

where m is the mass, r is the considered radius, C_u the component of the absolute velocity in the tangential direction, at that radius, and t the time.

The mass flow rate $\Delta m/\Delta t = \dot{m}$ is constant for steady flow processes. Hence (Figure 4-2, where $C_{\theta 1} = C_{u1}$ and $C_{\theta 2} = C_{u2}$):

$$\tau = \dot{m}\Delta(rC_u) = \dot{m}(r_1C_{u1} - r_2C_{u2}). \quad (4.2)$$

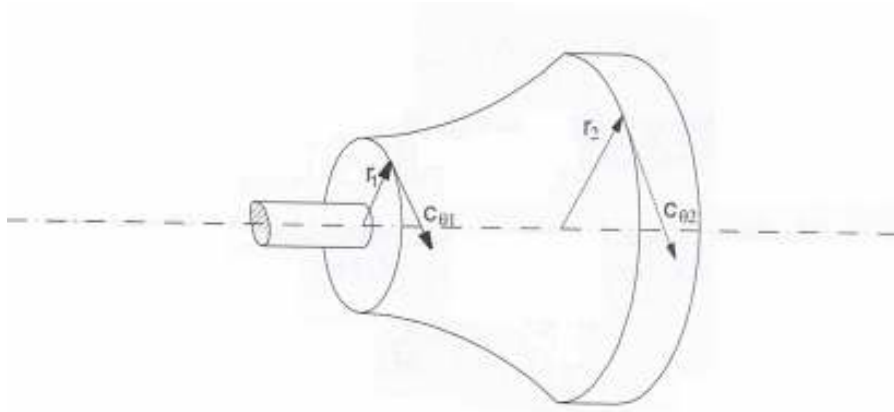


Figure 4-2. The Euler turbomachinery equation [21]

The torque exerted on a fluid element is equal to the mass flow rate times the change of rCu . For a flow in which the torque is zero, rCu is constant. This is called a *free vortex flow*. In the impeller of a compressor, however, the torque exerted on the fluid is not zero. The amount of work per unit mass flow is equal to the product of the torque and the angular velocity ω :

$$W_x = \frac{\tau\omega}{\dot{m}} = \omega(r_1C_{u1} - r_2C_{u2}), \quad (4.3)$$

where $r\omega$ is now the blade speed U .

In addition, the energy equation (First Law of Thermodynamics) states that the work done per unit mass flow is equal to the total enthalpy change in an adiabatic process.

After a substitution, the Euler turbomachinery equation is obtained:

$$\Delta h_0 = h_{01} - h_{02} = U_1C_{u1} - U_2C_{u2}. \quad (4.4)$$

This equation can be applied to turbines, pumps or compressors. It can be applied either to the ideal velocity triangles to determine the ideal enthalpy change (or head), or to the actual velocity triangles to

deduce the actual change in enthalpy. Since it is derived from the energy and momentum equations, it is independent of internal losses of the flow in the blade passages.

4.3. Model for determining the geometry of the compressor

A one-dimensional model has been used to determine the preliminary geometry of the radial compressor.

Furthermore, it has been stated [25] that a one-dimensional model assumes uniform flow conditions due to its own conditions and the air behaves as a perfect gas.

A perfect or ideal gas is a hypothetical gas consisting of identical particles of negligible volume, with no intermolecular forces. Additionally, the constituent atoms or molecules undergo perfectly elastic collisions with the walls of the container. Real gases do not exhibit these exact properties, but the approximation is often good enough to describe them. However, it breaks down at high pressures and low temperatures, where the intermolecular forces play a greater role in determining the properties of the gas. The thermodynamic properties of an ideal gas can be described by two equations:

- The equation of state of a classical ideal gas is given by the ideal gas law:

$$pV = nRT_0 = NkT_0, \quad (4.5)$$

where p is the pressure, V is the volume, n is the amount of gas, R is the gas constant, T_0 is the total temperature, N is the number of particles and k is the Boltzmann constant; $nR = Nk$ is the amount of energy per degree .

- The internal energy of an ideal gas is given by:

$$U = \hat{c}_v nRT_0 = \hat{c}_v NkT_0, \quad (4.6)$$

where U is the internal energy and $\hat{c}_v$ is a constant equal to 3/2 for monatomic gases and to 5/2 for diatomic gases.

4.4. Impeller inlet design calculations

It's important to examine the first station before calculations can be applied.

The sign convention to be employed for velocities and angles is shown in Figure 4-3.

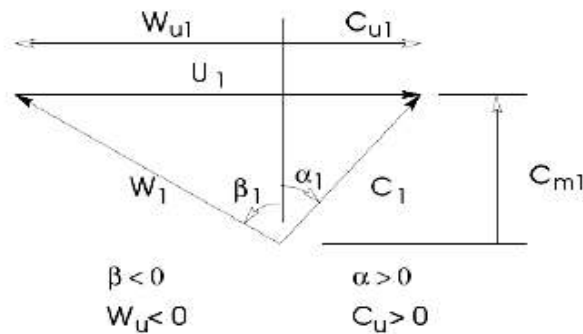


Figure 4-3. Sign convention for velocity triangles [22]

Angles and components of velocity in the direction of the rotation are positive, whereas angles and velocities opposed to it are negative. Thus, in Figure 4-3 the relative flow angle β and the relative velocity W are negative, while the absolute flow angle α and the absolute velocity C are positive. The inlet velocity triangle is shown in Figure 4-4.

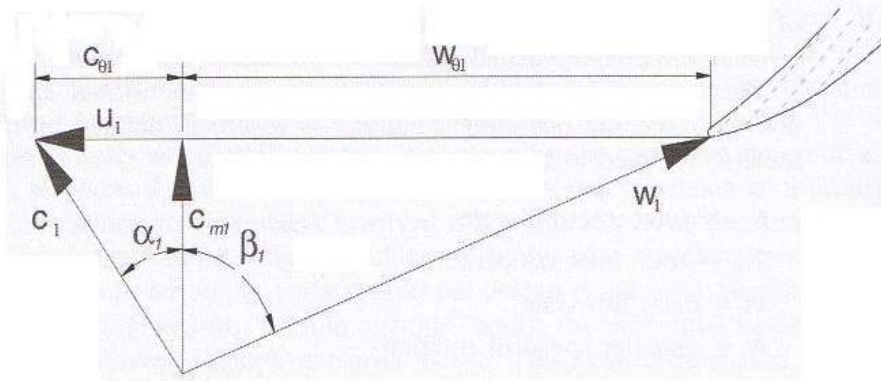


Figure 4-4. Impeller inlet velocity triangle [21]

In Figure 4-4, a variety of possible inlet states is shown for a flow being conveyed to the eye of the impeller with a meridional velocity C_{m1} .

The vector triangle is written according to the fundamental principle of vector addition:

relative velocity+wheel speed=absolute velocity.

For an inlet flow where $C_{u1} = 0$, the relative flow angle is set by the inlet meridional velocity (and hence by the mass flow and density variations) and the local wheel speed U_1 . The use of preswirl (non-zero C_{u1}) is comparatively rare for a pump but not at all rare for a compressor.

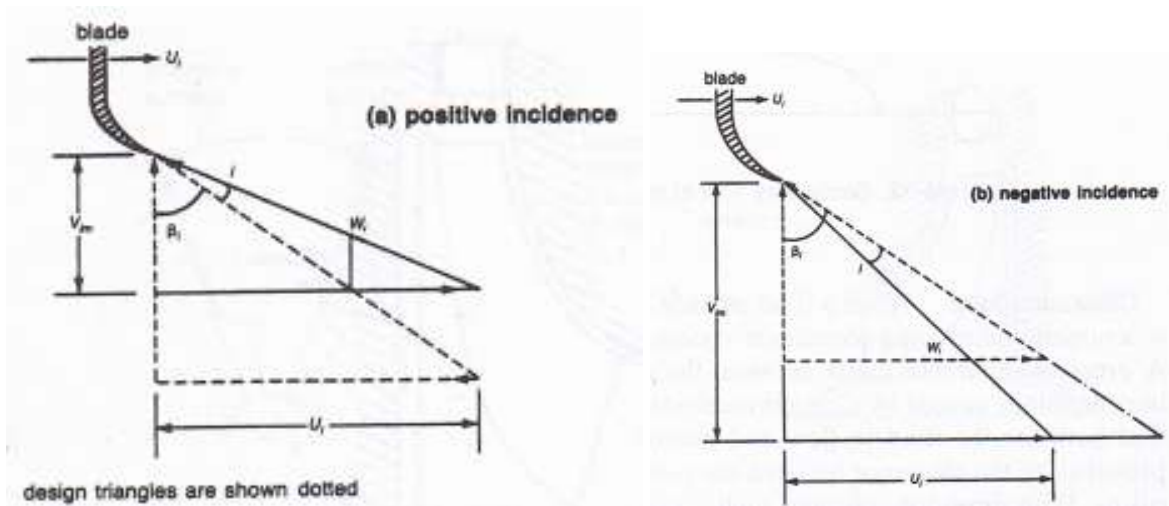


Figure 4-5. Inlet velocity triangles at nonzero incidents [23]

The resulting relative flow angle is β_1 . If this angle is equal to the blade angle, then the flow approaches the blading dead-on. Under many operating conditions, however, the angle will be either

larger or lower than the blade angle, and hence the blading will be subject to an angle of incidence, which is defined in Figure 4-5 as:

$$i_1 = \beta_{B1} - \beta_1 ,$$

where β_{B1} is the inlet blade angle and $V_m = C_m$ (Figure 4-5).

The inlet flow state can be modified substantially by the use of preswirl, either with or against the direction of rotation.

This has the effect of modifying the absolute flow angle α_1 and the tangential component of velocity C_{u1} .

Now that the impeller inlet has been examined with the help of velocity triangles, next step is to apply calculations to the impeller inlet.

The design of a radial compressor starts from the impeller inlet, also known as *inducer*.

Furthermore the tip relative Mach number M_{1t} is limited to prevent any malfunction of the compressor such as choking. Choking occurs when the velocity of fluid in a passage reaches the speed of sound at any cross section and air ceases to flow.

The total pressure p_{00} and temperature T_{00} at the impeller inlet and the mass flow rate $\dot{m}$ are necessary to determine the thermodynamic state of the gas at the impeller inlet.

Moreover, the rotational speed of the impeller N and the hub radius r_{1h} are known and can be used to determine additional inlet geometry.

Then, the inlet velocity triangle can be determined by an initial estimation of the meridional flow velocity C_{m1} and the amount of preswirl.

In addition, an empirical blockage factor B_1 is introduced as a measure of the thickness of the boundary layers at the impeller inlet.

Aerodynamic blockage is directly related to the displacement thickness concept of boundary layers and represents the fraction by which a flow passage is effectively *blocked* by the presence of the low-momentum boundary layer regions [24].

Furthermore, it has been claimed [25] that for an ideal simple axial inlet the blockage factor varies from 0.02 to 0.04, while the highest values (up to 0.14) are used for radial inlet guide vane systems.

With these parameters known, the calculations of the inlet state can be made. As stated before, the inducer design should be optimised for a limited Mach number. In order to accomplish this, the starting guess of the meridional flow velocity should be iterated while the tip relative Mach number stays below the limited value. Another approach is to select C_{m1} in a certain range of values, in order to find the M_{1t} below the limited within this interval. In this report an algorithm has been used to perform the optimization calculations. The chosen algorithm is given in the following chapter.

Equations 4.7 to 4.18 summarise the calculations required to determine the inlet state of the impeller.

$$C_{u1} = C_{m1} \tan \alpha_1 \quad (4.7)$$

$$C_1 = \sqrt{C_{m1}^2 + C_{u1}^2} \quad (4.8)$$

$$T_1 = T_{00} - \frac{C_1^2}{2c_p} \quad (4.9)$$

$$M_1 = \frac{C_1}{\sqrt{\gamma RT_1}} \quad (4.10)$$

$$p_1 = p_{00} \left(\frac{T_1}{T_{00}} \right)^{\frac{\gamma}{\gamma-1}} \quad (4.11)$$

$$\rho_1 = \frac{p_1}{RT_1} \quad (4.12)$$

$$A_{f1} = \frac{\dot{m}}{\rho_1 C_{m1} (1 - B_1)} \quad (4.13)$$

$$r_{1t} = \sqrt{\frac{A_{f1}}{\pi} + r_{1h}^2} \quad (4.14)$$

$$U_{1t} = \frac{2\pi N r_{1t}}{60} \quad (4.15)$$

$$W_{1t} = \sqrt{C_{m1}^2 + (U_{1t} - C_{u1})^2} \quad (4.16)$$

$$M_{1t} = \frac{W_{1t}}{\sqrt{\gamma RT_1}} \quad (4.17)$$

$$\beta_1 = \arctan \left(\frac{U_{1t} - C_{u1}}{C_{m1}} \right) \quad (4.18)$$

4.5 Impeller exit design calculations

It's important to examine the exit station before calculations can be applied.

The compressor rotor exit velocity triangle is shown in Figure 4-6.

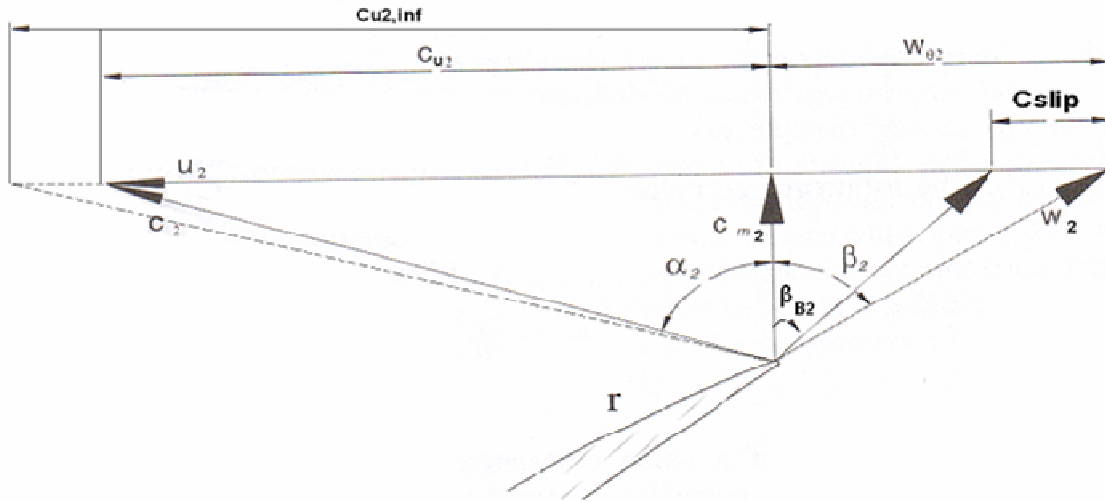


Figure 4-6. Impeller exit velocity triangle [21]

The meridional velocity component C_{m2} is governed by the relationship of conservation of mass, and the absolute flow angle is a result of the relative flow angle and the wheel speed according to the velocity triangle relations. In the past, high speed impellers were traditionally designed with radial blades at exit ($\beta_{B2} = 0$), but it has been stated that [26], for a modern impeller design the blades at exit will typically have a backswept angle of -30° to -40° . Moreover, a backswept angle of -45° has been stated [54]. This angle is negative according to the sign convention. In practice, the exit flow angle does not precisely follow the blading, but differs by an appreciable amount. This difference, in terms of tangential velocity, is known as slip velocity C_{slip} :

$$C_{slip} = C_{u2,\infty} - C_{u2}, \quad (4.19)$$

where $C_{u2,\infty}$ is the tangential component of the absolute velocity which would exist if the flow exactly followed the blades.

The slip velocity is an alternative to the angle of deviation, which is used to express the difference between the blade and flow angles at the exit of an axial stage. The slip velocity rather than the angle of deviation is most often (but not exclusively) used in radial stages. Based on the slip velocity, a slip factor can be derived; it is defined as follows:

$$\sigma = 1 - \frac{C_{slip}}{U_2}. \quad (4.20)$$

An alternative definition frequently used in Europe is:

$$\sigma' = \frac{C_{u2}}{C_{u2,\infty}}. \quad (4.21)$$

The magnitudes are similar but the meaning is different and attention must be paid in their usage. For either a radial or a backswept blade, the resulting absolute flow angle α_2 will be very large, typically in the range of 50° to 80° and most commonly between 65° and 75° [26].

This velocity triangle is important not only in determining the level of work input, or pressure rise, but also in understanding the variation in pressure rise with changes in mass flow. Many parameters must be specified before a precise calculation of such results can be made.

The impeller exit diameter, depth, flow state and velocity triangle at the impeller exit can be calculated, starting from the impeller inlet designed.

The total pressure and temperature at the impeller inlet and the mass flow rate are again input quantities, as well as the desired stage pressure ratio pr_{stage} . The meridional flow velocity at the inlet, the amount of preswirl and the rotational speed come from the inducer design.

At this point, a choice must be made on the desired shape of the velocity triangles presented above.

This shape is a function of the swirl ratio λ_2 and the exit blade angle β_{B2} . As a guideline to the value of the swirl parameter, the following relation between λ_2 and the specific speed N_s has been claimed [25]:

$$\lambda_2 = 6.5 - 0.025 N_s. \quad (4.22)$$

The specific speed has been defined and calculated in Appendix D [27].

For the calculations, some other empirical parameters must be assumed, such as the rotor efficiency η_{rotor} and the slip factor σ . The following equation for η_{rotor} has been assumed [26]:

$$\eta_{rotor} = \frac{\left(\frac{p_{02}}{p_{01}} \right)^{\frac{\gamma-1}{\gamma}} - 1}{\left(\frac{T_{02}}{T_{01}} \right) - 1}, \quad (4.23)$$

which is the relation defined for compressible flow.

Wiesner's relation has been used to determine the slip factor from the backsweep angle β_{B2} and the number of impeller blades Z_r (both selected by the designer):

As an initial guess, a value of the stage efficiency η_{stage} must be given: it should then be iterated until

the computed value of $\frac{p_{s2}}{p_{00}}$ reaches the desired stage pressure ratio.

Furthermore, it should be noted that the stage efficiency as calculated in Equation 4.42 is the total-to-static efficiency.

The calculations have been performed with Equations 4.22 to 4.44 except 4.23:

$$\lambda_2 = 6.5 - 0.025 N_s \quad (4.24)$$

$$\sigma = 1 - \frac{\sqrt{\cos \beta_{B2}}}{Z_r^{0.7}} \quad (4.25)$$

$$\mu_2 = \frac{\sigma \lambda_2}{\lambda_2 - \tan \beta_{B2}} \quad (4.26)$$

$$\Delta h_{0,is} = c_p T_{00} \left(pr^{\frac{\gamma-1}{\gamma}} \right) \quad (4.27)$$

$$W_x = \frac{\Delta h_{0,is}}{\eta_{stage}} \quad (4.28)$$

$$T_{02} = T_{00} + \frac{W_x}{c_p} \quad (4.29)$$

$$U_2 = \sqrt{\frac{U_1 C_{u1} + W_x}{\mu_2}} \quad (4.30)$$

$$D_2 = \frac{60 U_2}{\pi N} \quad (4.31)$$

$$C_{u2} = \mu_2 U_2 \quad (4.32)$$

$$C_{m2} = \frac{C_{u2}}{\lambda_2} \quad (4.33)$$

$$C_2 = \sqrt{C_{u2}^2 + C_{m2}^2} \quad (4.34)$$

$$T_{s2} = T_{02} - \frac{C_2^2}{2c_p} \quad (4.35)$$

$$p_{02} = p_{00} \left(1 + \frac{\eta_{rotor} W_x}{c_p T_{00}} \right)^{\frac{\gamma}{\gamma-1}} \quad (4.36)$$

$$p_{s2} = p_{02} \left(\frac{T_{s2}}{T_{02}} \right)^{\frac{\gamma}{\gamma-1}} \quad (4.37)$$

$$\rho_2 = \frac{p_{s2}}{RT_{s2}} \quad (4.38)$$

$$A_{f2} = \frac{\dot{m}}{\rho_2 C_{m2}} \quad (4.39)$$

$$b_2 = \frac{A_2}{\pi D_2} \quad (4.40)$$

$$pr_{ts} = \frac{p_{s2}}{p_{00}} \quad (4.41)$$

$$\eta_{ts,stage} = \frac{(pr_{ts})^{\frac{\gamma-1}{\gamma}} - 1}{\frac{T_{02}}{T_{00}} - 1} \quad (4.42)$$

$$W_2 = \sqrt{(U_2 - C_{u2})^2 + C_{m2}^2} \quad (4.43)$$

$$DR = W_{1r} / W_2 \quad (4.44)$$

The desired stage pressure ratio p_r appears in Equation 4.27, while the estimated stage efficiency η_{stage} appears in Equation 4.28. The assumed rotor efficiency η_{rotor} appears in Equation 4.36. At the end of the calculation the following must be accomplished:

$$pr_{ts} = pr ,$$

$$\eta_{ts,stage} = \eta_{stage} .$$

Chapter 5. Problem statement and results

This chapter starts with choosing a suitable optimization algorithm for the assignment.

Moreover the design parameters followed by design variables, objective function and constraints of the radial compressor will be presented in this sections.

Further on the design space and optimality of the compressor and optimization algorithm respectively will be presented.

In the following section the inlet design results will be used as a starting point to avoid M_{1t} higher than one.

Finally, optimization Inlet and outlet results and effects of impeller outlet on performance will be presented.

5.1. Chosen optimization Algorithm

In this section, the advantages and disadvantages of algorithms presented in this report are given.

Furthermore, the advantages and disadvantages have been given to choose one algorithm based on the criteria's given at the end of this section.

Then the chosen algorithm will be used to perform optimization calculation for the compressor design.

Algorithm	Hooke and Jeeves Method (pattern search)
Advantage	1. They are convenient because they do not require the determination of the derivatives of the objective function or the constraints. This implies that they are generally applicable because they can be used for functions whose differentiation is difficult and even for functions whose derivatives are discontinuous.
Disadvantage	1. No optimality conditions can be applied to them, solutions obtained by these methods cannot be considered optimum (stuck in local optimum). 2. No convergence criteria can be established and hence the search may be terminated prematurely.

Algorithm	The Simplex method
Advantage	1. Very effective for solving linear objective function.
Disadvantage	1. Not capable for solving nonlinear objective function.

Algorithm	The Two-Phase Method
Advantage	1. Very effective for solving linear objective function. 2. Capable for solving problems with inequality constraints of the form $\geq$ or equality constraints ($=$).
Disadvantage	1. Not capable for solving nonlinear objective function.

Algorithm	The Dual Approach
Advantage	1. Very effective for solving linear objective function. 2. Conversion of a minimization problem into a maximization problem and id.
Disadvantage	1. Not capable for solving nonlinear objective function

Algorithm	Lagrange's Method
Advantage	1. High overall computational efficiency, which leads to conversion in fewer steps.
Disadvantage	1. The amount of computational effort. 2. This method also requires the evaluation of the derivatives of the objective function.

Algorithm	Quadratic Programming
Advantage	1. No starting point needed.
Disadvantage	1. This method also requires the evaluation of the derivatives of the objective function.

Algorithm	Sequential Quadratic Programming
Advantage	1. Capable for solving problems dealing with nonlinear constraint optimization. 2. No starting point needed. 3. Fewer function and gradient evaluation needed.
Disadvantage	1. This method also requires the evaluation of the derivatives of the objective function.

Algorithm	GA
Advantage	1. GA's begins a search from a population of points instead of a single point. Consequently, searches generate a population of points. As a result, many points are considered avoiding local optimum. 2. They don't use the information from objective function derivatives, but only the values of the objective function. 3. They use probabilistic transition rules to select improved population points. 4. No starting point is needed.
Disadvantage	1. They do not strictly guarantee the optimality of the solution. 2. They do not guarantee that the solution is feasible because the solution is obtained by a statistical method. 3. Computational time.

Algorithm	SA
Advantage	1. Random search method which avoids to get stuck in local optimums and therefore leads to global minimum or maximum.
Disadvantage	1. The major difficulty (art) in implementation of the algorithm is that there is no obvious analogy for the temperature T with respect to a free parameter in the problem. 2. The avoidance of entrainment in local minima (quenching) is dependent on the "annealing schedule", the choice of initial temperature, the number of iterations performed at each temperature and the temperature decrease at each step as cooling proceeds. 3. A starting point is needed. 4. Computational time.

The objective function has been analyzed and the following have been noticed:

- The function is nonlinear;
- The derivatives of the objective function can't be used, because the objective variable disappears when the derivatives are calculated.

A suitable algorithm has been chosen after reviewing the advantage(s) and disadvantage(s) of some optimization algorithms concerning the objective functions.

Further on, after reviewing some articles [48], [49] and [50] concerning optimization in engineering design, it has been concluded that a GA will be used for the optimization of the design.

5.2. Operating conditions

Figure 5-1 illustrates the meridional view of a radial compressor. The figure shows the impeller inlet (station 1) and exit (station 2).

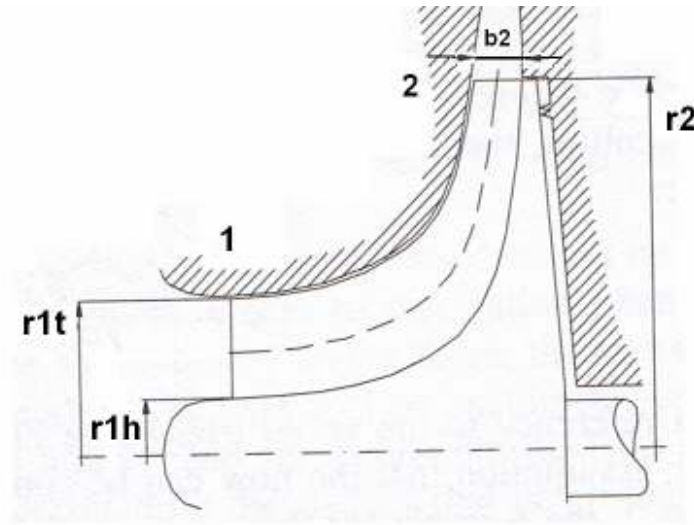


Figure 5-1. Meridional view of a radial compressor [21]

The microturbine does not contain any diffuser. As a consequence, the maximum pressure recovery will be attained in the rotor.

Furthermore, the total pressure p_{00} and temperature T_{00} and the mass flow rate $\dot{m}$ are necessary to determine the thermodynamic state of the gas at the inlet. Moreover, the rotational speed of the impeller N is known and can be used to determine additional inlet geometry. Besides, the inlet velocity triangle can be determined by the amount of preswirl and an empirical blockage factor B_1 is introduced as a measure of the thickness of the boundary layers at the inlet. The pressure ratio pr and the rotor efficiency η_{rotor} are input parameters necessary to determine the impeller exit.

A rotational speed of $600,000 \text{ rpm}$ have been chosen and it has been confirmed [28] that, some authors have verified that the efficiency peaks in the range between $95 \text{ rpm} \cdot \text{ft}^{3/4} / \text{s}^{1/2}$ and $120 \text{ rpm} \cdot \text{ft}^{3/4} / \text{s}^{1/2}$ (between $7 \text{ rpm} (\text{m}^3 / \text{s})^{0.5} / (\text{J/kg})^{0.75}$ and $9 \text{ rpm} (\text{m}^3 / \text{s})^{0.5} / (\text{J/kg})^{0.75}$ in the International System of Units). It can be noticed that the calculated specific speed ($N_s \approx 126$) for $N=600,000 \text{ rpm}$ is slightly higher than the right-hand limit of the range (see Appendix D).

Figures 5-2 and 5-3 illustrate the effect of the meridional flow velocity on the tip relative Mach number.

The design parameters are summarized in Table V-1.

T_{00} [K]	298
p_{00} [Pa]	101,325
$\dot{m}$ [kg/s]	0.005
N [rpm]	600,000
α_1 [°]	0
B_l [-]	0.02-0.04
pr [-]	2
η_{rotor} [-] starting value	0.92

Table V-1. Operating conditions

The optimization calculations for the impeller inlet and exit will be performed using a GA available in Matlab. Moreover, the impeller inlet has been calculated apart from the exit and considered as a pre design of the inlet to avoid M_{1t} higher than one.

Then the impeller inlet and impeller exit have been calculated at once, which is not the case with the procedure presented in [25]. The procedure described there performs calculations on the impeller inlet apart from the impeller exit only. Furthermore, the best results concerning η_{stage} will be chosen.

5.3. Design variables

After analysing the inlet calculation formula's and presenting the operation conditions, it can be concluded that C_{m1} and r_{1h} are design variables.

The inlet velocity triangle can be determined by an initial estimation of the meridional flow velocity C_{m1} and the amount of preswirl.

According to section 4.4, the tip relative Mach number M_{1t} is limited to prevent any malfunction of the compressor and can be considered a constraint.

Furthermore, in section 5.5 the objective will be presented, concerning the maximization of stage efficiency $\eta_{ts,stage}$.

Finally, according to section 4.5, the backsweep angle β_{B2} and the number of impeller blades Z_r must be selected by the designer, but also can be considered as design variables.

The design variables are then summarized in Table V-2.

C_{m1}	Z_r
r_{1h}	β_{B2}

Table V-2. Design variables

5.4. Design objective

The design objective for this compressor is to find the maximum stage efficiency $\eta_{ts,stage}$ after satisfying the constraints specified in the following section.

With a rotational speed of $N=600,000 \text{ rpm}$ and a C_{m1} within the range of 0 m/s to 300 m/s [22], the optimum M_{1t} has been found for different values of the hub radius r_{1h} .

According to some publications ([29], [30], [31], [32], [28]), the hub radii chosen are summarized in Table V-3.

$r_{1h} \text{ [mm]}$	1	2	3	4	5	6	9	12	15	18	21	24
-----------------------	---	---	---	---	---	---	---	----	----	----	----	----

Table V-3. Hub radii chosen in the design

These values have been used as a reference for the optimization calculation, but there is a possibility that the optimized hub radii could deviate from the reference values.

Furthermore these values has been used to determine the final range of hub radii without exceeding the limited value of M_{1t}

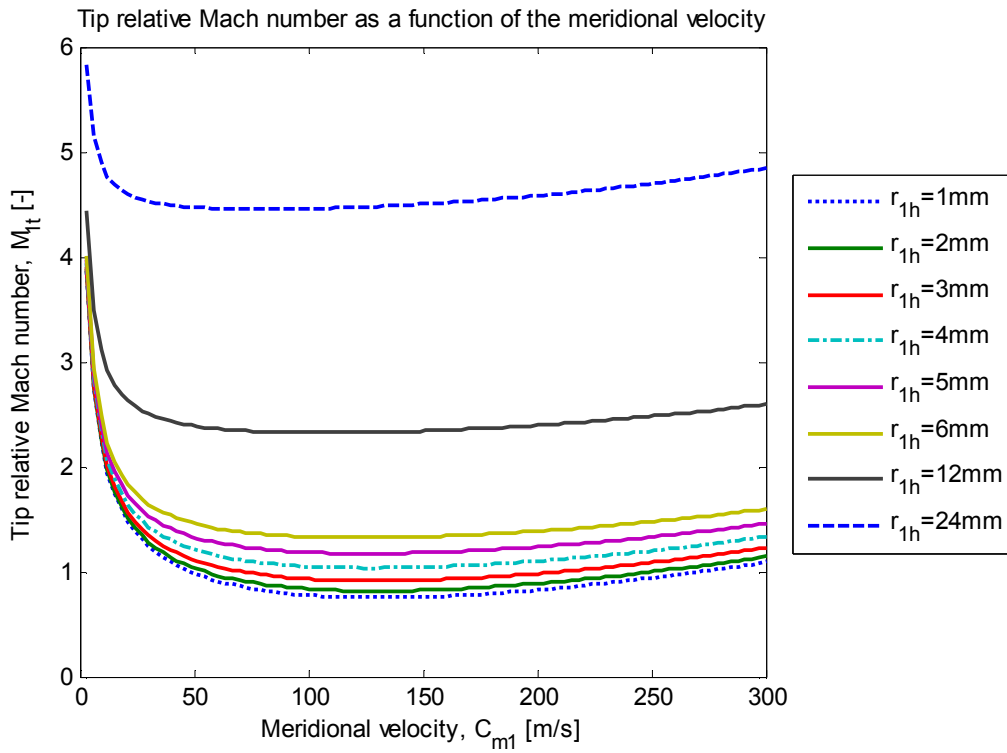


Figure 5-2. Tip relative Mach number as function of meridional velocity for $N=600,000\text{ rpm}$ and $B_l=0.02$

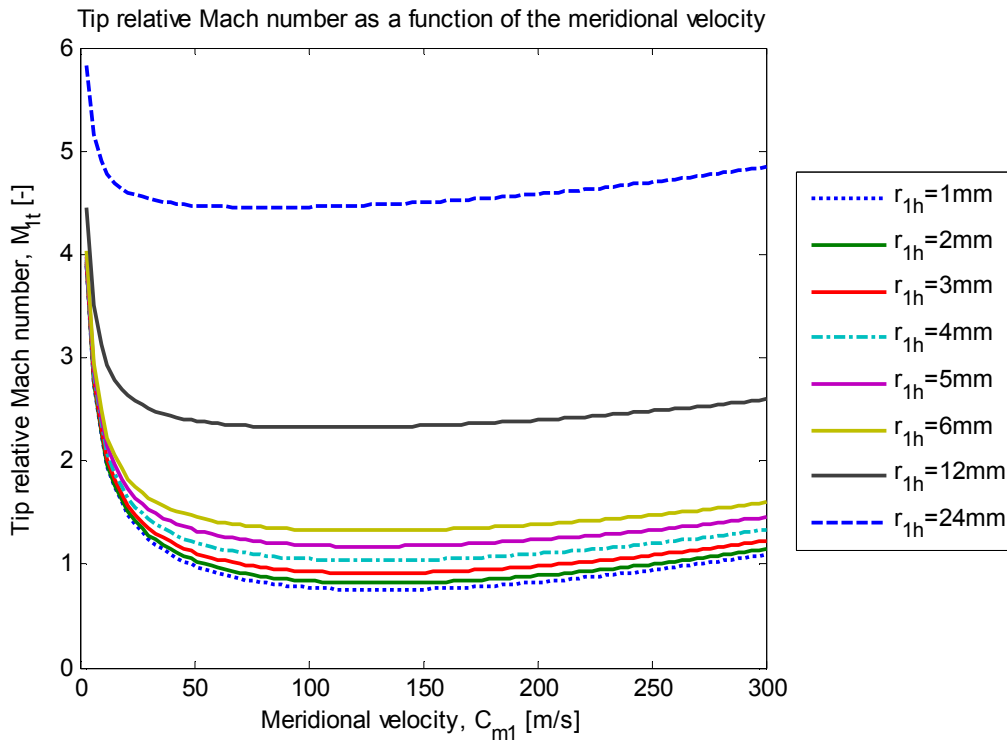


Figure 5-3. Tip relative Mach number as function of meridional velocity for $N=600,000\text{ rpm}$ and $B_l=0.03$

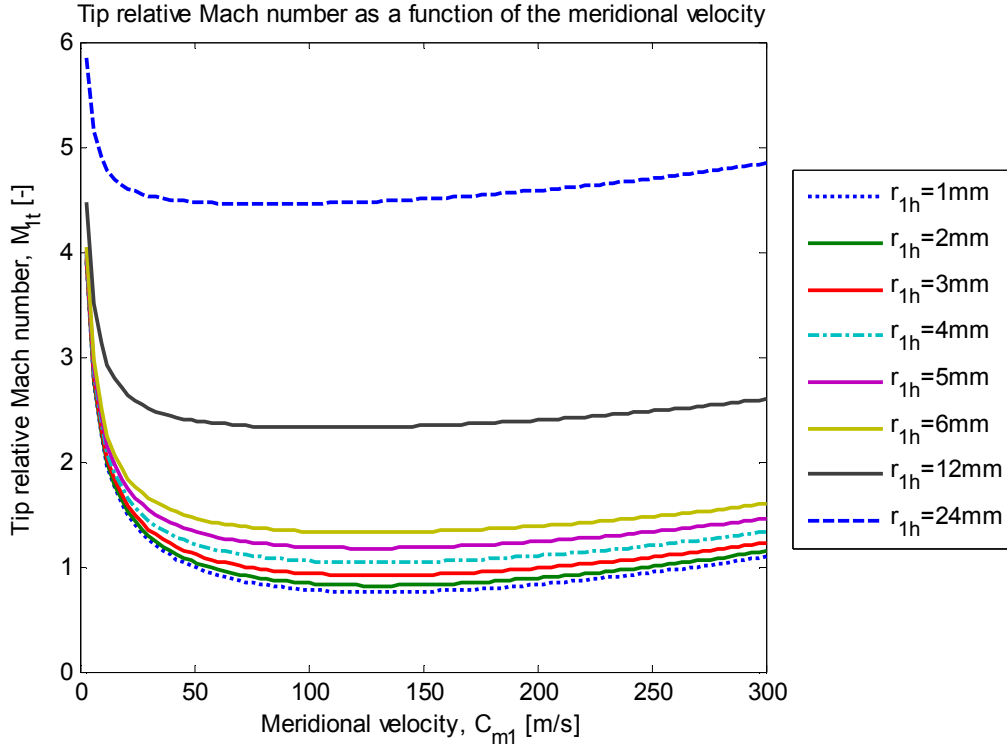


Figure 5-4. Tip relative Mach number as function of meridional velocity for $N=600,000 \text{ rpm}$ and $B_I=0.04$

For clearness of the graphs only values of hub radius between 1 and 6 mm and then 12 mm and 24 mm have been depicted.

In Figures 5-2 to 5-4 it can be noticed that, with increasing C_{m1} , M_{It} decreases until C_{m1} reaches a minimum value and then increases.

The trend of these figures can be explained recalling Figure 5-1 and Equations 4.7 to 4.18. If the hub radius, the total pressure and temperature at the impeller inlet, the rotational speed, the absolute inlet flow angle, the mass flow rate, the blockage factor and the flow properties are constant, by changing C_{m1} the A_{f1} also changes. The two are related through the continuity equation (see Equation 4.13). If the C_{m1} increases (like in these calculations), the A_{f1} must decrease, in order to keep the mass flow rate constant. As a consequence, the impeller tip radius decreases according to Equation 4.14. This will lead to a decrease in the blade speed U_{It} (see Equation 4.15). The W_{It} and M_{It} are dependent on both U_{It} and C_{m1} (see Equations 4.15 and 4.16). As a consequence, M_{It} will decrease until the decreasing contribution of U_{It} larger than the increasing contribution of C_{m1} .

5.5. Design Constraints

In section 1.4, optimization constraints have been described as numerical values of identified conditions that must be satisfied to achieve a feasible solution to a given problem.

The pr_{ts} must be iterated until it reaches the pressure ratio pr . In this case a pressure ratio of 2 must be achieved. Moreover the slip factor σ must be iterated between 0.8 and 0.9 [33].

According to the operating conditions the absolute flow inlet angle is 0. Then, $C_{u1} = 0$.

This requires that the absolute speed of air entering the compressor should vary from impeller root to tip. The tip velocity U_{1t} then becomes a maximum at the tip of the impeller. In addition it is quite possible that W_{1t} becomes so large that the inlet Mach number goes beyond unity. Then, the compressor inlet will be choked. Even when this Mach number is less than unity, it is possible that the local Mach number can become large at some point within the passage and choke the flow. In order that no choking may occur, it is essential to maintain the flow Mach number at the inlet tip below 0.9 [47].

Furthermore, $\dot{m}$ and α_1 can also be considered as constraints, because all the calculation will be based on the same values.

Finally the last constraint concerning this design has been the diffusion ratio W_{1t}/W_2 .

It has been reported [34] that the diffusion ratio is a critical design parameter and it has been stated [35] that a diffusion ratio of 2.0 is typically suggested as a surge criterion. Also a diffusion ratio in the range of 1.9 to 2.0 as surge limit has been claimed [36].

Nonetheless, impellers with a diffusion ratio above 2.0 have been described.

Similarly, it has been claimed [38] that values between 0.45 and 0.60 are normally chosen for the simple relative velocity ratio W_2/W_{1t} , which they state to be an adequate design parameter. The reciprocal of that ratio gives then a diffusion ratio in the range of $1.67 \div 2.22$.

The constraints are summarized in Table V-4.

pr_{ts} [-]	2
σ [-]	0.8 - 0.9
$\dot{m}$ [kg/s]	0.005
α_1 [°]	0
M_{1t} [-]	≤ 0.9
W_{1t}/W_2 [-]	1.67 – 2.22

Table V-4. Design Constraints

5.6. Design Space

The design space for C_{m1} is between 0 and $300 \frac{m}{s}$ [22] as presented in section 5.4 and the estimated stage efficiency η_{stage} is from 0.1 to 1.

Moreover, the blades at exit are usually backswept by -30° to -40° [26], -45° [54], so the calculations have been carried out with the values summarized in Table V-5.

$\beta_{B2} [^\circ]$	-45	-40	-30	-20	-10	0
-----------------------	-----	-----	-----	-----	-----	---

Table V-5. Exit blade angles chosen in the design

According to some publications ([29], [30], [31], [32] and [28]), the hub radii design space is between 1 mm and 24 mm.

Moreover, although from literature the blade number usually appears to be between 26 and 30, [30] developed a MEMS-based turbocharger whose compressor has a pressure ratio of 2, a rotational speed of 1,000,000 rpm and 10 blades. Besides, [31] designed a microscale, high-speed compressor impeller with a design pressure ratio of 3, a shaft which must spin at 800,000 rpm and 12 blades (six fullblades and six splitters). Furthermore, the microturbine developed by [46] has a compressor with an extrapolated pressure ratio of 1.63 at 250,000 rpm, and 20 blades.

As a consequence, the calculations have been performed with the values of blade number summarized in Table V-6.

Z_r	5	10	15	20	25	30
-------	---	----	----	----	----	----

Table V-6. Blade numbers reference values

These values have been presented as a reference for the optimization calculation, but there is a possibility that the optimized blade number could deviate from the reference values.

5.7. Design Optimality

Although the optimization algorithms used for this design were only capable to minimize objective functions, a procedure described in section 1.6 has been used to convert minimization into maximization.

Moreover, GAs are more likely capable to find global optimums than a local optimum.

This means that more than one solution can be obtained and also the possibility to get the local optimum is available.

5.8. Inlet Design

Although calculations have been performed with the inlet and outlet combined, the inlet design results will be presented apart from exit results. These results will be considered as a pre design of the inlet to avoid M_{1t} higher than 0.9 and to determine the final r_{1h} range. M_{1t} higher than 0.9 will lead to shock waves on the blade tips and also stresses on the blades will grow.

The tip relative Mach number has been calculated for each impeller hub radius (see Table V-3). Moreover, as described in previous section, more than one solution can be obtained when GAs are

used. Thus the calculations concerning GAs have been done 5 times and the best from these solutions has been taken.

Finally, Tables V-7 to V-12 summarizes the outcomes for $N=600,000$ rpm.

r_{1h} [m]	0.001	0.002	0.003	0.004	0.005	0.006
D_{1h} [m]	0.002	0.004	0.006	0.008	0.01	0.012
$(M_{1t})_{\min}$ [-]	0.7507	0.8156	0.9137	1.0354	1.1735	1.3226
r_{1t} [m]	0.0035	0.0039	0.0045	0.0052	0.0061	0.0069
D_{1t} [m]	0.0070	0.0078	0.0090	0.0104	0.0122	0.0138
r_{1t}/r_{1h} [-]	3.5000	1.9500	1.5000	1.3000	1.2200	1.1500
β_I [°]	59.3130	62.1499	65.6126	68.9170	71.7935	74.2059
A_{fl} [m ²]	3.5432e-5	3.5591e-5	3.5886e-5	3.6273e-5	3.6757e-5	3.7331e-5
U_{1t} [m/s]	220.1667	246.0006	283.9477	329.7680	380.6394	434.7796
W_{1t} [m/s]	256.0174	278.2267	311.7650	353.4262	400.6997	451.8384
$C_{ml}=C_l$ [m/s]	130.6580	129.9762	128.7290	127.1342	125.1955	122.9822

Table V-7. Inducer design results for $N=600,000$ rpm, $B_I=0.02$ and $r_{1h}=1, 2, 3, 4, 5, 6$ mm

r_{1h} [m]	0.009	0.012	0.015	0.018	0.021	0.024
D_{1h} [m]	0.018	0.024	0.030	0.036	0.042	0.048
$(M_{1t})_{\min}$ [-]	1.8080	2.3216	2.8476	3.3799	3.9158	4.4539
r_{1t} [m]	0.0097	0.0125	0.0155	0.0184	0.0214	0.0244
D_{1t} [m]	0.0194	0.0250	0.0310	0.0368	0.0428	0.0488
r_{1t}/r_{1h} [-]	1.0777	1.0416	1.0333	1.0222	1.0190	1.0166
β_I [°]	79.2556	82.2539	84.1549	85.4298	86.3280	80.8709
A_{fl} [m ²]	3.9498e-5	4.2156e-5	4.5118e-5	4.8252e-5	5.1522e-5	5.4734e-5
U_{1t} [m/s]	607.7907	788.3297	972.0910	1.1575e+003	1.3438e+003	1.5306e+003
W_{1t} [m/s]	618.6363	795.5894	977.1715	1.1612e+003	1.3465e+003	1.5327e+003
$C_{ml}=C_l$ [m/s]	115.3315	107.2321	99.5151	92.5219	86.2394	80.8709

Table V-8. Inducer design results for $N=600,000$ rpm, $B_I=0.02$ and $r_{1h}=9, 12, 15, 18, 21, 24$ mm

r_{1h} [m]	0.001	0.002	0.003	0.004	0.005	0.006
D_{1h} [m]	0.002	0.004	0.006	0.008	0.01	0.012
$(M_{1t})_{\min}$ [-]	0.7533	0.8181	0.9159	1.0374	1.1752	1.3242
r_{1t} [m]	0.0035	0.0039	0.0045	0.0053	0.0061	0.0069
D_{1t} [m]	0.0070	0.0078	0.009	0.0106	0.0121	0.0139
r_{1t}/r_{1h} [-]	3.5164	1.9637	1.50961	1.31423	1.21312	1.15437
β_I [°]	59.3237	62.1694	65.5922	68.891	71.7661	74.1734
A_{fl} [m ²]	3.5704e-5	3.5889e-5	3.6160e-5	3.6553e-5	3.7044e-5	3.7614e-5
U_{1t} [m/s]	220.9399	246.7597	284.5541	330.3017	381.1133	435.1887
W_{1t} [m/s]	256.8879	279.0351	312.4814	354.0602	401.2618	452.3361
$C_{ml}=C_l$ [m/s]	131.0609	130.2959	129.1262	127.5125	125.5536	123.3642

Table V-9. Inducer design results for $N=600,000$ rpm, $B_I=0.03$ and $r_{1h}=1, 2, 3, 4, 5, 6$ mm

r_{1h} [m]	0.009	0.012	0.015	0.018	0.021	0.024
D_{1h} [m]	0.018	0.024	0.030	0.036	0.042	0.048
$(M_{1t})_{\min}$ [-]	1.8092	2.3226	2.8485	3.3807	3.9165	4.4546
r_{1t} [m]	0.0097	0.0126	0.0155	0.0184	0.0214	0.0244
D_{1t} [m]	0.0194	0.0252	0.0310	0.0368	0.0428	0.0488
r_{1t}/r_{1h} [-]	1.0754	1.0459	1.0317	1.0236	1.0185	1.0151
β_I [°]	79.2306	82.231	84.1387	85.4165	86.3145	86.9676
A_{fl} [m ²]	3.9803e-5	4.2464e-5	4.5457e-5	4.8610e-5	5.1869e-5	5.5155e-5
U_{1t} [m/s]	608.106	788.5753	972.3105	1.1577e3	1.3439e3	1.5308e3
W_{1t} [m/s]	619.008	795.8805	977.4205	1.1614e3	1.3467e3	1.5329e3
$C_{ml}=C_l$ [m/s]	115.666	107.5861	99.8153	92.8077	86.5663	81.0932

Table V-10. Inducer design results for $N=600,000$ rpm, $B_I=0.03$ and $r_{1h}=9, 12, 15, 18, 21, 24$ mm

r_{1h} [m]	0.001	0.002	0.003	0.004	0.005	0.006
D_{1h} [m]	0.002	0.004	0.006	0.008	0.01	0.012
$(M_{1t})_{\min}$ [-]	0.7559	0.8205	0.9181	1.0394	1.1770	1.3257
r_{1t} [m]	0.0035	0.0039	0.0045	0.0053	0.0061	0.0069
D_{1t} [m]	0.0070	0.0078	0.0090	0.0106	0.0122	0.0138
r_{1t}/r_{1h} [-]	3.5000	1.9500	1.5000	1.3250	1.2200	1.1500
β_I [°]	59.3420	62.1572	65.5765	68.8634	71.7320	74.1429
A_{fl} [m ²]	3.5987e-5	3.6160e-5	3.6445e-5	3.6837e-5	3.7326e-5	3.7907e-5
U_{1t} [m/s]	221.7429	247.4501	285.1826	330.8407	381.5778	435.6113
W_{1t} [m/s]	257.7728	279.8473	313.2106	354.7040	401.8297	452.8438
$C_{ml}=C_l$ [m/s]	131.4417	130.7018	129.5057	127.9037	125.9583	123.7347

Table V-11. Inducer design results for $N=600,000$ rpm, $B_I=0.04$ and $r_{1h}=1, 2, 3, 4, 5, 6$ mm

r_{lh} [m]	0.009	0.012	0.015	0.018	0.021	0.024
D_{lh} [m]	0.018	0.024	0.030	0.036	0.042	0.048
(M_{lt})_{min} [-]	1.8104	2.3236	2.8494	3.3815	3.9173	4.4552
r_{lt} [m]	0.0097	0.0126	0.0155	0.0184	0.0214	0.0244
D_{lt} [m]	0.0194	0.0252	0.0310	0.0368	0.0428	0.0488
r_{lt}/r_{lh} [-]	1.0777	1.0500	1.0333	1.0222	1.0190	1.0166
β_l [°]	79.2010	82.2104	84.1200	85.4018	86.3009	86.9562
A_{fl} [m²]	4.0099e-5	4.2790e-5	4.5787e-5	4.8963e-5	5.2221e-5	5.5527e-5
U_{lt} [m/s]	608.4124	788.8350	972.5236	1.1579e+003	1.3441e+003	1.5309e+003
W_{lt} [m/s]	619.3813	796.1817	977.6675	1.1616e+003	1.3469e+003	1.5331e+003
C_{ml}=C_l [m/s]	116.0500	107.9109	100.1578	93.1225	86.8991	81.4072

Table V-12. Inducer design results for $N=600,000$ rpm, $B_l=0.04$ and $r_{lh}=9, 12, 15, 18, 21, 24$ mm

Some observations concerning the inlet design (Tables V-7 to V-12) will follow. It can be noticed that the tip relative Mach number increases with increasing impeller hub radius. This trend is due to the fact that if the impeller tip radius is held constant, an increase in the hub radius decreases the inlet flow

area $\left(A_{fl} = (r_{lt}^2 - r_{lh}^2) \pi\right)$. According to the continuity equation $\left(A_{fl} = \frac{\dot{m}}{\rho_l C_{ml} (1 - B_l)}\right)$, this increases

the meridional velocity, because in this design the mass flow rate is fixed.

Anyway, in Tables V-7 to V-12 it can also be noticed that the tip relative Mach numbers are higher than unity starting from $r_{lh}=4mm$ for $N=600,000$ rpm. This implies that for $N=600,000$ rpm only values of impeller hub radius equal to 1, 2 and 3 will be considered in the design process.

Moreover it can be noticed that the highest C_{ml} for $N=600,000$ rpm can be found in Table V-12. That implies that the design space for C_{ml} can be reduced to $0 \div 150$ m/s.

Furthermore reducing the design space [43] will improve the optimization calculation, because the space for exploration has been reduced, which can lead to more accurate and quick solutions.

Figure 5-5 shows M_{lt} as function of C_{m1} for $r_{lh}=1, 2, 3$ mm and $B_l=0.02$.

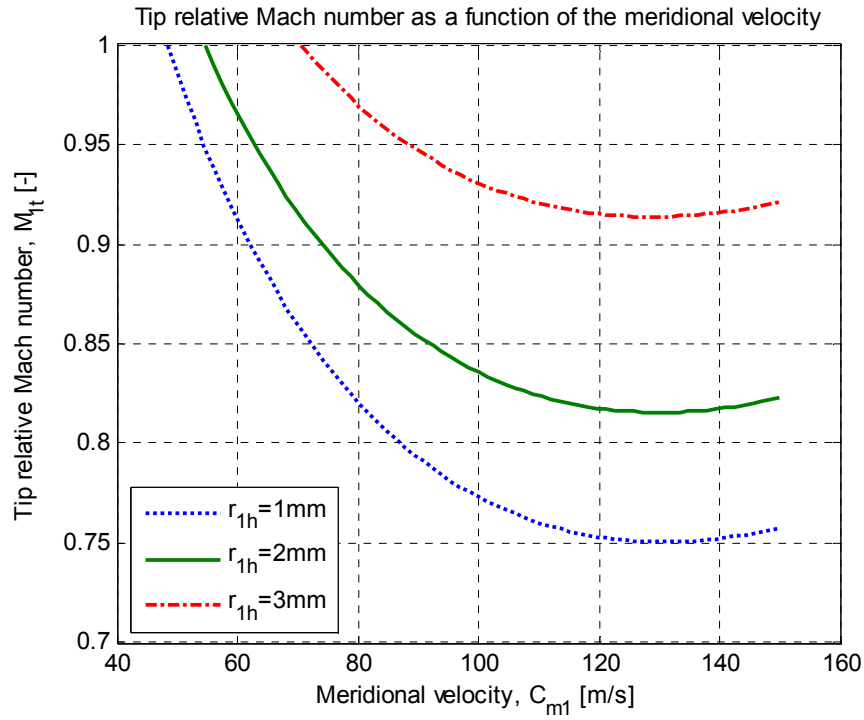


Figure 5-5. Tip relative Mach number as function of meridional velocity for $N=600,000$ rpm, $B_l=0.02$ and $r_{lh}=1, 2, 3$ mm

Figure 5-6 shows M_{lt} as function of C_{m1} for $r_{lh}=1, 2, 3$ mm and $B_l=0.03$.

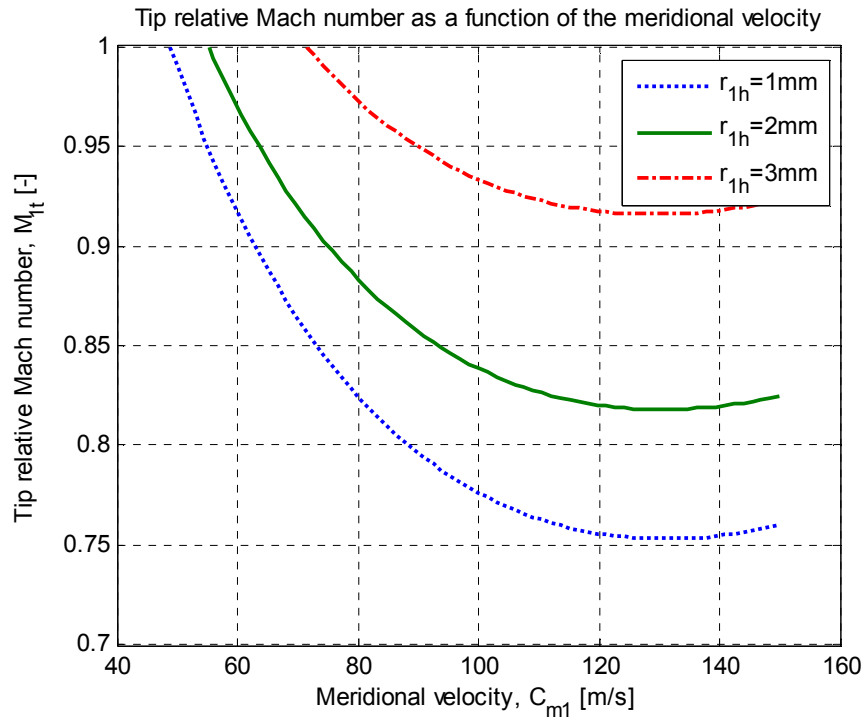


Figure 5-6. Tip relative Mach number as function of meridional velocity for $N=600,000$ rpm, $B_l=0.03$ and $r_{lh}=1, 2, 3$ mm

Figure 5-7 shows M_{1t} as function of C_{m1} for $r_{1h}=1, 2, 3$ mm and $B_I=0.04$.

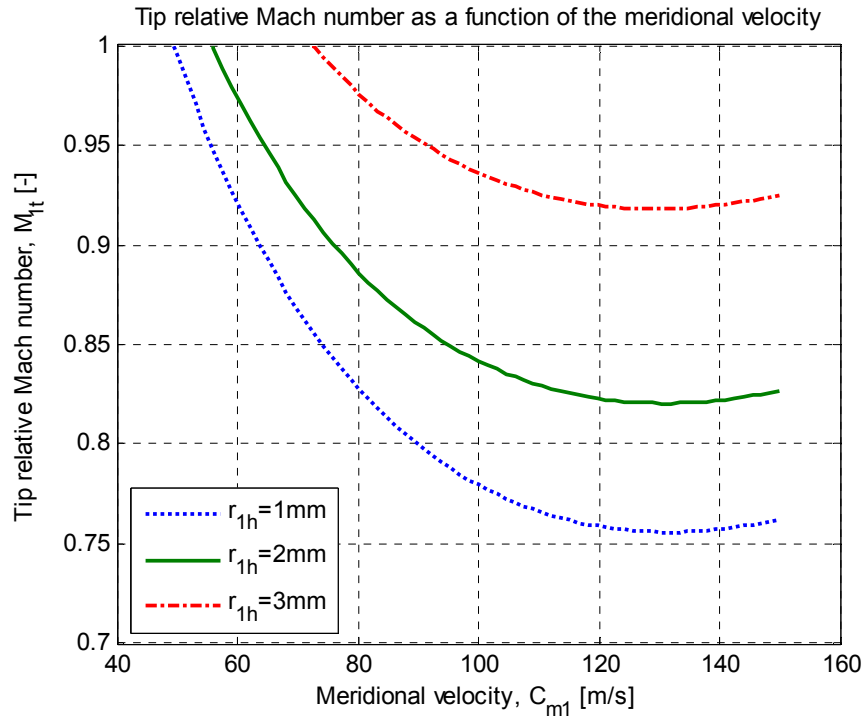


Figure 5-7. Tip relative Mach number as function of meridional velocity for $N=600,000$ rpm, $B_I=0.03$ and $r_{1h}=1, 2, 3$ mm

Table V-13 to V-15 summarizes the impeller inlet geometry at $600,000$ rpm for three impeller hub radii chosen according to the remarks about the tip relative Mach number.

Furthermore, the optimized hub radii will be calculated in the range of $1 \div 3$ mm for $N=600,000$ rpm.

r_{1h} [m]	0.001	0.002	0.003
$(M_{1t})_{\min}$ [-]	0.7507	0.8156	0.9137
r_{1t} [m]	0.0035	0.0039	0.0045
r_{1t}/r_{1h} [-]	3.5000	1.9500	1.5000
β_I [°]	59.3130	62.1499	65.6126
A_n [m ²]	3.5432e-5	3.5591e-5	3.5886e-5
U_{1t} [m/s]	220.1667	246.0006	283.9477
W_{1t} [m/s]	256.0174	278.2267	311.7650
$C_{m1}=C_1$ [m/s]	130.6580	129.9762	128.7290

Table V-13. Inducer design results for $N=600,000$ rpm, $B_I=0.02$ and $r_{1h}=1,2,3$ mm respectively

r_{1h} [m]	0.001	0.002	0.003
$(M_{1t})_{\min}$ [-]	0.7533	0.8181	0.9159
r_{1t} [m]	0.0035	0.0039	0.0045
r_{1t}/r_{1h} [-]	3.5164	1.9637	1.50961
β_1 [°]	59.3237	62.1694	65.5922
U_{1t} [m/s]	220.9399	246.7597	284.5541
W_{1t} [m/s]	256.8879	279.0351	312.4814
$C_{m1}=C_1$ [m/s]	131.0609	130.2959	129.1262

Table V-14. Inducer design results for $N=600,000$ rpm, $B_I=0.03$ and $r_{1h}=1,2,3$ mm respectively

r_{1h} [m]	0.001	0.002	0.003
$(M_{1t})_{\min}$ [-]	0.7559	0.8205	0.9181
r_{1t} [m]	0.0035	0.0039	0.0045
r_{1t}/r_{1h} [-]	3.5000	1.9500	1.5000
β_1 [°]	59.3420	62.1572	65.5765
A_{fl} [m ²]	3.5987e-5	3.6160e-5	3.6445e-5
U_{1t} [m/s]	221.7429	247.4501	285.1826
W_{1t} [m/s]	257.7728	279.8473	313.2106
$C_{m1}=C_1$ [m/s]	131.4417	130.7018	129.5057

Table V-15. Inducer design results for $N=600,000$ rpm, $B_I=0.04$ and $r_{1h}=1,2,3$ mm respectively

5.9. Inlet and outlet optimization results and effects of impeller outlet on performance

The optimization calculations have been performed by taking $\dot{m}$, α_1 , DR, σ , pr_{ts} and M_{1t} as constraints:

- σ and M_{1t} are inequality constraints ($\leq, \geq, <$, and $>$);
- $\dot{m}$, α_1 , DR and pr_{ts} are equality constraints ($=$).

In section 5.5, several DR's has been presented by some author's and a range has been chosen. DR is a very important constraint, because this value is the only connection between the inlet and outlet design. This is very important when the inlet and outlet design must be calculated at the same time, which is the intention in this report.

Furthermore, some (original) values in the DR range has been chosen (see first row of Tables V-16 to V-21) and used to perform optimization calculations. Besides there is a possibility that the original DR value will slightly change when the calculation has been completed, because of the tolerance installed in the optimization algorithm.

Moreover, the influence on the impeller exit of the blade number Z_r and the backsweep angle β_{B2} [33] will be presented in this section. Thus, the effects of these factors on the performance of the impeller will be carefully analyzed, with the help of the results of the optimized design.

DR original [-]	1.67	1.7	1.8	1.9	2.0	2.1	2.2
DR [-]	1.6700	1.6826	1.7998	1.8772	1.9733	2.1000	2.2200
r_{1h} [m]	0.0018	0.0016	0.0024	0.0020	0.0024	0.0024	0.0025
D_{1h} [m]	0.0036	0.0032	0.0048	0.0040	0.0048	0.0048	0.0050
(M_{1t}) [-]	0.8004	0.7816	0.8615	0.8143	0.8450	0.8619	0.8684
r_{1t} [m]	0.0039	0.0035	0.0040	0.0037	0.0039	0.0044	0.0040
D_{1t} [m]	0.0078	0.0070	0.0080	0.0074	0.0078	0.0088	0.0080
r_{1t}/r_{1h} [-]	2.2182	2.3996	1.6439	1.9463	1.7249	1.8195	1.6148
β_1 [°]	64.4031	56.1175	59.1841	57.3580	58.8629	68.6207	59.5167
A_{fl} [m²]	3.8634e-5	3.1940e-5	3.1625e-5	3.1745e-5	3.1873e-5	4.2005e-5	3.1670e-5
U_{1t} [m/s]	246.8465	220.3922	251.1738	232.8060	245.6187	275.0076	254.0777
W_{1t} [m/s]	273.7095	265.4743	292.4645	276.4728	286.9603	295.3299	294.8301
$C_{m1}=C_1$ [m/s]	118.2527	147.9997	149.8240	149.1260	148.3836	107.6596	149.5637
Z_r [-]	10.9554	12.6045	12.4197	13.4940	21.4181	21.8970	19.0446
β_{B2} [°]	-41.7842	-40.2209	-43.3790	-32.8465	-39.3189	-35.5499	-25.1377
η_{stage} [-]	0.5677	0.5581	0.5655	0.5401	0.5366	0.5273	0.5102
b_2 [m]	7.6095e-4	7.4828e-4	7.5784e-4	7.2490e-4	7.2000e-4	7.0771e-4	6.8508e-4
r_2 [m]	0.0067	0.0066	0.0066	0.0066	0.0066	0.0065	0.0065
U_2 [m/s]	417.8592	415.7503	417.4069	412.3216	411.8139	410.4174	408.3196
W_2 [m/s]	163.8979	157.7788	162.4978	147.2809	145.4212	140.6331	132.8061
M_2 [-]	0.7467	0.7633	0.7505	0.7948	0.8011	0.8179	0.8495

Table V-16. Optimization results with diffusion ratio as equality constraint for $N=600,000$ rpm and $B_1=0.02$

In Table V-16 it can be noticed that the highest efficiency can be found at DR =1.67 (original).

Furthermore it can be noticed that the variable Z_r is not an integer, so it must be converted into integer to continue with calculations (Table V-17).

DR original [-]	1.67	1.7	1.8	1.9	2.0	2.1	2.2
DR [-]	1.6701	1.7351	1.8000	1.9002	2.0172	2.1110	2.2202
r_{1h} [m]	0.0018	0.0015	0.0025	0.0021	0.0024	0.0025	0.0025
D_{1h} [m]	0.0036	0.0030	0.0050	0.0042	0.0048	0.0050	0.0050
(M_{1t}) [-]	0.8071	0.8005	0.8637	0.8299	0.8633	0.8770	0.8675
r_{1t} [m]	0.0036	0.0036	0.0040	0.0038	0.0040	0.0041	0.0041
D_{1t} [m]	0.0072	0.0072	0.0080	0.0076	0.0080	0.0082	0.0082
r_{1t}/r_{1h} [-]	2.0213	2.0978	1.6345	1.8172	1.6360	1.5818	1.6187
β_1 [°]	56.8148	56.6768	59.3181	58.8047	59.4955	59.7525	59.8789
A_{fl} [m ²]	3.1601e-5	3.1714e-5	3.1661e-5	3.2283e-5	3.1799e-5	3.1600e-5	3.1963e-5
U_{1t} [m/s]	229.3032	227.0938	252.1632	241.2259	252.5821	257.1874	254.8671
W_{1t} [m/s]	273.9893	271.7782	293.2083	282.0013	293.1583	297.7198	294.6555
$C_{m1}=C_1$ [m/s]	149.9673	149.3043	149.6156	146.0643	148.8089	149.9724	147.8668
Z_r [-]	11	13	12	13	21	22	19
β_{B2} [°]	-42	-40	-43	-33	-39	-36	-25
η_{stage} [-]	0.5679	0.5563	0.5661	0.5421	0.5365	0.5282	0.5100
b_2 [m]	7.6148e-4	7.4637e-4	7.5891e-4	7.2750e-4	7.2005e-4	7.0932e-4	6.8501e-4
r_2 [m]	0.0067	0.0066	0.0066	0.0066	0.0066	0.0065	0.0065
U_2 [m/s]	417.8814	415.2884	417.5061	412.6812	411.7447	410.4444	408.2624
W_2 [m/s]	164.0509	156.6369	162.8929	148.4050	145.3305	141.0294	132.7154
M_2 [-]	0.7462	0.7663	0.7493	0.7912	0.8013	0.8162	0.8498

Table V-17. Optimization results after converting Z_r into an integer (values closed to the solutions) for $N=600,000$ rpm and $B_f=0.02$

In Table V-17 it can be noticed that the highest efficiency is still present for DR=1.67 (original). The same has been concluded in the previous table.

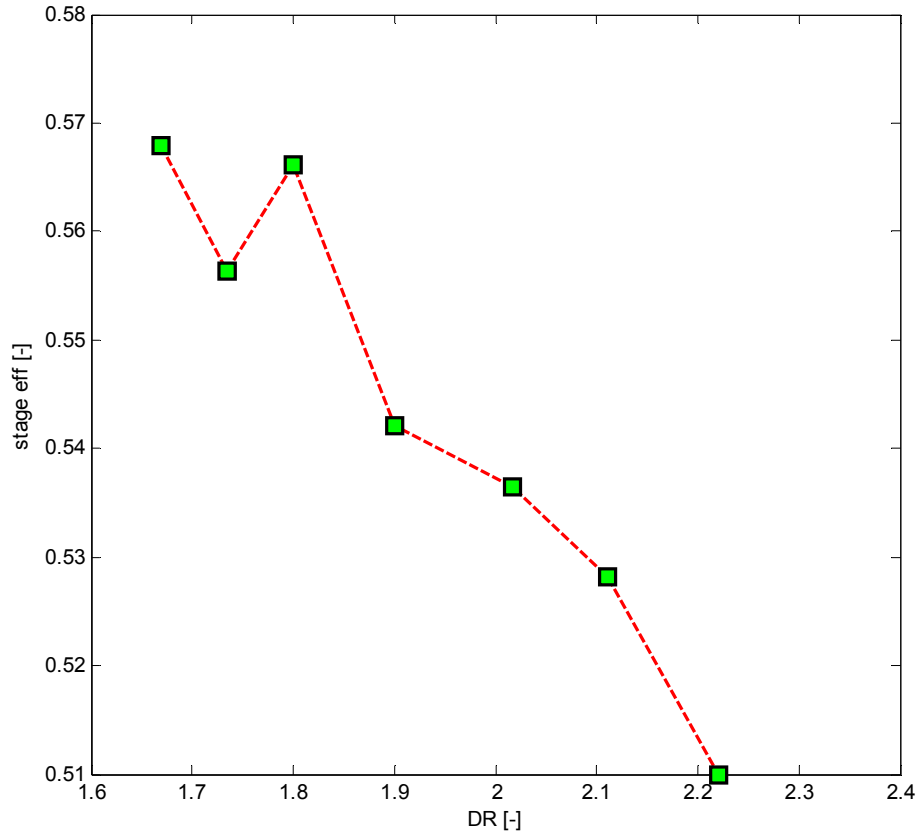


Figure 5-8 η_{stage} as function of DR for $N=600,000$ rpm and $B_I=0.02$

DR original [-]	1.67	1.7	1.8	1.9	2.0	2.1	2.2
DR [-]	1.6700	1.6998	1.7998	1.8893	2.0248	2.1188	2.2253
r_{1h} [m]	0.0022	0.0020	0.0022	0.0027	0.0024	0.0020	0.0022
D_{1h} [m]	0.0044	0.0040	0.0044	0.0054	0.0048	0.0040	0.0044
(M_{1t}) [-]	0.8395	0.8223	0.8374	0.8869	0.8757	0.8440	0.8624
r_{1t} [m]	0.0039	0.0038	0.0041	0.0042	0.0041	0.0043	0.0045
D_{1t} [m]	0.0078	0.0076	0.0082	0.0084	0.0082	0.0086	0.0090
r_{1t}/r_{1h} [-]	1.7753	1.8975	1.8658	1.5592	1.5994	2.0165	2.0391
β_1 [°]	59.0100	59.4220	64.8079	60.1877	59.6947	68.9160	71.2815
A_{fl} [m ²]	3.2475e-5	3.3326e-5	3.8032e-5	3.1974e-5	3.1922e-5	4.3748e-5	4.7552e-5
U_{1t} [m/s]	244.4882	240.7959	258.9489	261.2521	256.6328	270.0097	280.4957
W_{1t} [m/s]	285.1982	279.6902	286.1675	301.1000	297.2527	289.3827	296.1606
$C_{m1}=C_1$ [m/s]	146.8453	142.2813	121.8085	149.6949	149.9960	104.1015	95.0434
Z_r [-]	8.3808	10.8199	14.6022	13.7814	15.7399	23.2603	23.4253
β_{B2} [°]	-39.7686	-41.9613	-43.7820	-43.1017	-35.6345	-32.5678	-29.0026
η_{stage} [-]	0.5779	0.5686	0.5600	0.5606	0.5392	0.5188	0.5108
b_2 [m]	7.7440e-4	7.6197e-4	7.5059e-4	7.5193e-4	7.2333e-4	6.9656e-4	6.8611e-4
r_2 [m]	0.0067	0.0067	0.0066	0.0066	0.0066	0.0065	0.0065
U_2 [m/s]	420.3463	418.1250	416.2026	416.2478	412.2410	409.2847	408.3685
W_2 [m/s]	170.7774	164.5407	158.9992	159.3714	146.8071	136.5800	133.0885
M_2 [-]	0.7293	0.7451	0.7600	0.7587	0.7965	0.8335	0.8482

Table V-18. Optimization results with diffusion ratio as equality constraint for $N=600,000$ rpm and $B_I=0.03$

In Table V-18 it can be noticed that the highest efficiency can be found for $DR = 1.67$ (original). Furthermore, it can be noticed that the variables β_{B2} and Z_r are not integers, so they must be converted into integers to continue with calculations (Table V-19).

DR original [-]	1.67	1.7	1.8	1.9	2.0	2.1	2.2
DR [-]	1.6891	1.7214	1.8090	1.9165	2.0033	2.1283	2.1898
r_{1h} [m]	0.0020	0.0020	0.0022	0.0027	0.0025	0.0023	0.0025
D_{1h} [m]	0.0040	0.0040	0.0044	0.0054	0.0050	0.0046	0.0050
(M_{1t}) [-]	0.8309	0.8310	0.8446	0.8968	0.8665	0.8600	0.8598
r_{1t} [m]	0.0038	0.0039	0.0039	0.0042	0.0040	0.0040	0.0040
D_{1t} [m]	0.0076	0.0078	0.0078	0.0084	0.0080	0.0080	0.0080
r_{1t}/r_{1h} [-]	1.8333	1.8313	1.7447	1.5277	1.6363	1.6656	1.6663
β_1 [°]	58.2242	59.0145	59.5246	60.4868	59.4726	59.2159	59.4499
A_{fl} [m ²]	3.2166e-5	3.2739e-5	3.2704e-5	3.1925e-5	3.2018e-5	3.2019e-5	3.2196e-5
U_{1t} [m/s]	239.8759	242.1178	247.3961	264.9447	253.4150	250.8309	251.4589
W_{1t} [m/s]	282.1687	282.4197	287.0533	304.4493	294.1944	291.9687	291.9918
$C_{m1}=C_1$ [m/s]	148.5891	145.3955	145.5842	149.9788	149.4362	149.4310	148.4172
Z_r [-]	9	11	15	14	16	23	23
β_{B2} [°]	-39	-42	-44	-43	-36	-33	-29
η_{stage} [-]	0.5724	0.5679	0.5595	0.5598	0.5393	0.5201	0.5114
b_2 [m]	7.6689e-4	7.6129e-4	7.5019e-4	7.5060e-4	7.2376e-4	6.9820e-4	6.8687e-4
r_2 [m]	0.0067	0.0067	0.0066	0.0066	0.0066	0.0065	0.0065
U_2 [m/s]	419.0308	417.9180	416.0553	416.1096	412.2094	409.4624	408.4359
W_2 [m/s]	167.0541	164.0653	158.6817	158.8578	146.8561	137.1845	133.3396
M_2 [-]	0.7387	0.7462	0.7608	0.7603	0.7962	0.8311	0.8471

Table V-19. Optimization results after converting Z_r into an integer (values closed to the solutions) for $N=600,000$ rpm and $B_I=0.03$

In Table V-19 it can be noticed that the highest efficiency is still present for $DR=1.67$ (original). The same has been concluded in the previous table.

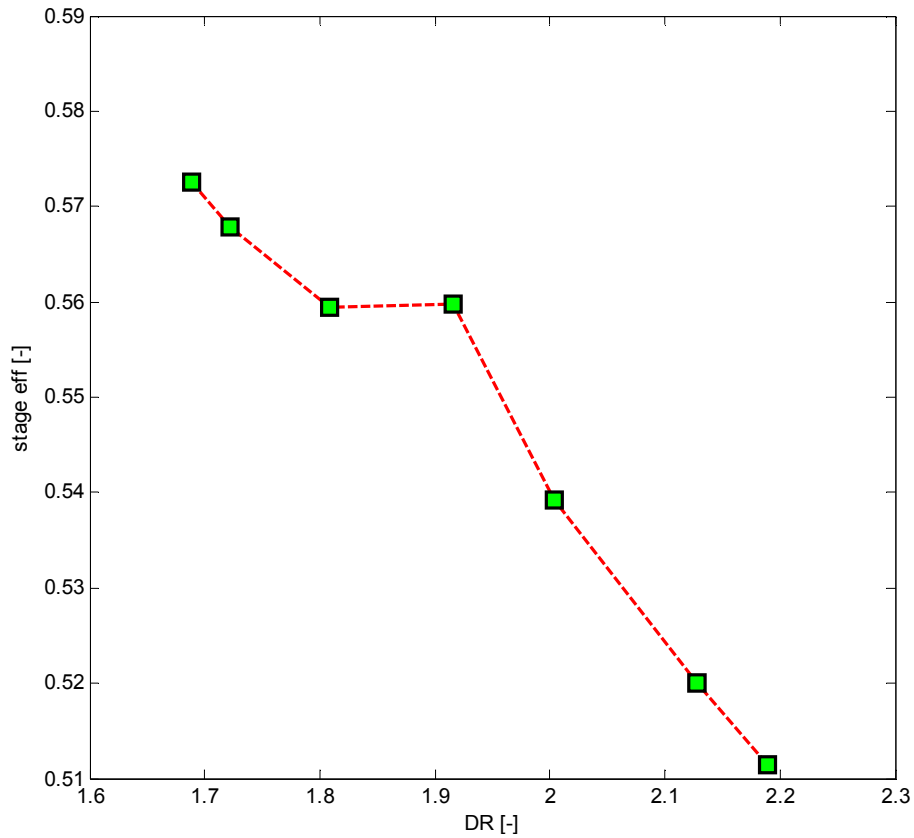


Figure 5-9 η_{stage} as function of DR for $N=600,000$ rpm and $B_I=0.03$

DR original [-]	1.67	1.7	1.8	1.9	2.0	2.1	2.2
DR [-]	1.6752	1.7	1.8	1.9182	2	2.1002	2.2
r_{1h} [m]	0.0022	0.002	0.0027	0.0021	0.002	0.0019	0.0013
D_{1h} [m]	0.0044	0.004	0.0054	0.0042	0.004	0.0038	0.0026
(M_{1t}) [-]	0.8490	0.8199	0.8933	0.8370	0.8168	0.8187	0.8363
r_{1t} [m]	0.0039	0.0039	0.0042	0.0039	0.0039	0.0037	0.0044
D_{1t} [m]	0.0078	0.0078	0.0084	0.0078	0.0078	0.0074	0.0088
r_{1t}/r_{1h} [-]	1.7379	1.9479	1.5497	1.8118	1.9946	1.9587	3.3177
β_1 [°]	59.1729	60.0972	60.3879	59.3371	61.5730	57.8560	73.4007
A_{fl} [m ²]	3.2644e-5	3.4303e-5	3.2281e-5	3.3137e-5	3.5728e-5	3.261e-5	5.5018e-5
U_{1t} [m/s]	247.6396	241.9341	263.6425	244.7124	244.8915	235.4437	275.7653
W_{1t} [m/s]	288.3831	279.0885	303.2498	284.4889	278.4679	278.0675	287.7572
$C_{m1}=C_1$ [m/s]	147.7818	139.1340	149.8435	145.0854	132.5615	147.9454	82.2055
Z_r [-]	9.6681	11.7314	9.4690	13.5016	21.6434	24.3227	24.5877
β_{B2} [°]	-44.7555	-43.4465	-41.5899	-33.7896	-34.0915	-28.7988	-25.6598
η_{stage} [-]	0.5798	0.5681	0.5745	0.5419	0.5244	0.5092	0.5022
b_2 [m]	7.7693e-4	7.6149e-4	7.6998e-4	7.2701e-4	7.0392e-4	6.8398e-4	6.7471e-4
r_2 [m]	0.0067	0.0067	0.0067	0.0066	0.0065	0.0065	0.0065
U_2 [m/s]	420.8649	417.9553	419.4995	412.6939	410.0245	408.1899	407.5058
W_2 [m/s]	172.1479	164.1694	168.4715	148.3132	139.2340	132.4027	129.6080
M_2 [-]	0.7260	0.7460	0.7350	0.7916	0.8231	0.8512	0.8645

Table V-20. Optimization results with diffusion ratio as equality constraint for $N=600,000$ rpm and $B_I=0.04$

In Table V-20 it can be noticed that the highest efficiency can be found for $DR = 1.67$ (original).

Furthermore, it can be noticed that the variables β_{B2} and Z_r are not integers, so they must be converted into integers to continue with calculations (Table V-21).

DR original [-]	1.67	1.7	1.8	1.9	2.0	2.1	2.2
DR [-]	1.6861	1.7173	1.8002	1.9041	2.0001	2.1261	2.2197
r_{1h} [m]	0.0022	0.0019	0.0019	0.002	0.0019	0.0019	0.0022
D_{1h} [m]	0.0044	0.0038	0.0038	0.004	0.0038	0.0038	0.0044
(M_{1t}) [-]	0.8509	0.8240	0.8184	0.8281	0.8182	0.8262	0.8481
r_{1t} [m]	0.0039	0.0038	0.0037	0.0038	0.0037	0.0040	0.0039
D_{1t} [m]	0.0078	0.0076	0.0074	0.0076	0.0074	0.0080	0.0078
r_{1t}/r_{1h} [-]	1.7279	1.9140	1.9340	1.8793	1.9670	1.9647	1.7438
β_1 [°]	58.8919	57.5814	57.7468	57.8164	57.3620	64.0640	58.6616
A_{fl} [m²]	3.2382e-5	3.2260e-5	3.2212e-5	3.2298e-5	3.2287e-5	3.7985e-5	3.2298e-5
U_{1t} [m/s]	247.3574	236.1340	235.0537	237.9429	233.9085	253.8142	245.9130
W_{1t} [m/s]	288.9036	279.7287	277.9404	281.1414	277.7697	282.2406	287.9171
C_{m1}=C₁ [m/s]	149.2635	149.9629	148.3262	149.7453	149.8093	123.4427	149.7432
Z_r [-]	10	12	16	14	22	24	25
β_{B2} [°]	-45	-43	-42	-34	-34	-29	-26
η_{stage} [-]	0.5787	0.5661	0.5525	-0.5407	-0.5237	0.5101	0.5024
b₂ [m]	7.7575e-4	7.5891e-4	7.4120e-4	7.2543e-4	7.0307e-4	6.8521e-4	6.7454e-4
r₂ [m]	0.0067	0.0066	0.0066	0.0066	0.0065	0.0065	0.0065
U₂ [m/s]	420.5193	417.5062	414.5875	412.4901	409.9064	408.2586	407.6117
W₂ [m/s]	171.3450	162.8929	154.3941	147.6509	138.8775	132.7515	129.7081
M₂ [-]	0.7278	0.7493	0.7729	0.7938	0.8244	0.8496	0.8643

Table V-21. Optimization results after converting Z_r into an integer (values closed to the solutions) for $N=600,000$ rpm and $B_I=0.04$

In Table V-21 it can be noticed that the highest efficiency is still present for $DR=1.67$ (original). The same has been concluded in the previous table.

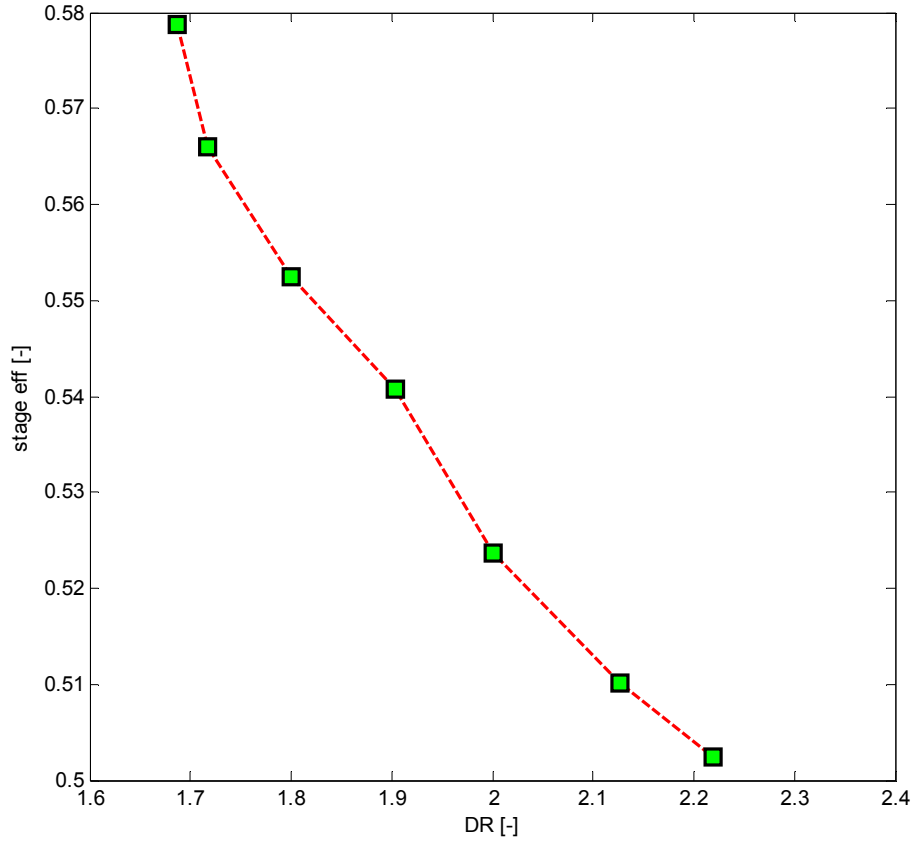


Figure 5-10 η_{stage} as function of DR for $N=600,000 \text{ rpm}$ and $B_I=0.04$

Figures 5-8 to 5-10 has been presented to give a clear view concerning DR as function of η_{stage} during optimization calculations.

Furthermore, the outcome of these tables (V16 to V21) has been influenced not only by DR but by several variables to reach the optimum. The DR can be considered a measure of stage efficiency, because it is an estimation of the amount of diffusion in the impeller and gives an indirect measurement of the boundary layer thickness in the impeller. This behaviour can be noticed in Figures 5-8 to 5-10. Figures 5-8 to 5-10 are not straight decreasing lines, because the number of blades and the blade backsweep angles are not the same for each blockage factor.

Flow separation is more likely to occur at high diffusion ratios, thus a lower diffusion ratio is desirable.

Anyway the highest η_{stage} has been found with $DR = 1.67$ and therefore this DR value will be used in the following calculations.

The following calculations will present the performance of the compressor by adjusting β_{B2} and Z_r .

5.9.1. Effects of backsweep angle on performance

In this section the effects of backsweep angle on performance for three blockage factors will be studied.

The designs with the highest η_{stage} has been found with DR = 1.67 for all three blockage factors (see Tables V-16 to V-21).

Anyway the results from all three designs presented above have been arranged again in Tables V-21 to V-23. Then some calculation have been performed by keeping fixed DR and Zr values from the designs presented above and varying β_{B2} between -45 and 0 (see Table V-22, V-23 and V-24) .

Tables V-22, V-23 and V-24 shows the calculation results in numbers whereas Figures 5-11 to 5-19 shows graphical results for a quick view of the effect for three different blockage factors.

DR original [-]	1.67	1.67	1.67
DR [-]	1.6700	1.6701	1.6700
r_{1h} [m]	0.0021	0.0018	0.0016
D_{1h} [m]	0.0042	0.0036	0.0032
(M_{1t}) [-]	0.8274	0.8071	0.7938
r_{1t} [m]	0.0038	0.0036	0.0036
D_{1t} [m]	0.0076	0.0072	0.0072
r_{1t}/r_{1h} [-]	1.8349	2.0213	2.1856
β_1 [°]	58.5934	56.8148	56.6403
A_{fl} [m ²]	3.2206e-5	3.1601e-5	3.1898e-5
U_{1t} [m/s]	239.9363	229.3032	225.1620
W_{1t} [m/s]	281.1234	273.9893	269.5791
$C_{mt}=C_l$ [m/s]	146.4955	149.9673	148.2397
Z_r [-]	11	11	11
β_{B2} [°]	-45	-42	-40
η_{stage} [-]	0.5743	0.5679	0.5639
b_2 [m]	7.6972e-4	7.6148e-4	7.5591e-4
r_2 [m]	0.0067	0.0067	0.0066
U_2 [m/s]	419.4505	417.8814	416.9948
W_2 [m/s]	168.3373	164.0509	161.4244
M_2 [-]	0.7353	0.7462	0.7533

Table V-22. Optimization results showing backsweep angle (β_{B2}) effects for $N=600,000$ rpm and $B_l=0.02$

In Table V-22 it can be noticed that for $\beta_{B2}=-30, -20, -10$ and 0 no solution has been obtained.

Moreover for those blade angles mentioned above the optimization calculation has been terminated for some reason and no solution has been obtained.

Anyway the optimization could be terminated (without solution), because of the following reasons:

- Stall time limit exceeded. Stall time limit is the time needed to solve the optimization problem.

The algorithm stops if there is no improvement in the objective function during an interval of time in seconds equal to Stall time limit.

- Stall generations (iterations) limit exceeded but constraints are not satisfied.

The algorithm stops if there is no improvement in the objective function for StallGenLimit consecutive generations but the calculated constraints do not mach the input constraint.

- Maximum number of generations (iterations) exceeded.

Furthermore it is possible to change the stall time limit and the stall generation, which could result in the third reason given above.

When there is no improvement in the objective function, this indicates that the Z_r, β_{B2} , the constraints and design space were mismatched and will not lead to a solution.

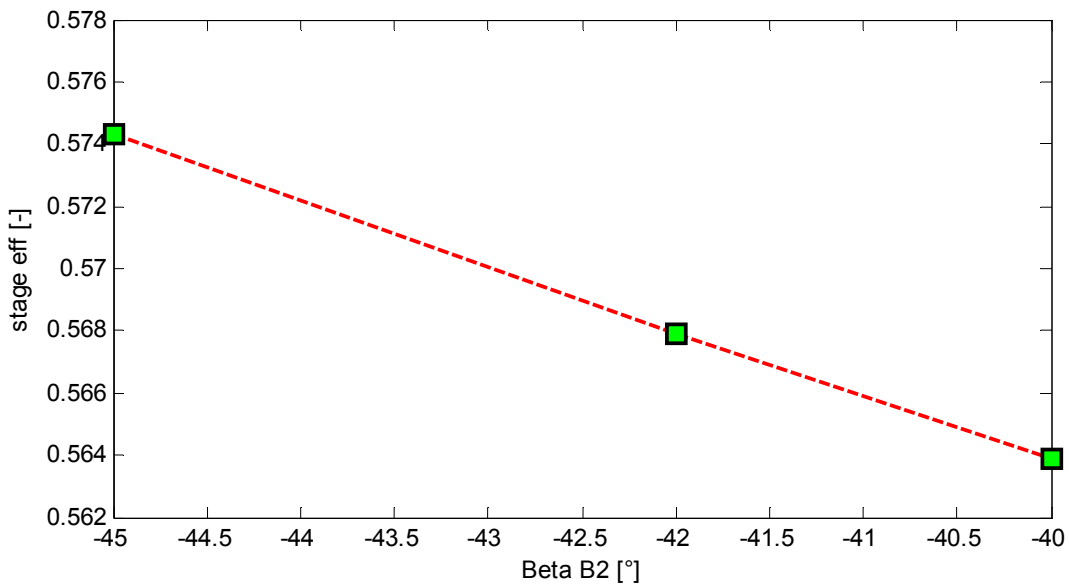


Figure 5-11 Effect of β_{B2} on η_{stage} for $N=600,000$ rpm and $B_1=0.02$

Figure 5-11 illustrates the effect of blade angle on stage efficiency. Here it can be noticed that the stage efficiency decreases as the blade backsweep angle decreases.

This means that decreasing blade backsweep angle have a negative effect on the stage efficiency.

Furthermore this behaviour has been reported by another author [25].

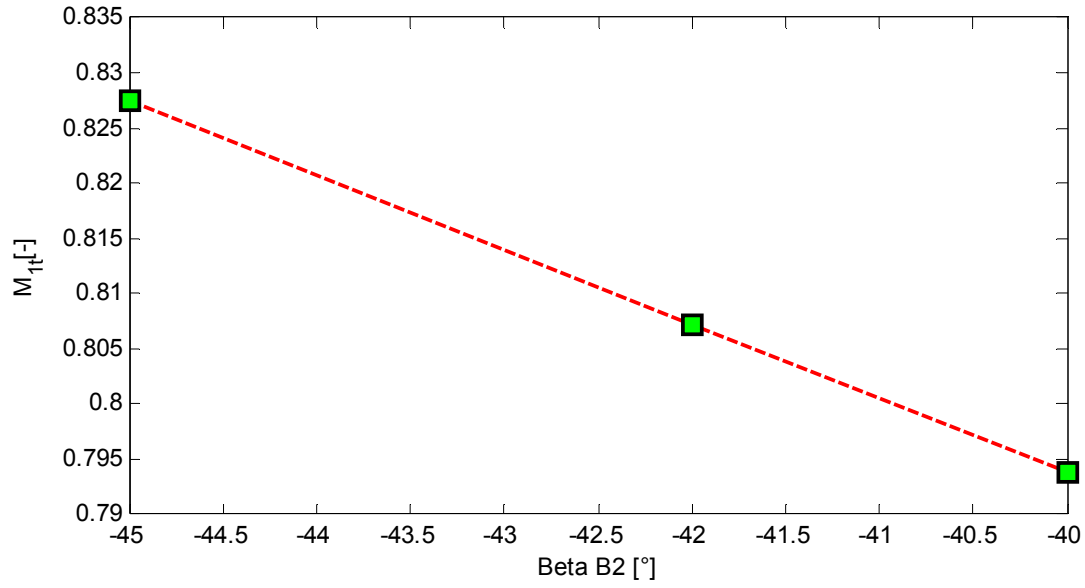


Figure 5-12 Effect of β_{B2} on M_{1t} for $N=600,000$ rpm and $B_I=0.02$

Figure 5-12 illustrates the effect of blade angle on tip Mach number. Here it can be noticed that decreasing the blade backsweep angle has a positive effect on the tip Mach number.

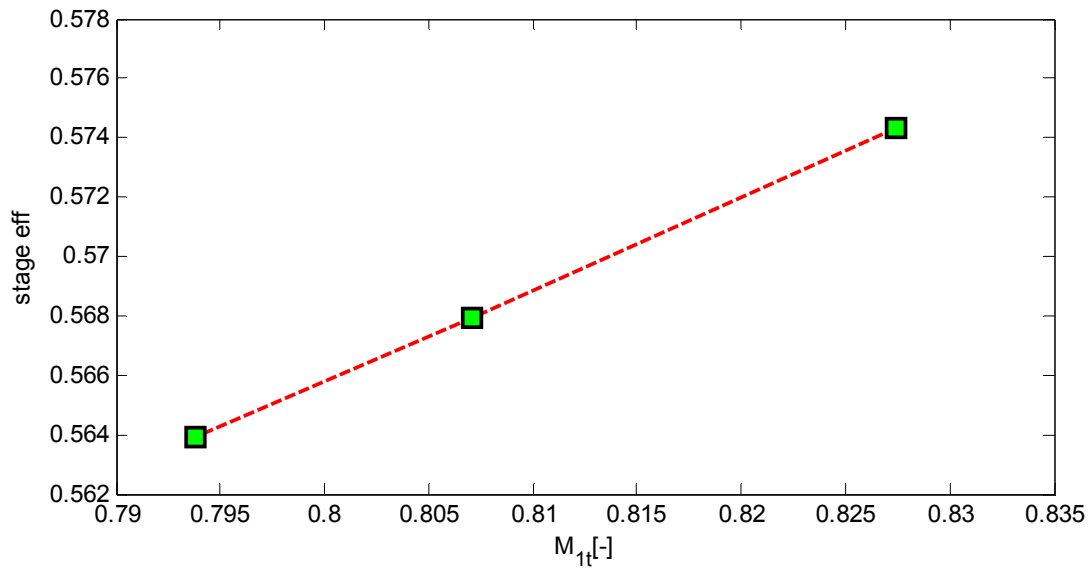


Figure 5-13 η_{stage} as function of M_{1t} for $N=600,000$ rpm and $B_I=0.02$

In this figure it can be noticed that the tip Mach number has a positive effect on stage efficiency as it increases, but the tip Mach number is limited.

The same procedure as described above has been performed for the $N=600,000$ rpm and $B_I=0.03$ design.

Table V-22 shows the calculation results in numbers whereas Figures 5-14 5-15 and 5-16 shows graphical results for a quick view of the effect

DR original [-]	1.67	1.67	1.67	1.67
DR [-]	1.6700	1.6910	1.6891	1.6702
r_{1h} [m]	0.0024	0.0021	0.0020	0.0013
D_{1h} [m]	0.0048	0.0042	0.0040	0.0026
(M_{1t}) [-]	0.8609	0.8358	0.8309	0.7710
r_{1t} [m]	0.0040	0.0040	0.0038	0.0035
D_{1t} [m]	0.0080	0.0080	0.0076	0.0070
r_{1t}/r_{1h} [-]	1.6612	1.8044	1.8333	2.7167
β_1 [°]	59.3758	60.7813	58.2242	56.6313
A_n [m ²]	3.2109e-5	3.3999e-5	3.2166e-5	3.2966e-5
U_{1t} [m/s]	251.5558	248.3247	239.8759	218.9046
W_{1t} [m/s]	292.3275	284.5272	282.1687	262.1147
$C_{m1}=C_1$ [m/s]	148.9129	138.8905	148.5891	144.1698
Z_r [-]	9	9	9	9
β_{B2} [°]	-45	-40	-39	-30
η_{stage} [-]	0.5839	0.5742	0.5724	0.5567
b_2 [m]	7.8239e-4	7.6982e-4	7.6689e-4	7.4674e-4
r_2 [m]	0.0067	0.0067	0.0067	0.0066
U_2 [m/s]	421.9489	419.3875	419.0308	415.4299
W_2 [m/s]	175.0465	168.2616	167.0541	156.9382
M_2 [-]	0.7190	0.7354	0.7387	0.7656

Table V-23. Optimization results showing backsweep angle (β_{B2}) effects for $N=600,000$ rpm and $B_1=0.03$

The optimization calculation for $\beta_{B2}=-20, -10$ and 0 has been terminated for some reasons and no solution has been obtained.

Anyway the following reasons for optimization termination have been found here:

- stall time limit exceeded;
- stall generations limit exceeded but constraints are not satisfied.

These reasons have been explained above.

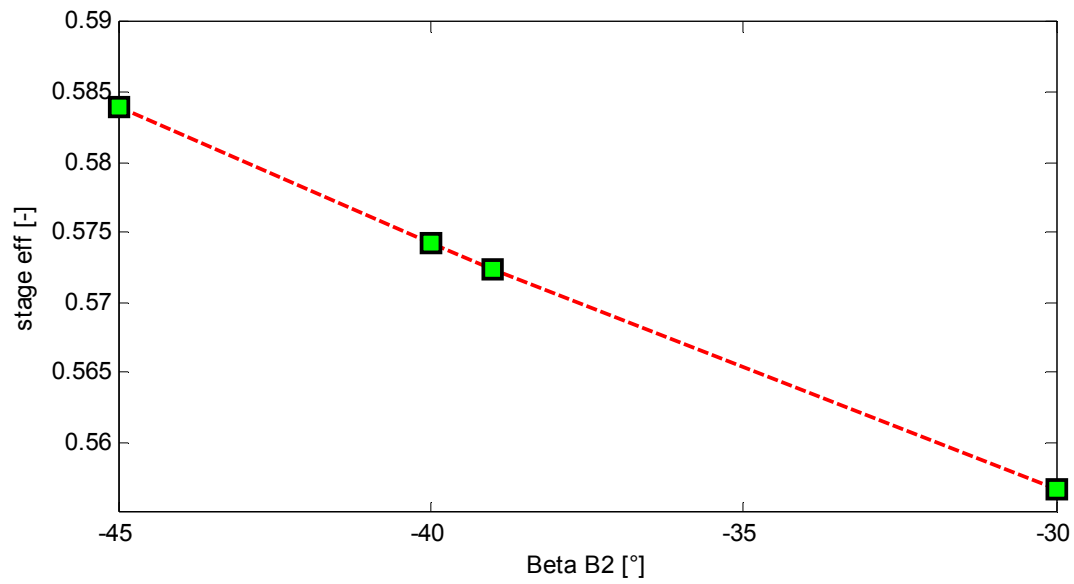


Figure 5-14 Effect of β_{B2} on η_{stage} for $N=600,000$ rpm and $B_1=0.03$

Figure 5-14 illustrates the same effect as Figure 5-11. Here it can be noticed that the stage efficiency decreases as the blade backsweep angle decreases.

This means that, decreasing blade backsweep angle have a negative effect on the stage efficiency.

Furthermore the same have been noticed by another author [25], which reported that every 10° of backsweep is an increase of one or two points of efficiency.

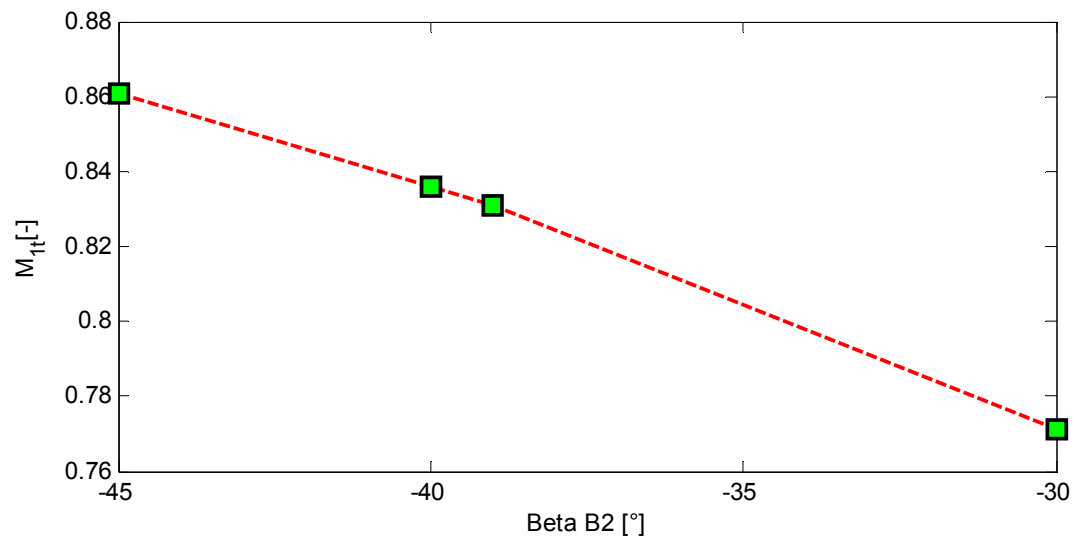


Figure 5-15 Effect of β_{B2} on M_t for $N=600,000$ rpm and $B_1=0.03$

Figure 5-15 illustrates the effect of blade angle on tip Mach number. Here it can be noticed that the overall effect is almost the same as the previous Figure 5-12.

Furthermore decreasing the blade backsweep angle has a positive effect on the tip Mach number.

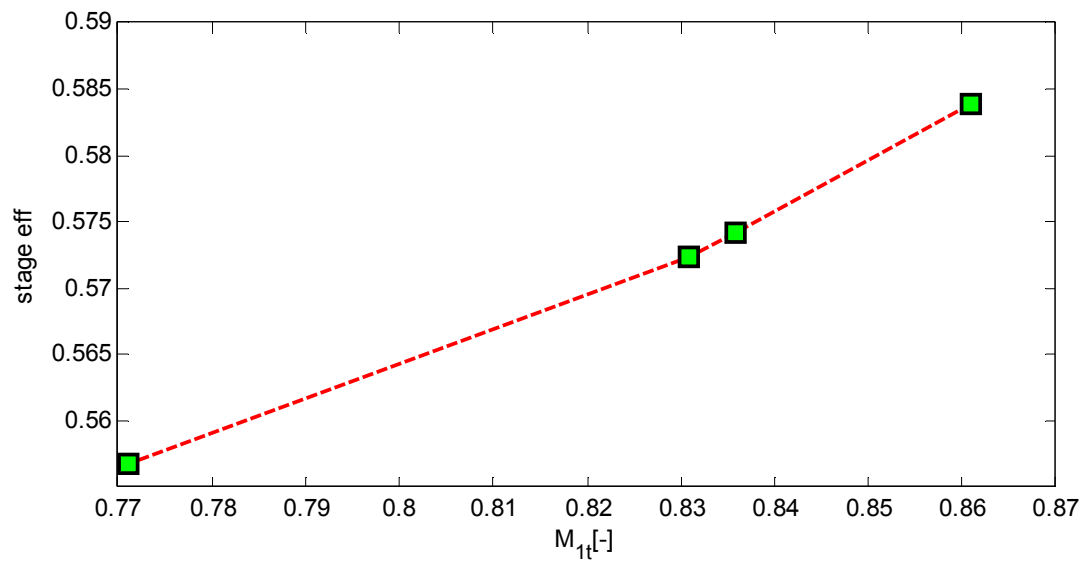


Figure 5-16 η_{stage} as function of M_{1t} for $N=600,000 \text{ rpm}$ and $B_1=0.03$

In this figure it can be noticed that stage efficiency increases as Mach number increases.

As stated before, tip Mach number have a positive effect on stage efficiency as it increases.

The same procedure as described above have been performed for the $N=600,000 \text{ rpm}$ and $B_1=0.04$ design.

Table V-24 shows the calculation results in numbers whereas Figures 5-17 5-18 and 5-19 shows graphical results for a quick view of the effect.

DR original [-]	1.67	1.67
DR [-]	1.6861	1.6702
r_{1h} [m]	0.0022	0.0017
D_{1h} [m]	0.0044	0.0034
(M_{1t}) [-]	0.8509	0.8025
r_{1t} [m]	0.0039	0.0040
D_{1t} [m]	0.0078	0.0080
r_{1t}/r_{1h} [-]	1.7279	2.3804
β_1 [°]	58.8919	65.7588
$A_{fl}[m^2]$	3.2382e-5	4.1125e-5
U_{1t} [m/s]	247.3574	250.5055
W_{1t} [m/s]	288.9036	274.7298
$C_{m1}=C_1$ [m/s]	149.2635	112.7982
Z_r [-]	10	10
β_{B2} [°]	-45	-40
η_{stage} [-]	0.5787	0.5686
b_2 [m]	7.7575e-4	7.6241e-4
r_2 [m]	0.0067	0.0067
U_2 [m/s]	420.5193	418.0303
W_2 [m/s]	171.3450	164.4911
M_2 [-]	0.7278	0.7450

Table V-24. Optimization results showing backsweep angle (β_{B2}) effects for $N=600,000$ rpm and $B_1=0.04$

The optimization calculation for $\beta_{B2}=-30, -20, -10$ and 0 has been terminated for some reasons and no solution has been obtained.

Anyway the following reasons for optimization termination have been found here:

- stall time limit exceeded;
- stall generations limit exceeded but constraints are not satisfied.

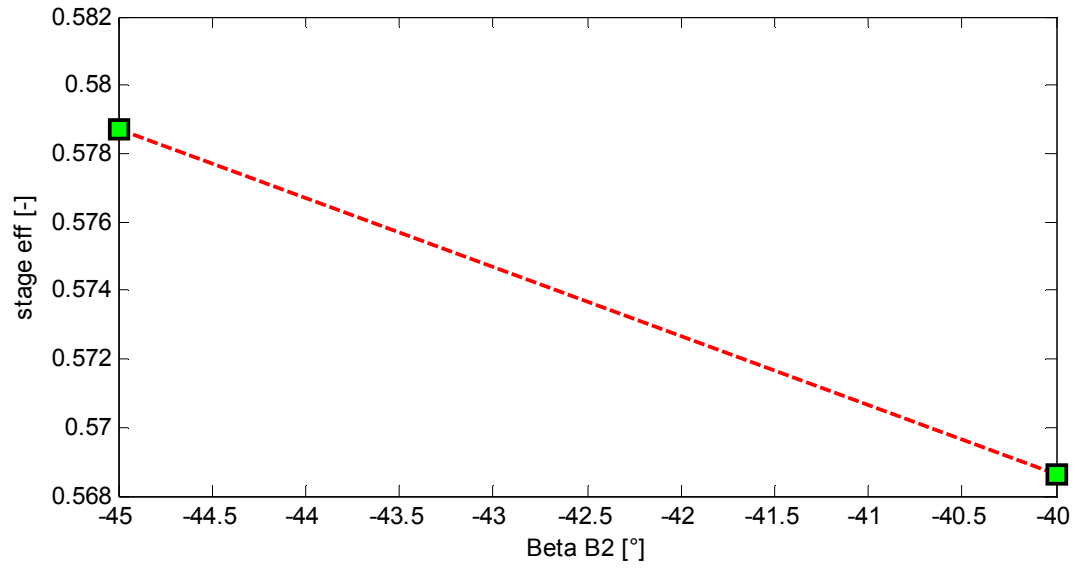


Figure 5-17 Effect of β_{B2} on η_{stage} for $N=600,000$ rpm and $B_I=0.04$

Figure 5-17 illustrates the same trend presented in Figure 5-11 and 5-14

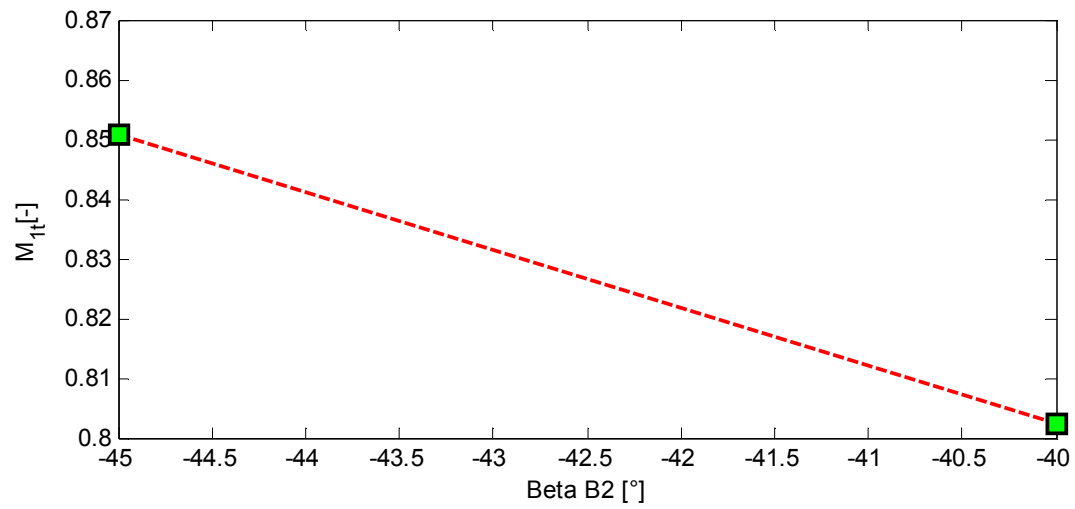


Figure 5-18 Effect of β_{B2} on M_{1t} for $N=600,000$ rpm and $B_I=0.04$

Figure 5-18 illustrates the same trend presented in Figure 5-12 and 5-15

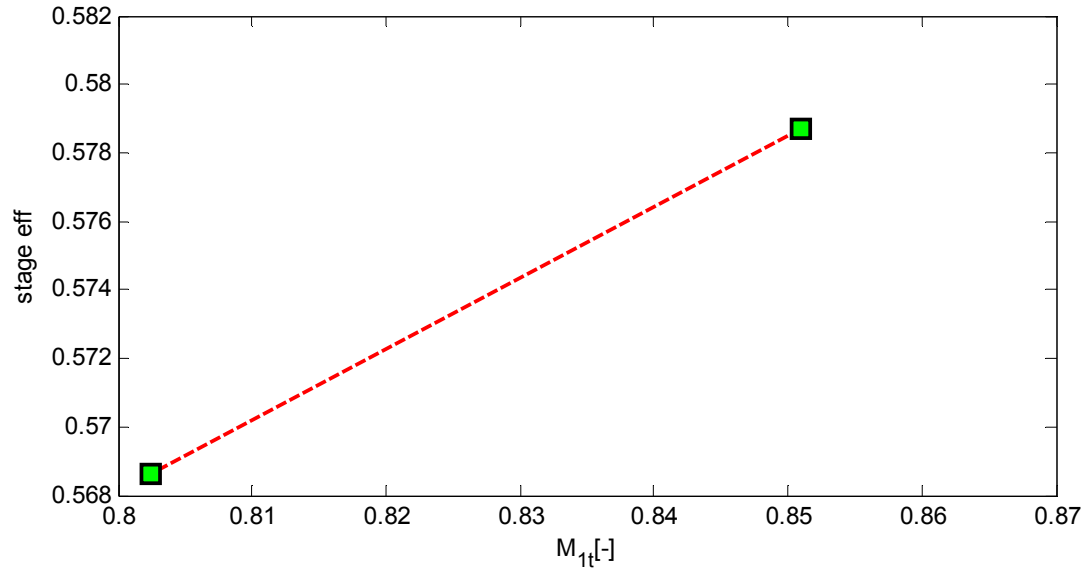


Figure 5-19 η_{stage} as function of M_{1t} for $N=600,000 \text{ rpm}$ and $B_I=0.04$

Figure 5-19 illustrates the same trend presented in Figure 5-13 and 5-16.

5.9.2 Effects of blade number on performance

In this section the effects of blade number on performance for both rotational speeds will be studied.

The designs with the highest η_{stage} has been found with $DR = 1.67$ for all three blockage factors (see Table V-16 to V-21).

Anyway the results from all three designs presented above have been arranged again in Table V-25 to V-27. Then some calculations have been performed by holding the DR and β_{B2} values from the designs presented above fixed and varying Zr between 5 and 30 (see Table V-25, V-26 and V-27).

Table V-25, V26 and V27 shows the calculation results in numbers whereas Figures 5-20 to 5-28 shows graphical results for a quick view of the effect.

DR original [-]	1.67	1.67	1.67
DR [-]	1.6700	1.6701	1.6854
r_{1h} [m]	0.0020	0.0018	0.0012
D_{1h} [m]	0.0040	0.0036	0.0024
(M_{1t}) [-]	0.8213	0.8071	0.7725
r_{1t} [m]	0.0038	0.0036	0.0035
D_{1t} [m]	0.0076	0.0072	0.0070
r_{1t}/r_{1h} [-]	1.8808	2.0213	2.5905
β_1 [°]	58.5324	56.8148	56.6682
A_{fl} [m ²]	3.2349e-5	3.1601e-5	3.2604e-5
U_{1t} [m/s]	238.0604	229.3032	219.4223
W_{1t} [m/s]	279.1072	273.9893	262.6230
$C_{m1}=C_1$ [m/s]	145.6986	149.9673	144.3076
Z_r [-]	10	11	15
β_{B2} [°]	-42	-42	-42
η_{stage} [-]	0.5725	0.5679	0.5549
b_2 [m]	7.6736e-4	7.6148e-4	7.4433e-4
r_2 [m]	0.0067	0.0067	0.0066
U_2 [m/s]	419.0128	417.8814	415.0573
W_2 [m/s]	167.1300	164.0509	155.8215
M_2 [-]	0.7384	0.7462	0.7687

Table V-25. Optimization results showing blade number (Z_r) effects for $N=600,000$ rpm and $B_I=0.02$

In Table V-25 it can be noticed that for $Z_r=5, 20, 25$ and 30 no solution has been obtained
The same reasons explained in section 5.9.1 could be applied here.

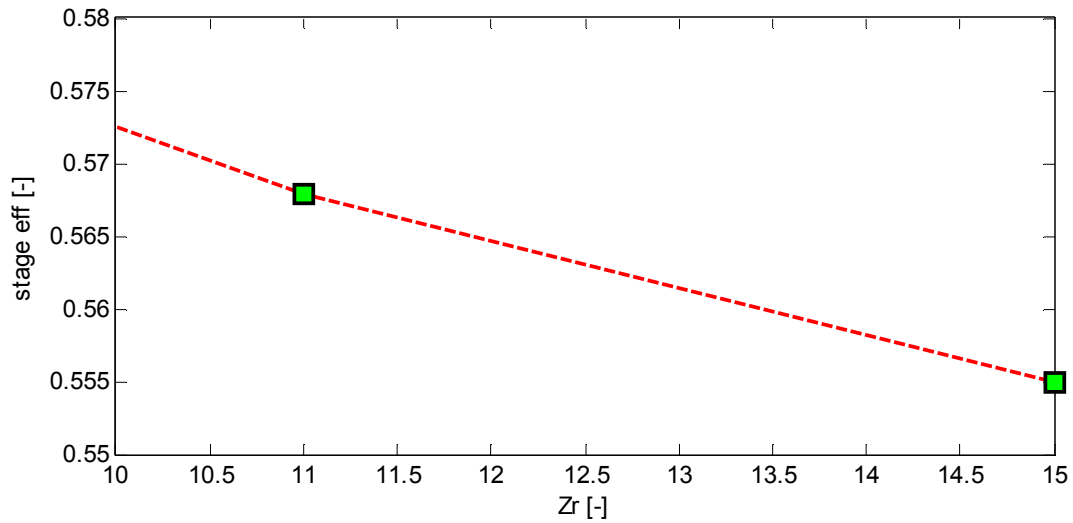


Figure 5-20 Effect of Z_r on η_{stage} for $N=600,000$ rpm and $B_I=0.02$

Figure 5-20 illustrates the influence of blade number on the stage efficiency.

Here it can be noticed that the stage efficiency decreases as the number of blades increases. This means that increasing number of blades has a negative effect on the stage efficiency.

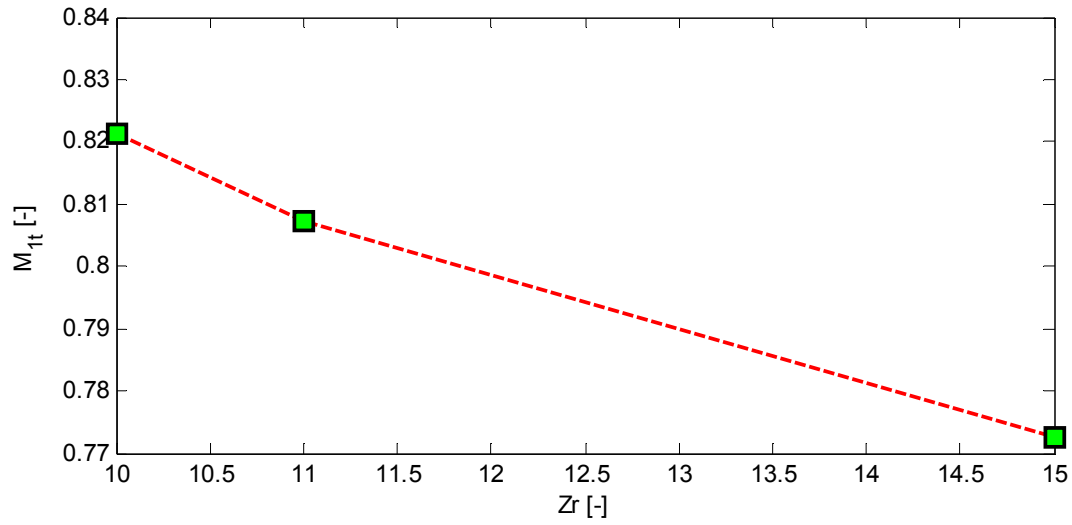


Figure 5-21 Effect of Z_r on M_{1t} for $N=600,000$ rpm and $B_I=0.02$

Figure 5-21 illustrates the influence of blade number on the tip Mach number.

Here it can be noticed that the tip Mach number decreases as the number of blades increases. This means that decreasing number of blades has a positive effect on the tip Mach number.

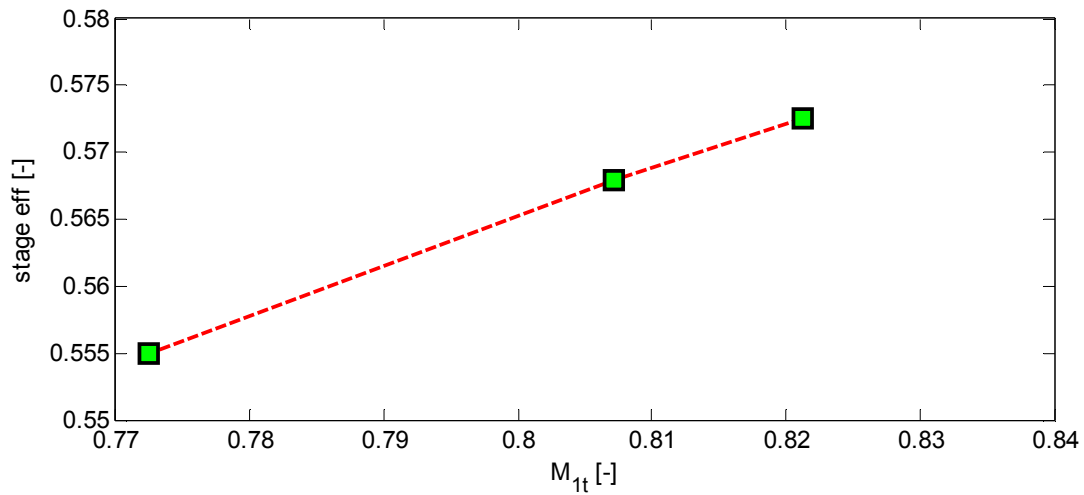


Figure 5-22 η_{stage} as function of M_{1t} for $N=600,000$ rpm and $B_I=0.02$

In this figure it can be noticed that the tip Mach number have a positive effect on stage efficiency as it increases, but the tip Mach number is limited.

The same procedure as described above have been performed for $N=600,000 \text{ rpm}$ and $B_I=0.03$ design.

Table V-26 shows the calculation results in numbers whereas Figures 5-23 5-24 and 5-25 shows graphical results for a quick view of the effect.

DR original [-]	1.67	1.67
DR [-]	1.6891	1.6730
r_{1h} [m]	0.0020	0.0015
D_{1h} [m]	0.0040	0.0030
(M_{1t}) [-]	0.8309	0.7970
r_{1t} [m]	0.0038	0.0040
D_{1t} [m]	0.0076	0.0080
r_{1t}/r_{1h} [-]	1.8333	2.5692
β_1 [°]	58.2242	66.8975
A_{fl} [m²]	3.2166e-5	4.2614e-5
U_{1t} [m/s]	239.8759	251.2186
W_{1t} [m/s]	282.1687	273.1218
$C_{m1}=C_1$ [m/s]	148.5891	107.1668
Z_r [-]	9	10
β_{B2} [°]	-39	-39
η_{stage} [-]	0.5724	0.5667
b_2 [m]	7.6689e-4	7.5964e-4
r_2 [m]	0.0067	0.0066
U_2 [m/s]	419.0308	417.6314
W_2 [m/s]	167.0541	163.2503
M_2 [-]	0.7387	0.7484

Table V-26. Optimization results showing blade number (Z_r) effects for $N=600,000 \text{ rpm}$ and $B_I=0.03$

The optimization calculation for $Z_r=5, 15, 20, 25$ and 30 has been terminated for the same reasons described above.

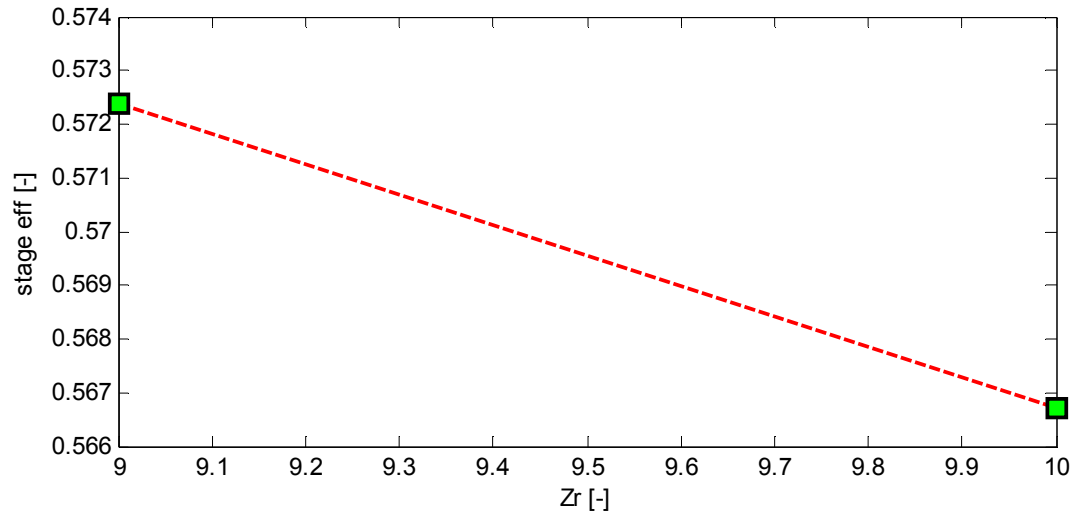


Figure 5-23 Effect of Z_r on η_{stage} for $N=600,000$ rpm and $B_I=0.03$

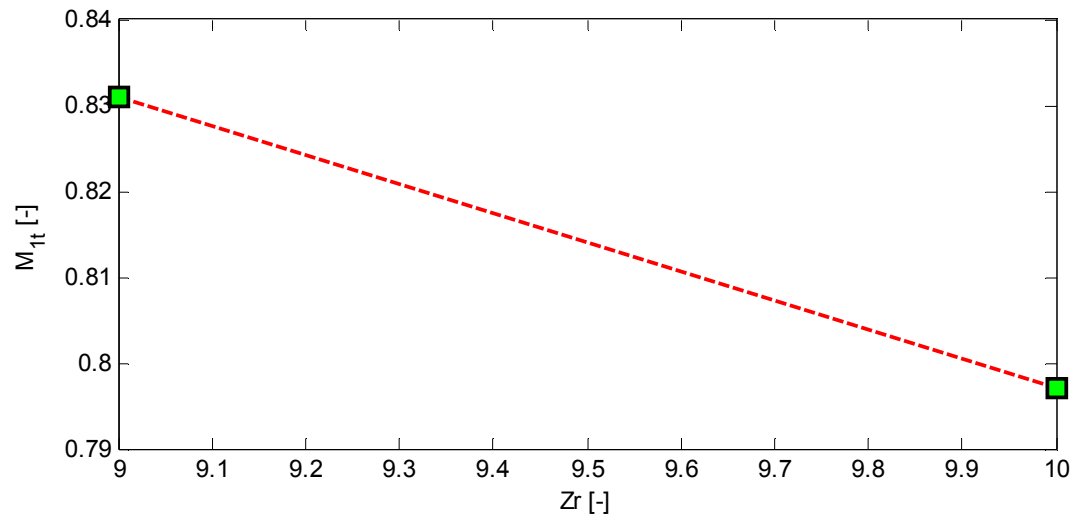


Figure 5-24 Effect of Z_r on M_{1t} for $N=600,000$ rpm and $B_I=0.03$

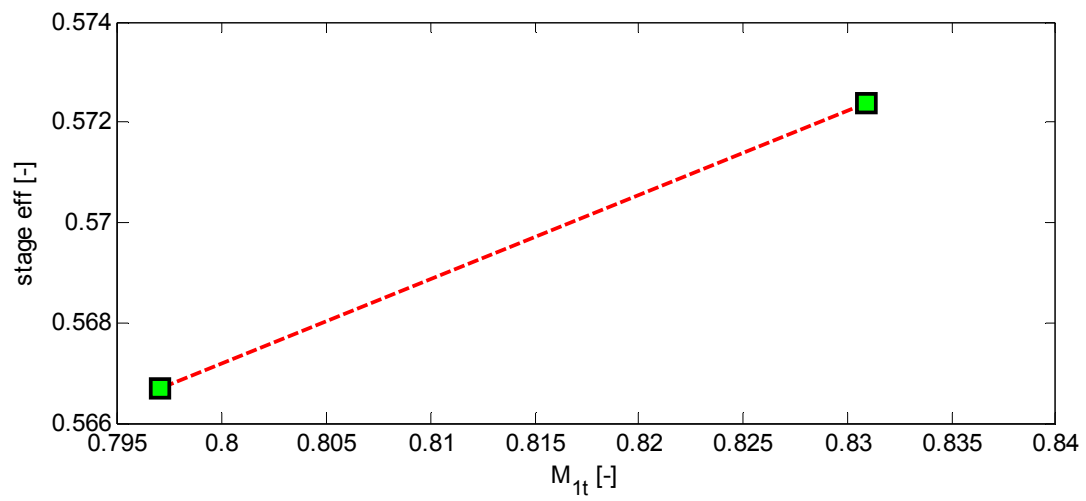


Figure 5-25 η_{stage} as function of M_{1t} for $N=600,000$ rpm and $B_I=0.03$

The same procedure as described above have been performed for $N=600,000$ rpm and $B_I=0.04$ design. Table V-26 shows the calculation results in numbers whereas Figures 5-26 5-27 and 5-28 shows graphical results for a quick view of the effect.

DR original [-]	1.67	1.67	1.67
DR [-]	1.6861	1.6700	1.6700
r_{1h} [m]	0.0022	0.0015	0.0010
D_{1h} [m]	0.0044	0.0030	0.0020
(M_{1t}) [-]	0.8509	0.7867	0.7581
r_{1t} [m]	0.0039	0.0036	0.0034
D_{1t} [m]	0.0078	0.0072	0.0068
r_{1t}/r_{1h} [-]	1.7279	2.3871	3.4159
β_1 [°]	58.8919	57.4951	56.7850
A_{fl} [m ²]	3.2382e-5	3.3391e-5	3.3872e-5
U_{1t} [m/s]	247.3574	225.5907	215.7638
W_{1t} [m/s]	288.9036	267.4950	257.8991
$C_{m1}=C_1$ [m/s]	149.2635	143.7443	141.2724
Z_r [-]	10	15	20
β_{B2} [°]	-45	-45	-45
η_{stage} [-]	0.5787	0.5619	0.5526
b_2 [m]	7.7575e-4	7.5333e-4	7.4101e-4
r_2 [m]	0.0067	0.0066	0.0066
U_2 [m/s]	420.5193	416.5648	414.6396
W_2 [m/s]	171.3450	160.1766	154.4303
M_2 [-]	0.7278	0.7566	0.7729

Table V-27. Optimization results showing blade number (Z_r) effects for $N=600,000$ rpm and $B_I=0.04$

The optimization calculation for $Z_r=5, 25$ and 30 has been terminated for the same reasons described above.

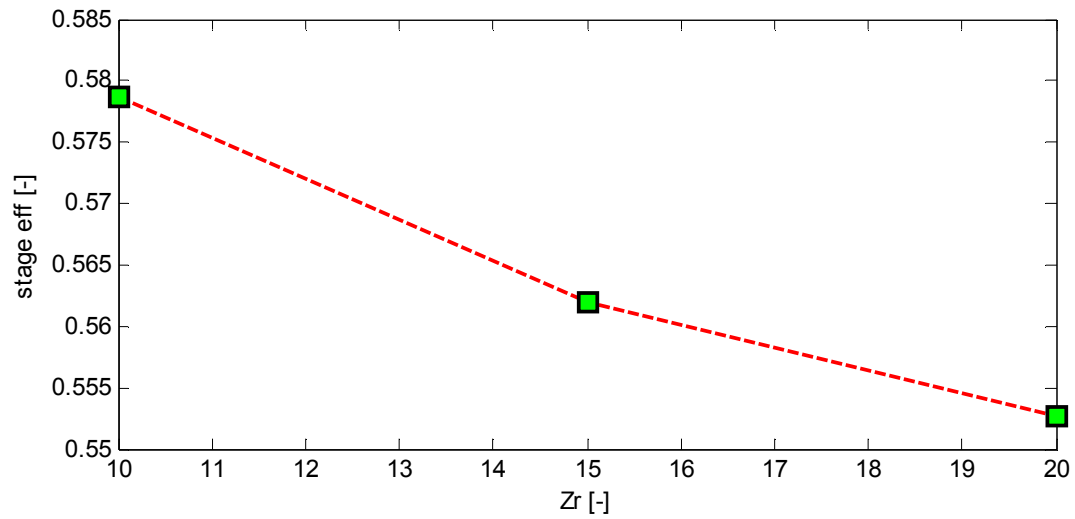


Figure 5-26 Effect of Z_r on η_{stage} for $N=600,000$ rpm and $B_I=0.04$

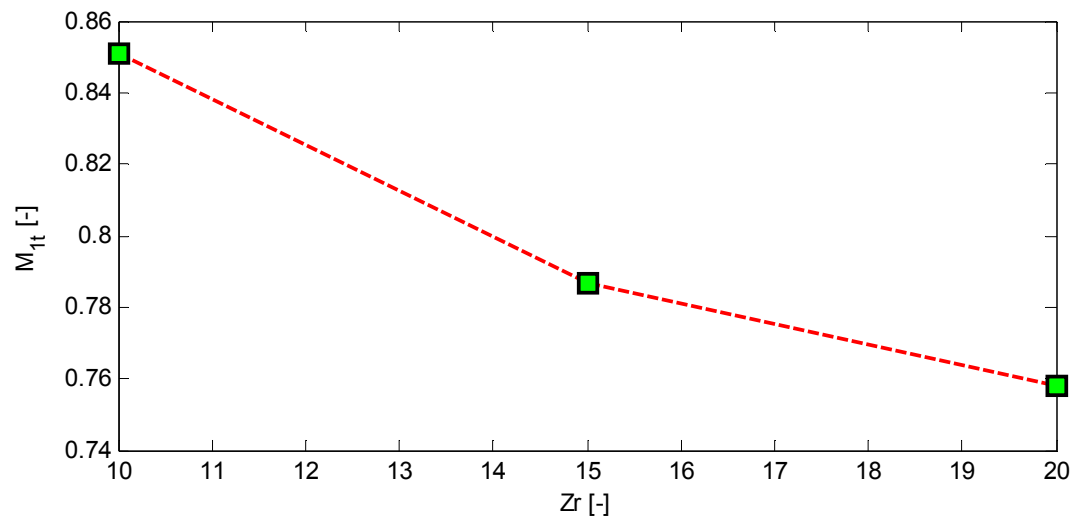


Figure 5-27 Effect of Z_r on M_{It} for $N=600,000$ rpm and $B_I=0.04$

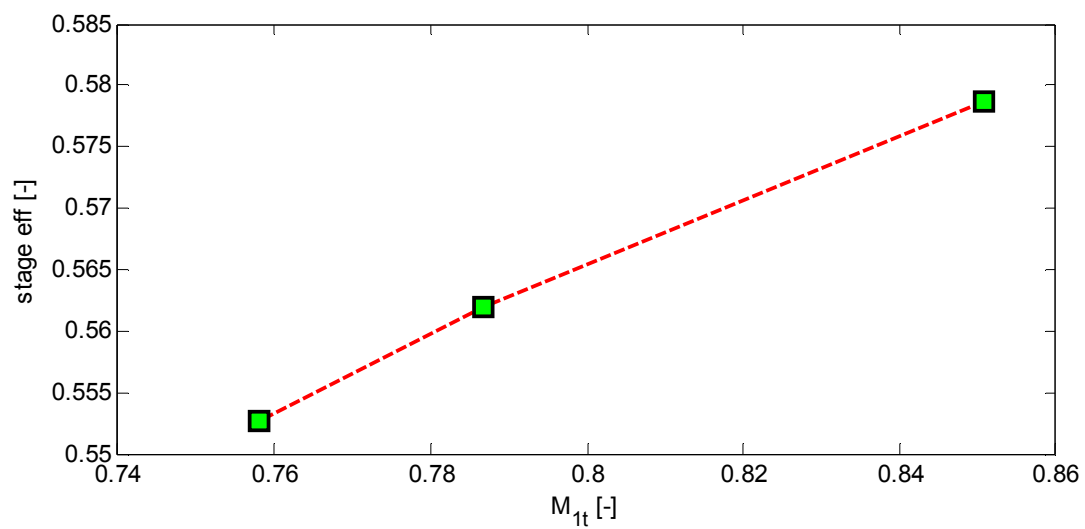


Figure 5-28 η_{stage} as function of M_{It} for $N=600,000$ rpm and $B_I=0.04$

Figure 5-20 to 5-22 , 5-23 to 5-25 and 5-26 to 5-28 illustrates the same trends as follows:

- increasing number of blades has a negative effect on the stage efficiency,
- increasing number of blades has a positive effect on the tip Mach number,
- increasing tip Mach number has a positive effect on the stage efficiency.

Furthermore, increasing the number of blades causes less slip, which leads to increased velocities at the exit. This gives an increased work input and a higher pressure at the impeller exit. Besides, the static pressure increases less than the total pressure due to the increased flow velocities, but the overall effect is still positive on the total to static pressure ratio. At the same time the increasing blade number cause more blade blockage, a narrower passage and more surface area. The higher flow velocities and surface area result in higher drag losses.

As a consequence of drag losses, the different increase in total pressure and static pressure, the efficiency decreases when the number of blades increases (Figure 5-18).

Chapter 6 Final design

In this chapter the best results from the previous chapter will be evaluated in order to reach to the final optimization of the compressor.

The chapter starts with the comparison of design data taken from literature and work of different authors.

Then, results with influence of blade angle and number of blades for $B_I=0.02$, 0.03 and 0.04 will be compared.

Futhermore the design with the highest η_{stage} has been chosen and the blade number Z_r and backsweep angle β_{B2} has been used to calculate η_{stage} with another blockage factor B_I .

At the end the final design will be chosen.

6.1. Optimized design for $N=600,000$ rpm

Table VI-1 shows optimization results for $B_I=0.02$, 0.03 and 0.04 without taking the influence of blade angle and blade number into consideration.

Furthermore, the optimization calculations could end here, but it has been decided to continue the search for the optimum by applying the influence of blade angle and blade number.

Besides, this could lead to a better solution concerning η_{stage} .

Table VI-1 shows the optimized designs without the influence of blade number Z_r and exit blade angle β_{B2} .

B₁ [-]	0.02	0.03	0.04
DR [-]	1.6701	1.6891	1.6861
r_{1h} [m]	0.0018	0.0020	0.0022
D_{1h} [m]	0.0036	0.0040	0.0044
(M_{1t})_{min} [-]	0.8071	0.8309	0.8509
r_{1t} [m]	0.0036	0.0038	0.0039
D_{1t} [m]	0.0072	0.0076	0.0078
r_{1t}/r_{1h} [-]	2.0213	1.8333	1.7279
β_1 [°]	56.8148	58.2242	58.8919
A_{fl} [m²]	3.1601e-5	3.2166e-5	3.2382e-5
U_{1t} [m/s]	229.3032	239.8759	247.3574
W_{1t} [m/s]	273.9893	282.1687	288.9036
C_{ml}=C₁ [m/s]	149.9673	148.5891	149.2635
Z_r [-]	11	9	10
β_{B2} [°]	-42	-39	-45
η_{stage} [-]	0.5679	0.5724	0.5787
b₂ [m]	7.6148e-4	7.6689e-4	7.7575e-4
r₂ [m]	0.0067	0.0067	0.0067
U₂ [m/s]	417.8814	419.0308	420.5193
W₂ [m/s]	164.0509	167.0541	171.3450
M₂ [-]	0.7462	0.7387	0.7278

Table VI-1. Optimization results for $B_1=0.02, 0.03, 0.04$ without taking the influence of blade angle and blade number into consideration

In section 5.2 it has been described that the best maximum efficiency is reached at a rotational speed of *600,000 rpm*: as a consequence, this value has been used for the optimization calculations.

Furthermore calculations from another author [33] resulted in a compressor high-speed radial compressor, which spins at *600,000 rpm*, an impeller hub radius and an impeller tip radius of 2 mm and 3.93 mm resp.

The final configuration selected shows a blade number 15 at a backsweep angle of 40. The pressure ratio is equal to 2 and the mass flow is 0.005 kg/s: they are chosen as design values.

The Belgian PowerMEMS [53] has developed a micro gas turbine with compressor and turbine impellers of 20 mm and will produce a power output of about 1000 W. The system (Figure 6-1) basically consists of a compressor, regenerator, combustion chamber, turbine and electrical generator.

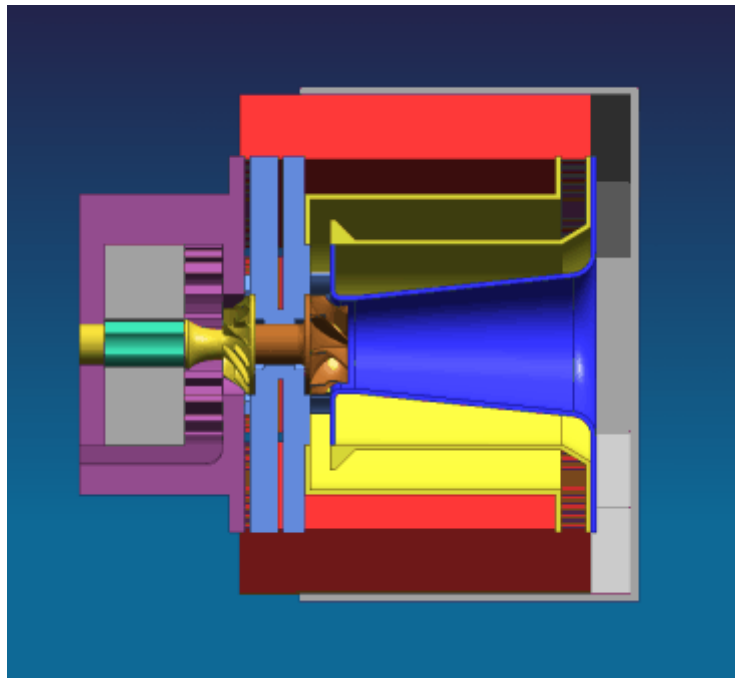


Figure 6-1 Gas turbine generator layout [52]

Furthermore the rotational speed is set to $500,000$ rpm and the compressor can achieve a pressure ratio of 3. The mass flow rate at the design point is 0.002 kg/s.

By comparing the values found on previous works with the values presented in Table VI-1, the results are in the same order of magnitude.

6.2. Optimized design for $N=600,000$ rpm for several exit blade angles and number of blades

In another work [33], papers from different authors have been used to compare results for the exit and inlet geometry.

Anyway in this report the work from the same authors [33] and an author concerning blade passages will be used.

Now by using the results from these authors the results obtained by the 1D model can be validated.

Furthermore another important consideration is the shape of the blade passages. The side elevation (r_2/r_{1t}) found in good pump practice are shown in Figure 6-2.

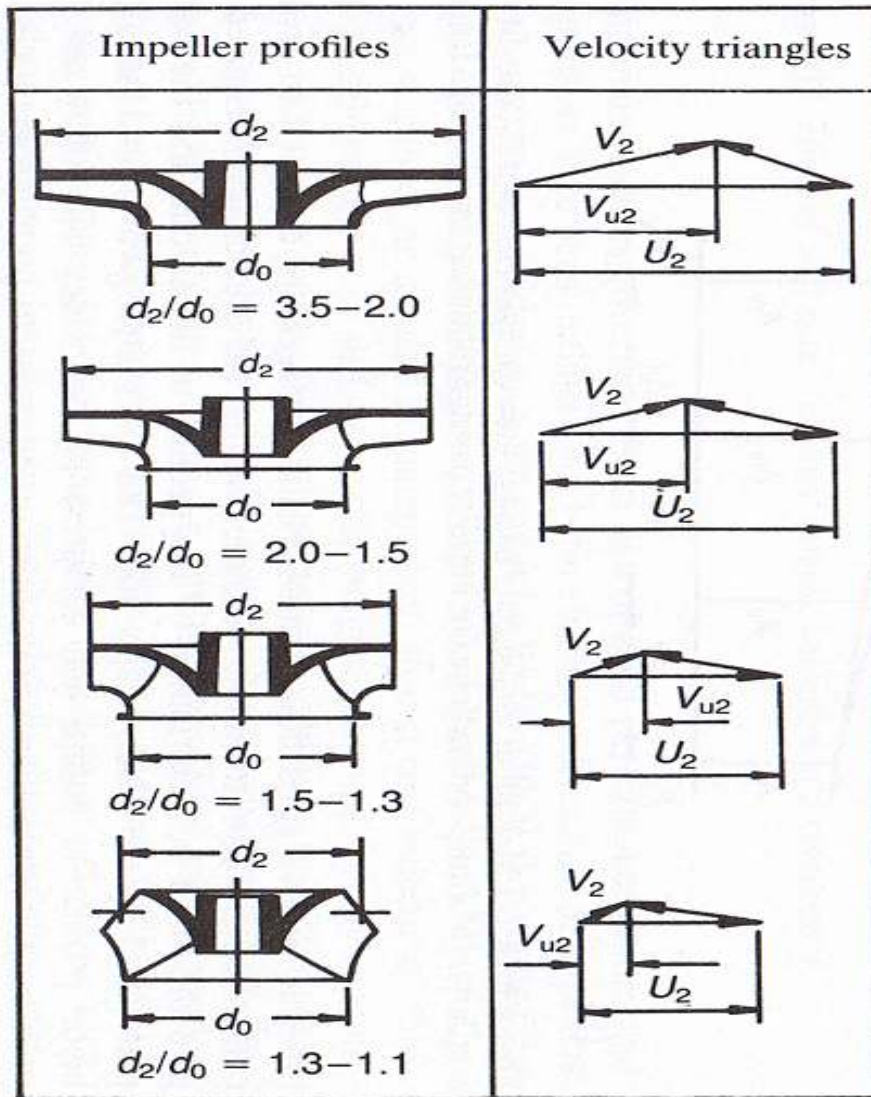


Figure 6-2 Pump side elevations [39]

Although the impeller profiles shown in Figure 6-2 are pump profiles, they can also be used to determine the shape of a compressor impeller.

Anyway it can be noticed that the range of r_2/r_{1t} is between 1.1 and 3.5.

By following what have been mentioned in section 6.1 about a better result for the η_{stage} , only tables V-21 to V-26 will be considered and the summarized results for $B_l = 0.02, 0.03$ and 0.04 will be presented in new tables.

According to the range of r_2/r_{1t} values presented in Figure 6-2, all designs presented in table V-22 to V-27 can be accepted.

Table VI-2 shows the summarized optimization calculation results, regarding the effects of blade number (Z_r) and exit blade angle (β_{B2}) for $N=600,000 \text{ rpm}$ and $B_I=0.02$.

Effects	β_{B2}			Z_r		
DR (original)	1.67	1.67	1.67	1.67	1.67	1.67
DR [-]	1.6700	1.6701	1.6700	1.6700	1.6701	1.6854
r_{1h} [m]	0.0021	0.0018	0.0016	0.0020	0.0018	0.0012
D_{1h} [m]	0.0042	0.0036	0.0032	0.0040	0.0036	0.0024
$(M_{1t})_{\min}$ [-]	0.8274	0.8071	0.7938	0.8213	0.8071	0.7725
r_{1t} [m]	0.0038	0.0036	0.0036	0.0038	0.0036	0.0035
D_{1t} [m]	0.0076	0.0072	0.0072	0.0076	0.0072	0.0070
r_{1h}/r_{1t} [-]	1.8349	2.0213	2.1856	1.8808	2.0213	2.5905
β_1 [$^\circ$]	58.5934	56.8148	56.6403	58.5324	56.8148	56.6682
A_{fl} [m ²]	3.2206e-5	3.1601e-5	3.1898e-5	3.2349e-5	3.1601e-5	3.2604e-5
U_{1t} [m/s]	239.9363	229.3032	225.1620	238.0604	229.3032	219.4223
W_{1t} [m/s]	281.1234	273.9893	269.5791	279.1072	273.9893	262.6230
$C_{m1}=C_1$ [m/s]	146.4955	149.9673	148.2397	145.6986	149.9673	144.3076
Zr [-]	11	11	11	10	11	15
β_{B2} [$^\circ$]	-45	-42	-40	-42	-42	-42
η_{stage} [-]	0.5743	0.5679	0.5639	0.5725	0.5679	0.5549
b_2 [m]	7.6972e-4	7.6148e-4	7.5591e-4	7.6736e-4	7.6148e-4	7.4433e-4
r_2 [m]	0.0067	0.0067	0.0066	0.0067	0.0067	0.0066
U_2 [m/s]	419.4505	417.8814	416.9948	419.0128	417.8814	415.0573
W_2 [m/s]	168.3373	164.0509	161.4244	167.1300	164.0509	155.8215
M_2 [-]	0.7353	0.7462	0.7533	0.7384	0.7462	0.7687
Φ [-]	0.0541	0.0543	0.0561	0.0542	0.0543	0.0564
r_2/r_{1t} [-]	1.7631	1.8611	1.8333	1.7631	1.8611	1.8857

Table VI-2. Summarized optimization results for $N=600,000 \text{ rpm}$ for $B_I=0.02$

Table VI-3 shows the summarized optimization calculation results, regarding the effects of blade number (Z_r) and exit blade angle (β_{B2}) for $N=600,000$ rpm and $B_I=0.03$.

Effects	β_{B2}				Z_r	
DR (original)	1.67	1.67	1.67	1.67	1.67	1.67
DR [-]	1.6700	1.6910	1.6891	1.6702	1.6891	1.6730
r_{1h} [m]	0.0024	0.0021	0.0020	0.0013	0.0020	0.0015
D_{1h} [m]	0.0048	0.0042	0.0040	0.0026	0.0040	0.0030
$(M_{1t})_{min}$ [-]	0.8609	0.8358	0.8309	0.7710	0.8309	0.7970
r_{1t} [m]	0.0040	0.0040	0.0038	0.0035	0.0038	0.0040
D_{1t} [m]	0.0080	0.0080	0.0076	0.0070	0.0076	0.0080
r_{1h}/r_{1t} [-]	1.6612	1.8044	1.8333	2.7167	1.8333	2.5692
β_1 [$^\circ$]	59.3758	60.7813	58.2242	56.6313	58.2242	66.8975
A_{fl} [m ²]	3.2109e-5	3.3999e-5	3.2166e-5	3.2966e-5	3.2166e-5	4.2614e-5
U_{1t} [m/s]	251.5558	248.3247	239.8759	218.9046	239.8759	251.2186
W_{1t} [m/s]	292.3275	284.5272	282.1687	262.1147	282.1687	273.1218
$C_{m1}=C_l$ [m/s]	148.9129	138.8905	148.5891	144.1698	148.5891	107.1668
Z_r [-]	9	9	9	9	9	10
β_{B2} [$^\circ$]	-45	-40	-39	-30	-39	-39
η_{stage} [-]	0.5839	0.5742	0.5724	0.5567	0.5724	0.5667
b_2 [m]	7.8239e-4	7.6982e-4	7.6689e-4	7.4674e-4	7.6689e-4	7.5964e-4
r_2 [m]	0.0067	0.0067	0.0067	0.0066	0.0067	0.0066
U_2 [m/s]	421.9489	419.3875	419.0308	415.4299	419.0308	417.6314
W_2 [m/s]	175.0465	168.2616	167.0541	156.9382	167.0541	163.2503
M_2 [-]	0.7190	0.7354	0.7387	0.7656	0.7387	0.7484
Φ [-]	0.0538	0.0542	0.05424	0.0564	0.05424	0.0561
r_2/r_{1t} [-]	1.675	1.675	1.7632	1.8857	1.7632	1.65

Table VI-3. Summarized optimization results for $N=600,000$ rpm for $B_I=0.03$

Table VI-4 shows the summarized optimization calculation results, regarding the effects of blade number (Z_r) and exit blade angle (β_{B2}) for $N=600,000$ rpm and $B_I=0.04$.

Effects	β_{B2}		Z_r		
DR (original)	1.67	1.67	1.67	1.67	1.67
DR [-]	1.6861	1.6702	1.6861	1.6700	1.6700
r_{1h} [m]	0.0022	0.0017	0.0022	0.0015	0.0010
D_{1h} [m]	0.0044	0.0034	0.0044	0.0030	0.0020
$(M_{1t})_{\min}$ [-]	0.8509	0.8025	0.8509	0.7867	0.7581
r_{1t} [m]	0.0039	0.0040	0.0039	0.0036	0.0034
D_{1t} [m]	0.0078	0.0080	0.0078	0.0072	0.0068
r_{1h}/r_{1t} [-]	1.7279	2.3804	1.7279	2.3871	3.4159
β_1 [°]	58.8919	65.7588	58.8919	57.4951	56.7850
A_{fl} [m ²]	3.2382e-5	4.1125e-5	3.2382e-5	3.3391e-5	3.3872e-5
U_{1t} [m/s]	247.3574	250.5055	247.3574	225.5907	215.7638
W_{1t} [m/s]	288.9036	274.7298	288.9036	267.4950	257.8991
$C_{ml}=C_1$ [m/s]	149.2635	112.7982	149.2635	143.7443	141.2724
Z_r [-]	10	10	10	15	20
β_{B2} [°]	-45	-40	-45	-45	-45
η_{stage} [-]	0.5787	0.5686	0.5787	0.5619	0.5526
b_2 [m]	7.7575e-4	7.6241e-4	7.7575e-4	7.5333e-4	7.4101e-4
r_2 [m]	0.0067	0.0067	0.0067	0.0066	0.0066
U_2 [m/s]	420.5193	418.0303	420.5193	416.5648	414.6396
W_2 [m/s]	171.3450	164.4911	171.3450	160.1766	154.4303
M_2 [-]	0.7278	0.7450	0.7278	0.7566	0.7729
Φ [-]	0.0541	0.0543	0.0541	0.0562	0.0564
r_2/r_{1t} [-]	1.7179	1.675	1.7179	1.8333	1.9411

Table VI-4. Summarized optimization results for $N=600,000$ rpm for $B_I=0.04$

6.3. Final overall design

For the final overall design, results for $B_I = 0.02$, 0.03 and 0.04 with the highest η_{stage} has been summarized in one table and compared to each other for the final design.

Table VI-5 shows the summarized optimization calculation results for the highest η_{stage} concerning $B_I = 0.02$, 0.03 and 0.04 .

DR (original)	1.67	1.67	1.67
DR [-]	1.6700	1.6700	1.6861
r_{1h} [m]	0.0021	0.0024	0.0022
D_{1h} [m]	0.0042	0.0048	0.0044
$(M_{1t})_{min}$ [-]	0.8274	0.8609	0.8509
r_{1t} [m]	0.0038	0.0040	0.0039
D_{1t} [m]	0.0076	0.0080	0.0078
r_{1h}/r_{1t} [-]	1.8349	1.6612	1.7279
β_1 [°]	58.5934	59.3758	58.8919
A_n [m ²]	3.2206e-5	3.2109e-5	3.2382e-5
U_{1t} [m/s]	239.9363	251.5558	247.3574
W_{1t} [m/s]	281.1234	292.3275	288.9036
$C_{m1}=C_1$ [m/s]	146.4955	148.9129	149.2635
Zr [-]	11	9	10
β_{B2} [°]	-45	-45	-45
η_{stage} [-]	0.5743	0.5839	0.5787
b_2 [m]	7.6972e-4	7.8239e-4	7.7575e-4
r_2 [m]	0.0067	0.0067	0.0067
U_2 [m/s]	419.4505	421.9489	420.5193
W_2 [m/s]	168.3373	175.0465	171.3450
M_2 [-]	0.7353	0.7190	0.7278
Φ [-]	0.0541	0.0538	0.0540
r_2/r_{1t} [-]	1.7631	1.675	1.7179

Table VI-5. Summarized optimization results for $B_I = 0.02$, 0.03 and 0.04 showing highest stage efficiencies (η_{stage})

In Table VI-5 it can be noticed that the highest η_{stage} can be found for $B_I=0.03$.

Moreover it can be noticed that the $B_I=0.03$ design have the smallest blade number compared to the other designs.

Furthermore in section 5.9.2 it can be noticed that decreasing blade number have a positive effect on η_{stage} and is the reason why $B_I=0.03$ design have the highest η_{stage} .

Anyway the same blade number Z_r and backsweep angle β_{B2} of the $B_I=0.03$ design will be used for the other blockage factors to see if there is some improvement of the η_{stage}

Table VI-6 shows the summarized optimization calculation results for different B_I , same blade number Z_r and backsweep angle β_{B2} .

B_I	0.02	0.03	0.04
DR original [-]	1.67	1.67	1.67
DR [-]	1.6700	1.6700	1.6700
r_{1h} [m]	0.0024	0.0024	0.0024
D_{1h} [m]	0.0048	0.0048	0.0048
(M_{1t}) [-]	0.8605	0.8609	0.8611
r_{1t} [m]	0.0040	0.0040	0.0040
D_{1t} [m]	0.0080	0.0080	0.0080
r_{1h}/r_{1t} [-]	0.6	0.6	0.6
β_1 [°]	59.7563	59.3758	59.1719
A_{fl} [m²]	3.2074e-005	3.2109e-005	3.2287e-005
U_{1t} [m/s]	252.5396	251.5558	251.0245
W_{1t} [m/s]	292.3281	292.3275	292.3281
$C_{m1}=C_1$ [m/s]	147.2394	148.9129	149.8078
Z_r [-]	9	9	9
β_{B2} [°]	-45	-45	-45
η_{stage} [-]	0.5839	0.5839	0.5839
b_2 [m]	7.8239e-004	7.8239e-004	7.8239e-004
r_2 [m]	0.0067	0.0067	0.0067
U_2 [m/s]	421.9489	421.9489	421.9489
W_2 [m/s]	175.0465	175.0465	175.0465
M_2 [-]	0.7190	0.7190	0.7190
Φ [-]	0.0538	0.0538	0.0538
r_2/r_{1t} [-]	1.675	1.675	1.675

Table VI-6 Final optimization results for $N=600,000$ rpm

In Table VI-6 it can be noticed that, the calculated η_{stage} is the same for all designs presented above.

Besides there is almost no effect of the blockage factors presented above on η_{stage} . Moreover an increasing flow area A_{fl} , meridional inlet velocity C_{m1} , tip radius (negligible) r_{1t} , and tip Mach number

can be noticed as the blockage factor B_I increases. The A_{fl} , C_{mI} and B_I are related through the continuity equation (see Equation 4.13). If the B_I increases, the A_{fl} and C_{mI} must increase, in order to keep the mass flow rate constant.

Although there is almost no effect of the blockage factors on η_{stage} , increasing blockage factor B_I could exceed the tip Mach number M_{It} limit of 0.9.

Furthermore, also the hub to tip radius ratio should be considered in the design of the impeller. The hub to tip ratio is the ratio of the hub radius to the tip radius.

It has been mentioned [41] that the range of hub to tip radius ratio between 0.25 and 0.45 for multistage compressors. The low and high values of hub to tip radius ratio corresponds to high and low flow coefficients respectively. The limits of flow coefficient are between 0.01 and 0.15 according to the same author [41].

The flow coefficient Φ is a dimensionless parameter, which describes the volume flow through the stage. It has been defined [42] as follows:

$$\Phi = \frac{\dot{V}}{U_2 D_2^2}$$

Although the values of hub to tip radius ratio mentioned above corresponds to multistage compressors, they could be used as reference for one stage compressors. The hub to tip radius ratio of all the designs are a little bit out of range but in the same order of magnitude.

Anyway all designs in Table VI-6 could be the final design, but in the aerodynamic point of view, the blockage factor should be kept as small as possible. For this reason the final design with an impeller hub radius equal to 2.4 mm and blockage factor of 0.02 is chosen.

Chapter 7 Conclusions and recommendations for future work

In this chapter conclusions (see section 7.1) of the work and some recommendations (see section 7.2) about further developments will be given.

7.1. Conclusions

Several optimization methods have been studied in this report and the most suitable for this assignment has been chosen. It has been concluded that GA's will be used to perform the optimization calculations.

Subsequently, the basic thermodynamics and flow phenomena involved in the design of radial compressors were discussed.

Furthermore, the impeller inlet and exit design parameters were investigated, and a one-dimensional model for a preliminary design of the geometry of radial compressors was explained.

The model was applied to design a very small radial compressor.

Moreover, the effects of the impeller hub and tip radii on the tip relative Mach number were studied.

The influence of the blade number and backsweep angle on the performance of the impeller was analyzed.

The final design of the radial compressor has rotational speed of $600,000 \text{ rpm}$ and consists of one stage. The impeller inlet and outlet design has been carried out through an optimization technique named genetic algorithms.

The purpose of the optimization was to optimize a compressor design to obtain a tip Mach number limited to 0.9 and maximum stage efficiency at once.

By analyzing the optimization results, it can be concluded that the influence of blockage factors used in this report are negligible, because of the influence of blade number Zr and exit blade angle β_{B2} .

Although larger blockage factors increases flow area A_{fl} and could exceed the tip Mach number M_{It} .

The optimization of the compressor design led to the choice of impeller hub radius $r_{lh} = 2.4 \text{ mm}$. The impeller tip radius is $r_{lt} = 4.0 \text{ mm}$. The relative flow angle is $\beta_1 = -59.76^\circ$, the relative tip mach number is $M_{It} = 0.86$ and the meridional velocity is $C_{m1} = 147.24 \text{ m/s}$.

The final configuration selected shows a blade number $Zr = 9$ at a backsweep angle $\beta_{B2} = -45^\circ$. The correlated impeller exit radius r_2 and blade height b_2 are equal to 6.7 mm and 0.78 mm respectively, while the stage efficiency $\eta_{stage} = 0.5839$.

The absolute and relative flow angles at the impeller exit are $\beta_2 = -62.93^\circ$ and $\alpha_2 = 73.33^\circ$. The Mach number at the impeller exit is $M_2 = 0.7190$ and the pressure ratio is equal to 2, which is the chosen design value.

All the geometric results of the radial compressor are summarized in Table VI-6.

7.2. Recommendations

- I. GA's has been chosen to optimize the design of the compressor. They perform random searches and are very suitable for this optimization problem.

Maybe there is a possibility to use GA's, which are a stochastic method, in combination with a deterministic method.

The recommendation is to use the GA's first and then a deterministic method and see if the optimized results can be improved

- II. After the 1D model is fully understood and optimized a 3D model must be developed and optimized to continue with the design. From the 3D model the full blading geometry must be specified and evaluated. It is essential to define all the blading (inlet guide vane, rotor, and return channel) in order to obtain the best performance for the compressor. Various geometric techniques are available to lay out the chosen blade shapes.

After trial blade shapes are completed, the next step is the three-dimensional flow analysis of the flow through the blading.

Commercial CFD codes are available and are an essential part of the design process. At the same time or sequentially, three-dimensional stress analysis can be conducted.

- III. A combination of GA's and CFD, leading to the possibility to optimize a design and see the behavior of the design at once can be performed as well.
- IV. Perform optimization calculation with the same design in this work with a diffuser design and analyze if there is some improvement concerning the performance.

Appendix A. Assignment



P&E Laboratory
Chair of Gas Turbines, Propulsion and Energy
Conversion
Leeghwaterstraat 44
2628 CA DELFT
Tel. (015)27 82179
Fax. (015) 27 82460

ASSIGNMENT

Classification: Final Assignment
Date: 16 October 2007

Name: Jairsinho Pietersz
Supervisor: Prof. Ir. J.P. van Buijtenen
Mentor: Mattia Olivero M.Sc.

Subject Optimization of the preliminary Design of a Radial Compressor

Introduction

To comply with a continuous increasing request of energy from our society, an interesting way seems to be the development of high efficiency systems which can be installed in each singular house and can produce electrical energy and heat using natural gas. A very innovative microturbine concept has been analyzed and patented by MTT b.v. for this purpose.

For that microturbine, a radial compressor without diffuser has been designed based on a 1D model ("Preliminary Design of a Radial Compressor of an Innovative Microturbine"). This thermodynamic model predicted the impeller inlet and exit design in two successive steps, each of which in turn included different optimization steps, carried Out several times one after the other. Compressor dimensions and flow properties (pressure, temperature, velocities...) have been obtained based on the optimized results of the quantities involved.

It is now of interest the investigation of optimization methods and their application to the design process of that particular radial compressor, in order to summarize the previous design in as few steps as possible, to speed the procedure up and to make It a continuous method. Particularly, the optimization activity is focused on the minimum tip relative Mach number for the inlet and on the stage efficiency – to reach the desired pressure ratio - for the outlet.

Objective

- Review of the optimization techniques
- Understanding of the proper formulation of an optimization task
- Choosing of the most suitable technique taking into account different aspects (complexity of the implementation, CPU cost.)
- Analyze the sensitivity of different engine configuration to different figures of merit.
- Study of the design constrains for a compressor on this small scale

Activities

The following steps are foreseen:

1. Survey of the literature available for the optimization techniques
2. Survey of the possible applications to the design of compressors
3. Development/use of a Matlab code to analyze and optimize a 1 D design process
4. Application of the selected optimization techniques

Report

The final results of this study have to be reported in an English document, including a copy of this assignment and a summary of the purpose and results of the research.

Supervisor,

Mentor,

Prof. Ir. J.P. van Buijtenen

Mattia Olivero M.Sc.

Appendix B. Software Used

The following software has been used for the outcome of this graduation thesis:

Adobe Acrobat Reader 8.0

Matlab R2007a/b

MathType 5.2c

Microsoft PowerPoint

Microsoft Word

Microsoft Excel

Paint

Appendix C. Optimization Method Examples

In this appendix some examples of optimization methods will be given.

The appendix starts with the following derivative-free deterministic methods:

- Hooke and Jeeves method;
- Simplex method;
- Two-Phase Method;
- Dual Approach.

Then the appendix presents an example of the following gradient-based deterministic methods:

- Lagrange's Method;
- Quadratic programming;
- Sequential Quadratic Programming.

The appendix ends with an example of the following stochastic methods:

- Genetic algorithm;
- Simulated annealing.

C1. Hooke and Jeeves method

The objective of this problem is to find the minimum of function $f(x_1, x_2) = x_1^2 x_2 + x_1 x_2^2 - 3x_1 x_2$,

starting at the point $x_0 = (2, 2)$ with a stopping criteria of $\epsilon = \frac{1}{8}$ and a step size of $h_0 = (0.5, 0.5)$.

SOLUTION

Success is denoted by S and failure by F.

Step 1

Using the initial base point $x_0 = (2, 2)$, first a local exploration is conducted in x_1 , keeping x_2 constant by evaluating $f(2, 2)$, $f(x_1 + h, x_2)$ and $f(x_1 - h, x_2)$:

$$f(2, 2) = 4.0,$$

$$f(2.5, 2) = 7.5, \quad (F)$$

$$f(1.5, 2) = 1.5. \quad (S)$$

Since, the result of $f(2.5, 2)$ is higher than $f(2, 2)$ and the result of $f(1.5, 2)$ is lower than $f(2, 2)$, the temporary base point $x'_0 = (1.5, 2)$. Next a local exploration can be conducted in x_2 by evaluating $f(x_1, x_2 + h)$ and $f(x_1, x_2 - h)$:

$$f(1.5, 2.5) = 3.75, \quad (F)$$

$$f(1.5, 1.5) = 0.00. \quad (S)$$

Since the result of $f(1.5, 1.5)$ is lower than $f(1.5, 2)$, the new base point $\mathbf{x}_1 = (1.5, 1.5)$.

Step 2

Now a pattern search can be conducted. The pattern point is determined by $\mathbf{x}_{p,1} = 2\mathbf{x}_1 - \mathbf{x}_0 = (1, 1)$.

Evaluating $f(\mathbf{x}_{p,1})$, that is $f(1.5, 2)$ gives -1. Since, the result of $f(\mathbf{x}_{p,1})$ is lower than $f(\mathbf{x}_1)$ a pattern move can be conducted to $\mathbf{x}_{p,1}$. For the next cycle of iteration $\mathbf{x}_2 = \mathbf{x}_{p,1} = (1, 1)$.

Furthermore a local exploration can be conducted in $\mathbf{x}_1$ starting at a base point $\mathbf{x}_2$:

$$f(1, 1) = -1.0,$$

$$f(1.5, 1) = -0.7499, \quad (F)$$

$$f(0.5, 1) = -0.75. \quad (F)$$

Both $f(1.5, 1)$ and $f(0.5, 1)$ are greater than $f(1, 1)$, therefore there is no improvement in the function. Return to the base point $\mathbf{x}_2$ and explore in $\mathbf{x}_2$:

$$f(1, 1.5) = -0.7499, \quad (F)$$

$$f(1, 0.5) = -0.75. \quad (F)$$

Again there is no improvement on the function, and the step size is greater than the desired accuracy.

The step size must be reduced and the exploration repeated from the base point $\mathbf{x}_2$

The new step size is given as follows:

$$\mathbf{h}_2 = 0.5$$

$$\mathbf{h}_0 = (0.25, 0.25).$$

Conducting local exploration in $\mathbf{x}_1$ gives:

$$f(1, 1) = -1.0,$$

$$f(1.25, 1) = -0.9375, \quad (F)$$

$$f(0.75, 1) = -0.9375, \quad (F)$$

and local exploration in $\mathbf{x}_2$ results in:

$$f(1, 1.25) = -0.9375, \quad (F)$$

$$f(1, 0.75) = -0.9375. \quad (F)$$

Exploration results in failure in both $\mathbf{x}_1$ and $\mathbf{x}_2$, so the step size must be reduced by half again and the local exploration repeated. Exploring in $\mathbf{x}_1$ with a step size of 0.125 results in:

$$f(1, 1) = -1.0,$$

$$f(1.125,1) = -0.98447, (F)$$

$$f(0.875,1) = -0.9844. (F)$$

Exploring in x_2 gives the following result:

$$f(1,1.125) = -0.9844, (F)$$

$$f(1,0.875) = -0.9844. (F)$$

Once more failure results, because no improvement is realized in the function, and the step size is equal to the desired tolerance.

The search terminates, giving the following results:

$$x_1 = 1.0,$$

$$x_2 = 1.0,$$

$$f_{\min} = -1.0.$$

C2. Simplex method

The objective is to maximize the following equation:

$$Z = 5x_1 + 4x_2 + 3x_3,$$

which is subjected to the following constraints:

$$2x_1 + 3x_2 + x_3 \leq 5,$$

$$4x_1 + x_2 + 2x_3 \leq 11,$$

$$3x_1 + 4x_2 + 2x_3 \leq 8,$$

$$x_1, x_2, x_3 \geq 0.$$

This problem has a feasible origin. Slack variables x_4, x_5 and x_6 are introduced to give the following first tableau:

$$x_4 = 5 - 2x_1 - 3x_2 - x_3 \geq 0 \mid x_1 \leq \frac{5}{2}(s),$$

$$x_5 = 11 - 4x_1 - x_2 - 2x_3 \geq 0 \mid x_1 \leq \frac{11}{4},$$

$$x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \geq 0 \mid x_1 \leq \frac{8}{3},$$

$$Z = 5x_1 + 4x_2 + 3x_3.$$

Here $5 > 0$ and $5 > 4 > 3$. So x_1 is chosen as incoming variable. To find the outgoing variable the constraints on all entries (see right side) has been calculated. The strictest (s) is given by the first entry, and thus the outgoing variable is x_4 .

The first entry in the next tableau is:

$$x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4.$$

Replacing this expression for x_1 in all other entries and simplifying to get the standard format leads to the second tableau:

$$x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \geq 0 \mid x_3 \leq 5,$$

$$x_5 = 1 + 5x_2 + 2x_4 \geq 0 \mid \text{no bound because } x_3 \text{ is not present in this equation,}$$

$$x_6 = \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \geq 0 \mid x_3 \leq 1(s),$$

$$Z = \frac{25}{5} - \frac{7}{2}x_2 + \frac{1}{3}x_3 - \frac{5}{2}x_4.$$

This completes the first iteration. For the next step it is clear that x_3 is the incoming variable and consequently the outgoing variable is x_6 . The first entry for the next tableau is thus $x_3 = 1 + x_2 + 3x_4 - 2x_6$ (third entry in previous tableau).

Replacing this expression for x_3 in all other entries and simplifying to get the standard format leads to the third tableau:

$$x_3 = 1 + x_2 + 3x_4 - 2x_6 \geq 0,$$

$$x_1 = 2 - 2x_2 - 2x_4 + x_6 \geq 0,$$

$$x_5 = 1 + 5x_2 + 2x_4 \geq 0,$$

$$Z = 13 - 3x_2 - x_4 - x_6.$$

In the last entry, all the coefficients of the nonbasic variables are negative.

Consequently it is not possible to obtain a further increase in Z by increasing one of the nonbasic variables. The optimal value of Z is thus 13 with $x_1^* = 2, x_2^* = 0, x_3^* = 1$.

C3. Two-Phase Method

The objective is to maximize the following equation:

$$3x_1 + 4x_2 - 3,$$

which is subjected to the following constraints:

$$x_1 + 3x_2 \geq 6, \quad (1)$$

$$-x_1 + 3x_2 = 3, \quad (2)$$

$$3x_1 - x_2 \leq 6, \quad (3)$$

$$x \geq 0.$$

The problem is first transformed into the standard form by introducing a surplus variable (a variable necessary to convert an inequality " $\geq$ " into an equality equation) in Equation (1), artificial variables in Equations (1) and (2) and a slack variable in equation (3). The problem has been transformed to maximize the following equation:

$$3x_1 + 4x_2 - 3,$$

which is subjected to the following constraints:

$$x_1 + 3x_2 - x_3 + x_4 = 6, \quad (4)$$

$$-x_1 + 3x_2 + x_5 = 3, \quad (5)$$

$$3x_1 - x_2 + x_6 = 6, \quad (6)$$

$$x \geq 0.$$

Phase I Solution

The problem is now to minimize the sum of x_4 and x_5 , which are the artificial variables, that is maximize $-(x_4 + x_5)$. From Equations (4) and (5) the objective function can be expressed in terms of other variables to maximize the following equation:

$$P = 6x_2 - x_3 - 9,$$

which is subjected to equations (4) through (6).

The initial tableau and subsequent iterations are shown from Tables C.III-1 to C.III-3. Phase 1 is now complete with the following results:

$$x_1 = x_2 = 1.5,$$

$$x_6 = 3,$$

$$x_3 = x_4 = x_5 = 0.$$

The optimum for phase 1 was achieved, because the objective function is zero. Also x_4 and x_5 have both been driven to zero.

X_1	X_2	X_3	X_4	X_5	X_6	Solution
1	3	-1	1	0	0	6
-1	3	0	0	1	0	3
3	-1	0	0	0	1	6
0	-6	1	0	0	0	-9

Table C.III-1. Initial tableau (phase 1)

X_1	X_2	X_3	X_4	X_5	X_6	Solution
2	0	-1	1	-1	0	3
$-\frac{1}{3}$	1	0	0	$\frac{1}{3}$	0	1
$\frac{8}{3}$	0	0	0	$\frac{1}{3}$	1	7
-2	0	1	0	2	0	-3

Table C.III-2. Iteration 1 (phase 1)

X_1	X_2	X_3	X_4	X_5	X_6	Solution
1	0	$-\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	0	1.5
0	1	$-\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	0	1.5
0	0	$\frac{4}{3}$	$-\frac{4}{3}$	$\frac{5}{3}$	1	3
0	0	0	1	1	0	0

Table C.III-3. Iteration 2 (phase 1)

Phase II Solution

Phase II begins with dropping x_4 and x_5 . The objective function of the original problem is now included in the objective function row of simplex tableau. The iterations for phase II are given in Tables 4 and 5. The final tableau demonstrates that an optimum has been achieved because there are

no negative elements in the objective function row. The optimum is then $x_1 = \frac{21}{8}$, $x_2 = \frac{15}{8}$ and the objective function at the optimum is $\frac{99}{8}$.

X_1	X_2	X_3	X_6	Solution
1	0	$-\frac{1}{2}$	0	1
0	1	$-\frac{1}{6}$	0	1
0	0	$\frac{4}{3}$	1	3
-3	-4	0	0	-3

Table C.III-4. Iteration 1 (phase 2)

X_1	X_2	X_3	X_6	Solution
1	0	0	$\frac{3}{8}$	$\frac{21}{8}$
0	1	0	$\frac{1}{8}$	$\frac{15}{8}$
0	0	1	$\frac{3}{4}$	$\frac{9}{4}$
0	0	0	$\frac{11}{8}$	$\frac{99}{8}$

Table C.III-5. Final Iteration (phase 2)

C4. Dual Approach

The objective is to minimize the following equation:

$$F = 3x_1 - x_2 + 4x_3,$$

which is subjected to the following constraints:

$$2x_1 + 2x_2 - x_3 \geq 7,$$

$$x_1 - 2x_2 + 3x_3 \geq 4,$$

$$-x_1 - 4x_2 + x_3 \leq -6.$$

Solution

The last constraint is converted to standard form by multiplying by -1 and the dual problem transforms in the following equation:

$$P = 7z_1 + 4z_2 + 6z_3,$$

The objective is to minimize the above equation which is subjected to the following transformed constraints:

$$2z_1 + z_2 + z_3 \leq 3,$$

$$2z_1 + z_2 + 4z_3 \leq -1,$$

$$-z_1 + 3z_2 + z_3 \leq 4.$$

After the transformation above the dual problem can be solved using the simplex algorithm.

C5. Lagrange's Method

The objective is to minimize the following equation:

$$f(x) = (x_1 - 2)^2 + (x_2 - 2)^2,$$

which is subjected to the following constraint:

$$h(x) = x_1 + x_2 = 6.$$

The Lagrangian is formulated as follows:

$$L(x, \lambda) = (x_1 - 2)^2 + (x_2 - 2)^2 + \lambda(x_1 + x_2 - 6),$$

then Karush-Kuhn-Tucker conditions for the constraint minimum are as follows:

$$\frac{\partial L}{\partial x_1} = 2(x_1 - 2) + \lambda = 0,$$

$$\frac{\partial L}{\partial x_2} = 2(x_2 - 2) + \lambda = 0,$$

$$\frac{\partial L}{\partial \lambda} = x_1 + x_2 - 6 = 0.$$

Solving these equations gives a candidate point:

$$x_1^* = 3, x_2^* = 3, \lambda^* = -2 \text{ with } f(x^*) = 2$$

This solution is also depicted in Figure C.IV-1.

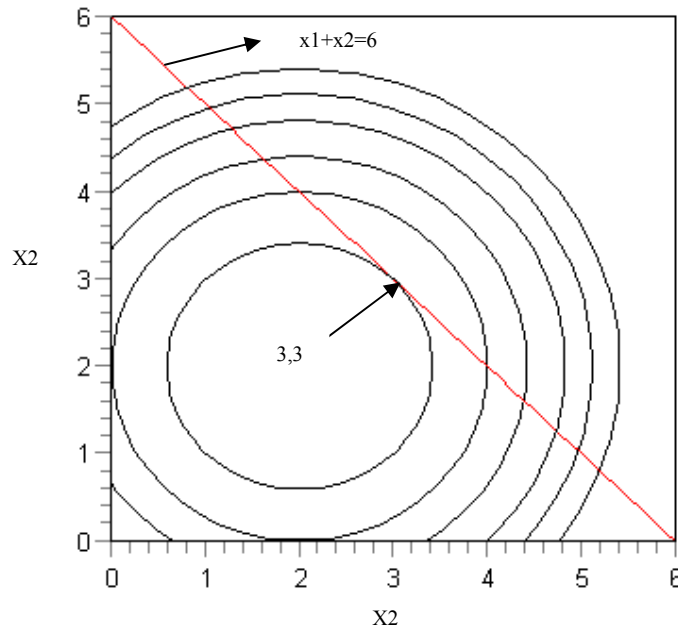


Figure C.V-1 Graphical solution to the example

C6. Quadratic programming

The objective is to minimize the following equation:

$$f(x) = x_1^2 + x_2^2 + x_3^2 ,$$

which is subjected to the following constraints:

$$h_1(x) = x_1 + x_2 + x_3 = 0 ,$$

$$h_2(x) = x_1 + 2x_2 + 3x_3 - 1 = 0 .$$

For the equality constraint problem the solution is obtained via the lagrangian method with:

$$L(x, \lambda) = x_1^2 + x_2^2 + x_3^2 + \lambda_1(x_1 + x_2 + x_3) + \lambda_2(x_1 + 2x_2 + 3x_3 - 1).$$

The necessary conditions for a minimum give:

$$\frac{\partial L}{\partial x_1} = 2x_1 + \lambda_1 + \lambda_2 ,$$

$$x_1 = -\frac{1}{2}(\lambda_1 + \lambda_2) ,$$

$$\frac{\partial L}{\partial x_2} = 2x_2 + \lambda_1 + 2\lambda_2 ,$$

$$x_2 = -\frac{1}{2}(\lambda_1 + 2\lambda_2) ,$$

$$\frac{\partial L}{\partial x_3} = 2x_3 + \lambda_1 + 3\lambda_2 ,$$

$$x_3 = -\frac{1}{2}(\lambda_1 + 3\lambda_2) .$$

Substituting the results obtained above in the equality constraints gives the following equation:

$$-\left(\frac{\lambda_1 + \lambda_2}{2} + \frac{\lambda_1 + 2\lambda_2}{2} + \frac{\lambda_1 + 3\lambda_2}{2}\right) = 0,$$

resulting in $\lambda_1 + 2\lambda_2 = 0$ and $\frac{1}{2}(\lambda_1 + \lambda_2) + (\lambda_1 + 2\lambda_2) + \frac{3}{2}(\lambda_1 + 3\lambda_2) = -1$ resulting in $3\lambda_1 + 7\lambda_2 = 1$.

Combining the two solutions and solving for the λ 's gives:

$$\lambda_2 = -1 \text{ and } \lambda_1 = 2.$$

The candidate solution is therefore:

$$\begin{aligned} x_1^* &= -\frac{1}{2}, \\ x_2^* &= 0, \\ x_3^* &= \frac{1}{2}. \end{aligned}$$

For further analysis:

$$f(x + \Delta x) = f(x) + \nabla^T f \Delta x + \frac{1}{2} \Delta x^T H \Delta x;$$

where $\nabla f = (2x_1, 2x_2, 2x_3)^T$ and thus $\nabla f(x^*) = [-1, 0, 1]^T$ and (Hessian matrix)

$$H = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \text{ is positive definite.}$$

For changes consistent with the constraints:

$$\begin{aligned} 0 = \Delta h_1 &= \nabla^T h_1 \Delta x + \frac{1}{2} \Delta x^T \nabla^2 h_1 \Delta x, \text{ with } \nabla h_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}; \\ 0 = \Delta h_2 &= \nabla^T h_2 \Delta x + \frac{1}{2} \Delta x^T \nabla^2 h_2 \Delta x, \text{ with } \nabla h_2 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}. \end{aligned}$$

It follows that $\Delta x_1 + \Delta x_2 + \Delta x_3 = 0$ and $\Delta x_1 + 2\Delta x_2 + 3\Delta x_3 = 0$ giving

$$-\Delta x_1 + \Delta x_3 = 0 \text{ and thus}$$

$$\Delta f(x^*) = -\Delta x_1 + \Delta x_3 + \frac{1}{2} \Delta x^T H \Delta x = \frac{1}{2} \Delta x^T H \Delta x \geq 0$$

The candidate point x^* above is therefore a constraint minimum.

C7.Sequential Quadratic Programming

The objective is to minimize the following equation:

$$4x_1^2 + x_1 - x_2 - 2.5,$$

which is subjected to the following constraints:

$$x_2^2 - 1.5x_1^2 + 2x_1 - 1 \geq 0,$$

$$x_2^2 + x_1^2 - 2x_1 - 4.25 = 0.$$

The first constraint may be written as the following equation:

$$-x_2^2 + 1.5x_1^2 - 2x_1 + 1 \leq 0.$$

With a starting point of (0.5, 1), the solutions are as follows:

$$\nabla f^T = [8x_1 + 1 \quad -1] = [5 \quad -1],$$

$$\nabla g^T = [3x_1 - 2 \quad -2x_2] = [-0.5 \quad -2],$$

$$\nabla h^T = [4x_1 - 2 \quad 2x_2] = [0 \quad 2],$$

$$g(0.5, 1) = -\frac{5}{8},$$

$$h(0.5, 1) = -3\frac{3}{4}.$$

The quadratic subproblem is, the minimization of the following equation:

$$5d_1 - d_2 + 0.5(d_1^2 + d_2^2),$$

which is subjected to the following constraints:

$$2d_2 \leq 3\frac{3}{4},$$

$$-0.5d_1 - 2d_2 \leq \frac{5}{8}.$$

Iteration	d	x
1	(0, 0.933)	(0.5, 1.933)
2	(0, 0.262)	(0.5, 2.195)

Table C.VII-1. Iteration results

Solving the quadratic problem by any method $d = (0, 0.933)$, $f(d) = 2.127$ is obtained. The new value of $x = (0.5, 1.933)$. the quadratic subproblem is formed using this new value of x . The minimum is at (0.5, 2.195) for a minimum function value of -3.195. Result of the iteration is given in the table above.

C8. Genetic algorithm

The whole process is illustrated by an example [16]. A simulation of a genetic algorithm for function optimization is runned. To begin with the simulation it is assumed that the population size $pop-size = 20$ and the probabilities of genetic operators are $p_c = 0.25$ and $p_m = 0.01$

The objective is to maximize the following equation:

$$f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2),$$

where $-3 \leq x_1 \leq 12.1$ and $4.1 \leq x_2 \leq 5.8$.

The graph of function f is given in figure C.VII-1.

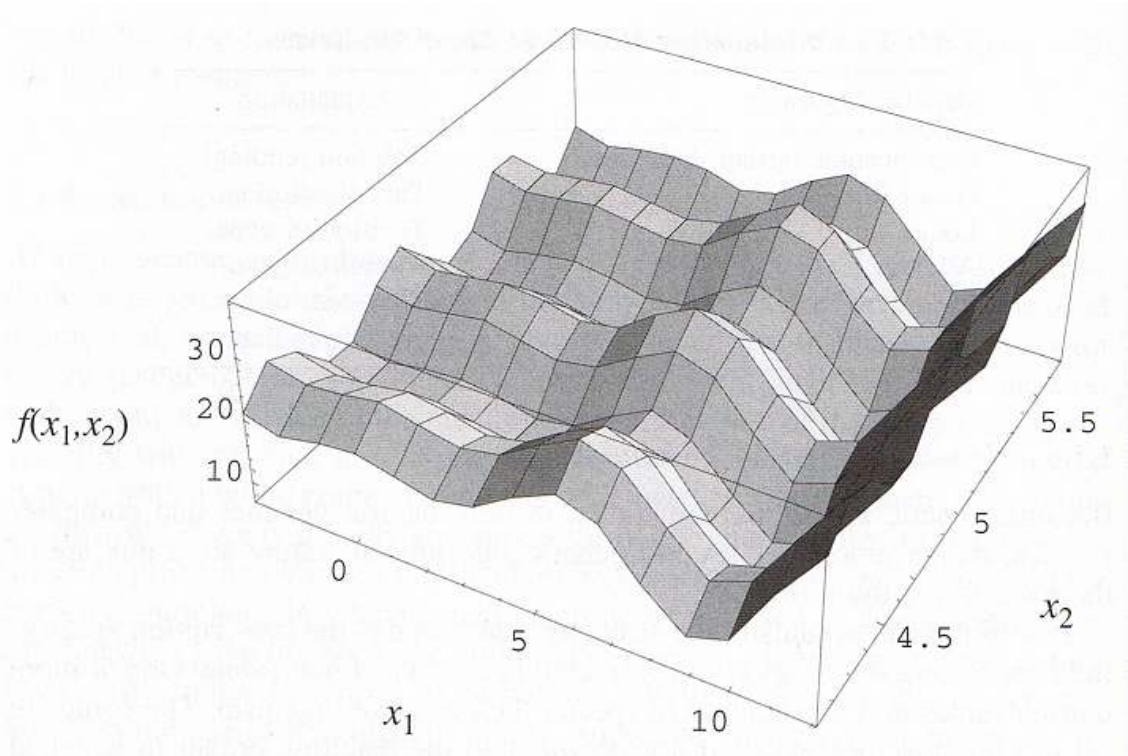


Figure C.VIII-1. Graph of function $f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$ [40]

Further on a precision of four decimal places for each variable is assumed. The domain of variable x_1 has length 15.1; the precision requirement implies that the range $[-3.0, 12.1]$ should be divided into at least $15.1 * 10000$ equal size ranges. This means that 18 bits are required as the first part of the chromosome:

$$2^{17} < 151000 \leq 2^{18}.$$

The domain of variable x_2 has length 1.7; the precision requirement implies that the range $(4.1, 5.8)$ should be divided into at least $1.7 * 10000$ equal size ranges. This means that 15 bits are required as the second part of the chromosome:

$$2^{14} < 17000 \leq 2^{15}.$$

The total length of a chromosome (solution vector) is then $m = 18 + 15 = 33$ bits, which first 18 bits code x_1 and remaining 15 bits (19-33) code x_2 .

For example the following chromosome is considered for this problem:

(01000100101101000 111110010100010).

The first 18 bits:

010001001011010000,

represent $x_1 = -3.0 + \text{decimal}(010001001011010000_2) * \frac{12.1 - (-3.0)}{2^{18} - 1} =$

$$-3.0 + 70352 * \frac{15.1}{262143} = -3.0 + 4.052426 = 1.052426.$$

The next 15 bits:

111110010100010,

represent $x_2 = 4.1 + \text{decimal}(111110010100010_2) * \frac{5.8 - 4.1}{2^{15} - 1} =$

$$4.1 + 31906 * \frac{1.7}{32767} = 4.1 + 1.655330 = 5.755330.$$

So the chromosome:

(010001001011010000111110010100010),

corresponds to $(x_1, x_2) = (1.052426, 5.755330)$. The fitness value for this chromosome is $f(1.052426, 5.755330) = 20.252640$.

To optimize the function f using a genetic algorithm, a population of 20 (*popsize_size*) chromosomes has been created. All 33 bits in all chromosomes are initialized randomly.

Here it has been assumed that after the initialization process the following population is obtained:

$$v_1 = (100110100000001111111010011011111);$$

$$v_2 = (111000100100110111001010100011010);$$

$$v_3 = (000010000011001000001010111011101);$$

$$v_4 = (100011000101101001111000001110010);$$

$$v_5 = (000111011001010011010111111000101);$$

$$v_6 = (00010100001001010100101011111011);$$

$$v_7 = (00100010000011010111101101111011);$$

$$v_8 = (100001100001110100010110101100111);$$

$$v_9 = (010000000101100010110000001111100);$$

$$v_{10} = (000001111000110000011010000111011);$$

$$v_{11} = (011001111110110101100001101111000);$$

$$v_{12} = (110100010111101101000101010000000);$$

$$v_{13} = (111011111010001000110000001000110);$$

$$v_{14} = (010010011000001010100111100101001);$$

$$\begin{aligned}
v_{15} &= (111011101101110000100011111011110); \\
v_{16} &= (110011110000011111100001101001011); \\
v_{17} &= (011010111111001111010001101111101); \\
v_{18} &= (011101000000001110100111110101101); \\
v_{19} &= (00010101001111111110000110001100); \\
v_{20} &= (101110010110011110011000101111110).
\end{aligned}$$

During the evaluation phase, each chromosome has been decoded and the fitness function values calculated from (x_1, x_2) values just decoded.

The results are as follows:

$$\begin{aligned}
eval(v_1) &= f(6.084492, 5.652242) = 26.019600 ; \\
eval(v_2) &= f(10.348434, 4.380264) = 7.580015 ; \\
eval(v_3) &= f(-2.516603, 4.390381) = 19.526329 ; \\
eval(v_4) &= f(5.278638, 5.593460) = 17.406725 ; \\
eval(v_5) &= f(-1.255173, 4.734458) = 25.341160 ; \\
eval(v_6) &= f(-1.811725, 4.391937) = 18.100417 ; \\
eval(v_7) &= f(-0.991471, 5.680258) = 16.020812 ; \\
eval(v_8) &= f(4.910618, 4.703018) = 17.959701 ; \\
eval(v_9) &= f(0.795406, 5.381472) = 16.127799 ; \\
eval(v_{10}) &= f(-2.554851, 4.793707) = 21.278435 ; \\
eval(v_{11}) &= f(3.130078, 4.996097) = 23.410669 ; \\
eval(v_{12}) &= f(9.356179, 4.239457) = 15.011619 ; \\
eval(v_{13}) &= f(11.134646, 5.378671) = 27.316702 ; \\
eval(v_{14}) &= f(1.335944, 5.151378) = 19.876294 ; \\
eval(v_{15}) &= f(11.089025, 5.054515) = 30.060205 ; \\
eval(v_{16}) &= f(9.211598, 4.993762) = 23.867227 ; \\
eval(v_{17}) &= f(3.367514, 4.571343) = 13.696165 ; \\
eval(v_{18}) &= f(3.843020, 5.158226) = 15.414128 ; \\
eval(v_{19}) &= f(-1.746635, 5.395584) = 20.095903 ; \\
eval(v_{20}) &= f(7.935998, 4.757338) = 13.666916.
\end{aligned}$$

It is clear that chromosome v_{15} is the strongest one and chromosome v_2 is the weakest.

Now the system constructs a roulette wheel for the selection process. The total fitness of the population is as follows:

$$F = \sum_{i=1}^{20} eval(v_i) = 387.776822.$$

The probability of a selection p_i for each chromosome v_i ($i=1, \dots, 20$) is:

$$\begin{aligned} p_1 &= eval(v_1) / F = 0.067099 ; \\ p_2 &= eval(v_2) / F = 0.019547 ; \\ p_3 &= eval(v_3) / F = 0.050355 ; \\ p_4 &= eval(v_4) / F = 0.044889 ; \\ p_5 &= eval(v_5) / F = 0.065350 ; \\ p_6 &= eval(v_6) / F = 0.046677 ; \\ p_7 &= eval(v_7) / F = 0.041315 ; \\ p_8 &= eval(v_8) / F = 0.046315 ; \\ p_9 &= eval(v_9) / F = 0.041590 ; \\ p_{10} &= eval(v_{10}) / F = 0.054873 ; \\ p_{11} &= eval(v_{11}) / F = 0.060372 ; \\ p_{12} &= eval(v_{12}) / F = 0.038712 ; \\ p_{13} &= eval(v_{13}) / F = 0.070444 ; \\ p_{14} &= eval(v_{14}) / F = 0.051257 ; \\ p_{15} &= eval(v_{15}) / F = 0.077519 ; \\ p_{16} &= eval(v_{16}) / F = 0.061549 ; \\ p_{17} &= eval(v_{17}) / F = 0.035320 ; \\ p_{18} &= eval(v_{18}) / F = 0.039750 ; \\ p_{19} &= eval(v_{19}) / F = 0.051823 ; \\ p_{20} &= eval(v_{20}) / F = 0.035244. \end{aligned}$$

The cumulative probabilities q_i for each chromosome v_i ($i=1, \dots, 20$) are:

$$\begin{aligned} q_1 &= 0.067099; \\ q_2 &= 0.086647; \\ q_3 &= 0.137001; \end{aligned}$$

$q_4 = 0.181890;$
 $q_5 = 0.247240;$
 $q_6 = 0.293917;$
 $q_7 = 0.335232;$
 $q_8 = 0.381546;$
 $q_9 = 0.423137;$
 $q_{10} = 0.478009;$
 $q_{11} = 0.538381;$
 $q_{12} = 0.577093;$
 $q_{13} = 0.647537;$
 $q_{14} = 0.698794;$
 $q_{15} = 0.776314;$
 $q_{16} = 0.837863;$
 $q_{17} = 0.873182;$
 $q_{18} = 0.912932;$
 $q_{19} = 0.964756;$
 $q_{20} = 1.000000.$

Now the roulette wheel can be spun 20 times, each time a single chromosome for a new population is selected. Than it is assumed that a (random) sequence of 20 numbers from the range (0..1) is generated.

The following sequence of random numbers is assumed:

0.513870	0.175741	0.308652	0.534534	0.947628;
0.171736	0.702231	0.226431	0.494773	0.424720;
0.703899	0.389647	0.277226	0.368071	0.983437;
0.005398	0.765682	0.646473	0.767139	0.780237.

The first number $r = 0.513870$ is greater than q_{10} and smaller than q_{11} , meaning the chromosome v_{11} is selected for the new population. The second number $r = 0.175741$ is greater than q_3 and smaller than q_4 , meaning the chromosome v_4 is selected for the new population, etc.

Finally, the new population consists of the following chromosomes:

$$\begin{aligned}
v'_1 &= (011001111110110101100001101111000) (v_{11}); \\
v'_2 &= (100011000101101001111000001110010) (v_4); \\
v'_3 &= (001000100000110101111011011111011) (v_7); \\
v'_4 &= (011001111110110101100001101111000) (v_{11}); \\
v'_5 &= (000101010011111111110000110001100) (v_{19}); \\
v'_6 &= (100011000101101001111000001110010) (v_4); \\
v'_7 &= (111011101101110000100011111011110) (v_{15}); \\
v'_8 &= (000111011001010011010111111000101) (v_5); \\
v'_9 &= (011001111110110101100001101111000) (v_{11}); \\
v'_{10} &= (000010000011001000001010111011101) (v_3); \\
v'_{11} &= (111011101101110000100011111011110) (v_{15}); \\
v'_{12} &= (010000000101100010110000001111100) (v_9); \\
v'_{13} &= (00010100001001010100101011111011) (v_6); \\
v'_{14} &= (100001100001110100010110101100111) (v_8); \\
v'_{15} &= (101110010110011110011000101111110) (v_{20}); \\
v'_{16} &= (100110100000001111111010011011111) (v_1); \\
v'_{17} &= (000001111000110000011010000111011) (v_{10}); \\
v'_{18} &= (111011111010001000110000001000110) (v_{13}); \\
v'_{19} &= (111011101101110000100011111011110) (v_{15}); \\
v'_{20} &= (110011110000011111100001101001011) (v_{16});
\end{aligned}$$

Now the recombination operator, crossover can be applied to the individuals in the new population (vectors v'_i). The probability of crossover $p_c = 0.25$, so (on average) 25% of chromosomes (i.e., 5 out of 20) is expected to undergo crossover.

Then, for each chromosome in the (new) population a random number r from the range (0..1) is generated. If $r < 0.25$ a given chromosome is selected for crossover.

The following sequence of random numbers is assumed:

0.822951	0.151932	0.625477	0.314685	0.346901;
0.917204	0.519760	0.401154	0.606758	0.785402;
0.031523	0.869921	0.166525	0.674520	0.758400;
0.581893	0.389248	0.200232	0.355635	0.826927.

This means that the chromosomes v'_2 , v'_{11} , v'_{13} and v'_8 were selected for crossover. It can be noted that the number of selected chromosomes is even, so they can be paired easily. If the number of selected chromosomes were odd, either one extra chromosome would be added or one selected chromosome would be removed (this choice is made randomly as well).

Now selected chromosomes can be mated randomly. For example, the first two (v'_2 and v'_{11}) and the next two (v'_{13} and v'_{18}) are coupled together. For each of these two pairs, a random integer number pos from the range (1..32) (33 is the total length — number of bits — in a chromosome) is generated. The number pos indicates the position of the crossing point.

The first pair of chromosomes is as follows:

$$v'_2 = (100011000 \mid 101101001111000001110010);$$

$$v'_{11} = (111011101 \mid 10111000010001111011110);$$

and the generated number $pos = 9$. These chromosomes are cut after the 9th bit and replaced by a pair of their offspring:

$$v''_2 = (100011000 \mid 10111000010001111011110);$$

$$v''_{11} = (111011101 \mid 101101001111000001110010).$$

The second pair of chromosomes is as follows:

$$v'_{13} = (00010100001001010100 \mid 1010111111011);$$

$$v'_{18} = (11101111101000100011 \mid 0000001000110);$$

and the generated number $pos = 20$. These chromosomes are replaced by a pair of their offspring:

$$v''_{13} = (00010100001001010100 \mid 0000001000110);$$

$$v''_{18} = (11101111101000100011 \mid 1010111111011).$$

The current version of the population is as follows:

$$v'_1 = (011001111110110101100001101111000);$$

$$v''_2 = (100011000 \mid 10111000010001111011110);$$

$$v'_3 = (00100010000011010111101101111011);$$

$$v'_4 = (011001111110110101100001101111000);$$

$$v'_5 = (000101010011111111110000110001100);$$

$$\begin{aligned}
v'_6 &= (100011000101101001111000001110010); \\
v'_7 &= (111011101101110000100011111011110); \\
v'_8 &= (000111011001010011010111111000101); \\
v'_9 &= (011001111110110101100001101111000); \\
v'_{10} &= (000010000011001000001010111011101); \\
v''_{11} &= (111011101 | 101101001111000001110010); \\
v'_{12} &= (010000000101100010110000001111100); \\
v''_{13} &= (00010100001001010100 | 0000001000110); \\
v'_{14} &= (100001100001110100010110101100111); \\
v'_{15} &= (101110010110011110011000101111110); \\
v'_{16} &= (100110100000001111111010011011111); \\
v'_{17} &= (000001111000110000011010000111011); \\
v''_{18} &= (11101111101000100011 | 1010111111011); \\
v'_{19} &= (111011101101110000100011111011110); \\
v'_{20} &= (110011110000011111100001101001011).
\end{aligned}$$

Next operator, mutation is performed on a bit—by—bit basis. The probability of mutation $p_m = 0.01$, so (on average) 1% of bits is expected to undergo mutation. There are $m \times \text{pop-size} = 33 \times 20 = 660$ bits in the whole population, so (on average) 6.6 mutations are expected per generation. Every bit has an equal chance to be mutated, so for every bit in the population, a random number r from the range (0..1) is generated. If $r < 0.01$, the bit will be mutated.

This means that 660 random numbers must be generated. In a sample run, 5 of these numbers were smaller than 0.01, the bit number and the random number are listed below in Table C8-1:

Bit position	Random number
112	0.000213
349	0.009945
418	0.008809
429	0.005425
602	0.002836

TableC.VIII-1. Bit position and random number.

The following table translates the bit position into chromosome number and the bit number within the chromosome:

Bit position	Chromosome number	Bit number within chromosome
112	4	13
349	11	19
418	13	22
429	13	33
602	19	8

Table C.VIII-2. Bit position, chromosome number and bit number within chromosome.

This means that four chromosomes are affected by the mutation operator. One of the chromosomes (the 13th) has two bits changed.

The final population is listed below and the mutated bits are in boldface.

The population is listed as new vectors v_i :

$$v_1 = (011001111110110101100001101111000);$$

$$v_2 = (100011000101110000100011111011110);$$

$$v_3 = (00100010000011010111101101111011);$$

$$v_4 = (011001111110010101100001101111000);$$

$$v_5 = (000101010011111111110000110001100);$$

$$v_6 = (100011000101101001111000001110010);$$

$$v_7 = (111011101101110000100011111011110);$$

$$v_8 = (000111011001010011010111111000101);$$

$$v_9 = (011001111110110101100001101111000);$$

$$v_{10} = (000010000011001000001010111011101);$$

$$v_{11} = (111011101101101001011000001110010);$$

$$v_{12} = (010000000101100010110000001111100);$$

$$v_{13} = (000101000010010101000100001000111);$$

$$v_{14} = (100001100001110100010110101100111);$$

$$v_{15} = (101110010110011110011000101111110);$$

$$v_{16} = (100110100000001111111010011011111);$$

$$v_{17} = (000001111000110000011010000111011);$$

$$v_{18} = (11101111101000100011101011111011);$$

$$v_{19} = (111011100101110000100011111011110);$$

$$v_{20} = (110011110000011111100001101001011).$$

With this new population just one iteration (i.e., one generation) of the while loop in the genetic procedure (Figure 3-1) is completed. It is interesting to examine the results of the evaluation process of the new population. During the evaluation phase, each chromosome is decoded and the fitness function values calculated from (x_1, x_2) values just decoded.

These are the results:

$$\begin{aligned} eval(v_1) &= f(3.130078, 4.996097) = 23.410669; \\ eval(v_2) &= f(5.279042, 5.054515) = 18.201083; \\ eval(v_3) &= f(-0.991471, 5.680258) = 16.020812; \\ eval(v_4) &= f(3.128235, 4.996097) = 23.412613; \\ eval(v_5) &= f(-1.746635, 5.395584) = 20.095903; \\ eval(v_6) &= f(5.278638, 5.593460) = 17.406725; \\ eval(v_7) &= f(11.089025, 5.054515) = 30.060205; \\ eval(v_8) &= f(-1.255173, 4.734458) = 25.341160; \\ eval(v_9) &= f(3.130078, 4.996097) = 23.410669; \\ eval(v_{10}) &= f(-2.516603, 4.390381) = 19.526329; \\ eval(v_{11}) &= f(11.088621, 4.743434) = 33.351874; \\ eval(v_{12}) &= f(0.795406, 5.381472) = 16.127799; \\ eval(v_{13}) &= f(-1.811725, 4.209937) = 22.692462; \\ eval(v_{14}) &= f(4.910618, 4.703018) = 17.959701; \\ eval(v_{15}) &= f(7.935998, 4.757338) = 13.666916; \\ eval(v_{16}) &= f(6.084492, 5.652242) = 26.019600; \\ eval(v_{17}) &= f(-2.554851, 4.793707) = 21.278435; \\ eval(v_{18}) &= f(11.134646, 5.666976) = 27.591064; \\ eval(v_{19}) &= f(11.059532, 5.054515) = 27.608441; \\ eval(v_{20}) &= f(9.211598, 4.993762) = 23.867227. \end{aligned}$$

It can be noticed that the total fitness of the new population F is 447049688, which is much higher than total fitness of the previous population, 387.776822. Also, the best chromosome now (v_{11}) has a

better evaluation (33.351874) than the best chromosome (v_{15}) from the previous population (30.060205).

Now the procedure can be repeated by running the selection process again and apply the genetic operators, evaluate the next generation, etc.

After 1000 generations the population is as follows:

$$\begin{aligned}
 v_1 &= (111011110110011011100101010111011); \\
 v_2 &= (111001100110000100010101010111000); \\
 v_3 &= (111011110111011011100101010111011); \\
 v_4 &= (111001100010000110000101010111001); \\
 v_5 &= (111011110111011011100101010111011); \\
 v_6 &= (111001100110000100000100010100001); \\
 v_7 &= (110101100010010010001100010110000); \\
 v_8 &= (111101100010001010001101010010001); \\
 v_9 &= (111001100010010010001100010110001); \\
 v_{10} &= (111011110111011011100101010111011); \\
 v_{11} &= (110101100000010010001100010110000); \\
 v_{12} &= (110101100010010010001100010110001); \\
 v_{13} &= (111011110111011011100101010111011); \\
 v_{14} &= (111001100110000100000101010111011); \\
 v_{15} &= (111001101010111001010100110110001); \\
 v_{16} &= (111001100110000101000100010100001); \\
 v_{17} &= (111001100110000100000101010111011); \\
 v_{18} &= (111001100110000100000101010111001); \\
 v_{19} &= (111101100010001010001110000010001); \\
 v_{20} &= (111001100110000100000101010111001).
 \end{aligned}$$

The calculated fitness values are as follows:

$$\begin{aligned}eval(v_1) &= f(11.120940, 5.092514) = 30.298543; \\eval(v_2) &= f(10.588756, 4.667358) = 26.869724; \\eval(v_3) &= f(11.124627, 5.092514) = 30.316575; \\eval(v_4) &= f(10.574125, 4.242410) = 31.933120; \\eval(v_5) &= f(11.124627, 5.092514) = 30.316575; \\eval(v_6) &= f(10.588756, 4.214603) = 34.356125; \\eval(v_7) &= f(9.631066, 4.427881) = 35.458636; \\eval(v_8) &= f(11.518106, 4.452835) = 23.309078; \\eval(v_9) &= f(10.574816, 4.427933) = 34.393820; \\eval(v_{10}) &= f(11.124627, 5.092514) = 30.316575; \\eval(v_{11}) &= f(9.623693, 4.427881) = 35.477938; \\eval(v_{12}) &= f(9.631066, 4.427933) = 35.456066; \\eval(v_{13}) &= f(11.124627, 5.092514) = 30.316575; \\eval(v_{14}) &= f(10.588756, 4.242514) = 32.932098; \\eval(v_{15}) &= f(10.606555, 4.653714) = 30.746768; \\eval(v_{16}) &= f(10.588814, 4.214603) = 34.359545; \\eval(v_{17}) &= f(10.588756, 4.242514) = 32.932098; \\eval(v_{18}) &= f(10.588756, 4.242410) = 32.956664; \\eval(v_{19}) &= f(11.518106, 4.472757) = 19.669670; \\eval(v_{20}) &= f(10.588756, 4.242410) = 32.956664.\end{aligned}$$

However, it can be noted that in earlier generations the fitness values of some chromosomes were better than the value 35.477938 of the best chromosome after 1000 generations. For example, the best chromosome in generation 396 had value of 38.827553. This is due to the stochastic errors of sampling.

It is relatively easy to keep track of the best individual in the evolution process. It is customary (in genetic algorithm implementations) to store the best ever individual at a separate location, in that way the algorithm would report the best value found during the whole process (as opposed to the best value in the final population).

C9. Simulated annealing

A test function defining the form of a deflected corrugated spring in n dimensions is given by:

$$f = -\cos(kR) + 0.1R^2$$

where R is the radius $R = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2 + \dots + (x_n - c_n)^2}$, c is the minimum point, and k defines the periodicity nature of the corrugations. For the case $n = 2$, $c_1 = 5$, $c_2 = 5$ and $k = 5$, the Simulated annealing (SA) algorithm is used to find the minimum of f .

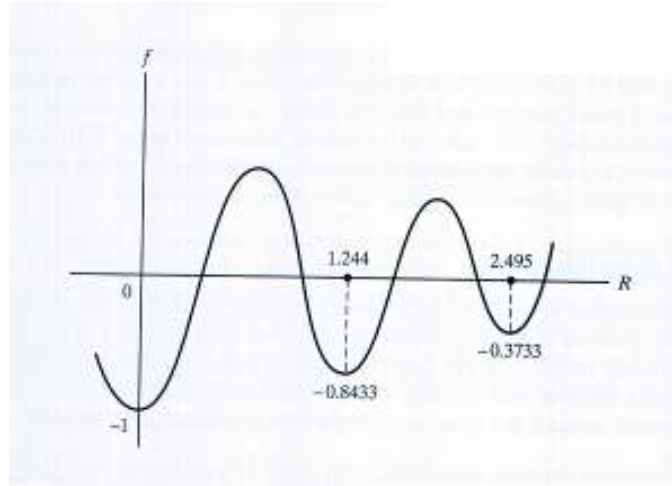


Figure C.IX-1. Function plot [11]

Appendix D. Specific speed and specific diameter

D.1 Introduction

The specific speed and specific diameter are similarity parameters, defined by [27]. They can be used together with the Mach and Reynolds numbers to give fairly complete information for the design of efficient turbomachines. The specific speed and specific diameter will be defined and examined in this appendix .

D.2 Specific speed

The specific speed has been defined by [27] in the following way:

$$N_s = \frac{N\sqrt{Q_{00}}}{H_{ad}^{\frac{3}{4}}} , \quad (D.1)$$

where N is the rotational speed expressed in revolutions per minute, Q_{00} is the volume flow expressed in ft^3/s and H_{ad} is the adiabatic head expressed in ft. The specific speed is then expressed in $\text{rpm}\cdot\text{ft}^{3/4}/\text{s}^{1/2}$.

The volume flow can be calculated in m^3/s and then converted to ft^3/s . It is usually calculated from inlet quantities for compressors. [36]

Here, the stagnation inlet density will be used to calculate the volume flow.

The adiabatic head can be calculated through the gas constant and isentropic temperature rise as follows:

$$H_{ad} = R \frac{\gamma}{\gamma - 1} T_{00} \left[\left(\frac{p_{02}}{p_{00}} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right] \quad (D.2)$$

where $R=53.34 \text{ ft}\cdot\text{lb}/\text{lbm}\cdot\text{R}$ and T_{00} is the impeller inlet total temperature in degrees Rankine.

Parallel to the original definition of the specific speed, a dimensionless specific speed also exists. It is defined by Equation D.3:

$$n_s = \frac{\omega\sqrt{Q_{00}}}{(gH_{ad})^{\frac{3}{4}}} \quad (D.3)$$

where ω is the angular speed expressed in $1/\text{s}$ and g is the gravitational acceleration expressed in ft/s^2 . The volume flow and adiabatic head are the same as defined before.

The dimensionless specific speed can be expressed in metric units as well:

$$n_s = \frac{\omega \sqrt{Q_{00}}}{\Delta h_{0,ts}^{\frac{3}{4}}} \quad (D.4)$$

where the specific enthalpy rise $\Delta h_{0,ts}$, is based on the total-to-total temperature rise.

Trazzi [2004] claims that some authors have verified the range between $95 \text{ rpm} \cdot \text{ft}^{3/4} / \text{s}^{1/2}$ and $120 \text{ rpm} \cdot \text{ft}^{3/4} / \text{s}^{1/2}$ (between $7 \text{ rpm} (\text{m}^3 / \text{s})^{0.5} / (\text{J/kg})^{0.75}$ and $9 \text{ rpm} (\text{m}^3 / \text{s})^{0.5} / (\text{J/kg})^{0.75}$ in the International System of Units).

With a rotational speed of $N=600,000 \text{ rpm}$, a mass flow rate $\dot{m} = 0.005 \text{ kg/s}$, a required pressure ratio of 2 and an impeller inlet total temperature $T_{00} = 298 \text{ K}$, specific speed $N_s = 126.3897 \text{ rpm} \cdot \text{ft}^{3/4} / \text{s}^{1/2}$ has been obtained. It can be noticed that (for $N=600,000$) it is slightly higher than the right-hand limit of the range.

Appendix E. Matlab Files

In Appendix E, four Matlab files of the one-dimensional model will be presented.

These files consists of an objective function file and a constraint function file.

Different files have been used for the one-dimensional model described in Chapter 5.

The first file has been used for preliminary inlet and outlet design of the compressor. Further more the second one has been used for preliminary inlet and outlet design with Z_r and β_{B2} as known values.

Since both files have been used for several blockage factors (0.02, 0.03, 0.04) the same equations but different input parameters, not every Matlab file will be displayed.

E1. Preliminary inlet and outlet design

Objective function

```
function eta_ts_stage = complete_fitness(x)
%Inlet
%-----
% Thermodynamic constants
g=1.4; % Isentropic coefficient (gamma) [-]
R=287; % Gas constant of air [J/kg/K]
cv=R/(g-1); % Specific heat of air at constant volume [J/kg/K]
cp=g*cv; % Specific heat of air at constant pressure [J/kg/K]
%-----
% Given thermodynamical
p00=101325; % Inlet total pressure [Pa]
T00=298; % Total inlet temperature [K]
m=0.005; % Mass flow rate [kg/s]
%-----
% Designer's choice
N=600000; % Rotational speed [rpm]
alpha1=0*pi/180; % Absolute inlet flow angle [deg]
% r_1h=1e-3; % Hub radius [m]
%-----
% Empirical
B1=0.02; % Inlet aerodynamical blockage factor [-]; from [25]
%-----
% x(1)=C_u1
% x(2)=r_1h
% Calculations
C_u1=x(1)*tan(alpha1)
C1=sqrt(C_u1.^2+x(1).^2)
T_s1=T00-(C1.^2)/(2*cp);
M1=C1./sqrt(g*R*T_s1);
p_s1=p00*((T_s1/T00).^(g/(g-1)));
RO_1=p_s1./(R*T_s1);
A_fl=m/(RO_1*x(1)*(1-B1))
r_1t=sqrt((A_fl./pi)+x(2).^2)
U_1t=2*pi*N*r_1t/60
W_1t=sqrt(x(1).^2+U_1t.^2)
M_1t=W_1t./sqrt(g*R*T_s1)
beta1=180*(atan(U_1t./x(1)))/pi
rr=r_1t/x(2)
```

```

%-----

%Outlet
%-----
% Designer's choice
% Zr=15; % Number of blades [-]
% betaB2_deg=-40; % Backsweep angle [°]
% eta_stage= 0.5504; % Stage efficiency [-]
eta_rotor=0.92; % Rotor efficiency [-]; from [26]
pr=2; % Stage pressure ratio

%Specific speed calculation
%Specific speed [rpm*(ft^0.75)/(s^0.5)]; it will be calculated with...
% ...an expression defined by [27]
RO_air=1.225; %In kg/m3
%Q_00 is the volume flow in m3/s(1m3=35.3146ft3)
%volume flow converted in ft3/s
Q_00=(m/RO_air)*35.3146
%Adiabatic head in ft
%To calculate the following equation gas constant R(J/kgK)unit must be
%converted in ft.lbf/lbm.R and the temperature T (K) unit must be converted
%in °R.
%1 J/kgK=0.33456*1/1.8ft.lbf/lbm.R and 1K=1.8°R.
R_conv=R*(0.33456*1/1.8)
T00_conv=T00*1.8
H_ad=R_conv*(g/(g-1))*T00_conv*(pr^((g-1)/g)-1)
Ns=(N*sqrt(Q_00))/(H_ad^0.75)
% Ns=105.3248;
%-----

% Calculations
% x(3)=betaB2_deg
% x(4)=Zr
% x(5)=eta_stage
betaB2=x(3)*pi/180 % Backsweep angle [rad]
L2m=6.5-0.025*Ns; % Swirl parameter
sigma=1-((sqrt(cos(betaB2)))/(x(4)^0.7)) % Wiener's slip factor
mu_2=sigma*L2m/(L2m-tan(betaB2));
Dh0_is=cp*T00*((pr^((g-1)/g))-1);
Wx=Dh0_is/x(5);
T02=T00+Wx/cp;
U2=sqrt(((U_1t*C_u1)+Wx)/mu_2)
D2=60*U2/(pi*N);
r2=D2/2
Cu2=mu_2*U2;
Cm2=Cu2/L2m;
C2=sqrt(Cu2^2+Cm2^2);
Ts2=T02-((C2^2)/(2*cp));
p02=p00*((1+(eta_rotor*Wx/(cp*T00)))^(g/(g-1)));
ps2=p02*((Ts2./T02)^(g/(g-1)));
ro_2=ps2/(R*Ts2);
M2=C2/(sqrt(g*R*Ts2))
A2=m/(Cm2*ro_2);
b2=A2/(pi*D2)
W2=sqrt((U2-Cu2)^2+(Cm2)^2)
M2_rel=W2/(sqrt(g*R*Ts2))
beta2m=(-acos(Cm2/W2))*(180/pi);
alfa2m=(atan(Cu2/Cm2))*(180/pi);

```

```

pr_ts=ps2/p00
eta_ts_stage=-(((pr_ts^((g-1)/g))-1)/((T02/T00)-1));
DR=W_1t/W2
%-----

```

Constraint function

```

function [c, ceq] =complete_constraint(x)

% Inlet
% Thermodynamic constants
g=1.4; % Isentropic coefficient (gamma) [-]
R=287; % Gas constant of air [J/kg/K]
cv=R/(g-1); % Specific heat of air at constant volume [J/kg/K]
cp=g*cv; % Specific heat of air at constant pressure [J/kg/K]
%-----
% Given thermodynamical
p00=101325; % Inlet total pressure [Pa]
T00=298; % Total inlet temperature [K]
m=0.005; % Mass flow rate [kg/s]
%-----
% Designer's choice
N=600000; % Rotational speed [rpm]
alpha1=0*pi/180; % Absolute inlet flow angle [deg]
% r_1h=1e-3; % Hub radius [m]
%-----
% Empirical
B1=0.02; % Inlet aerodynamical blockage factor [-]; from [26]
%-----
% Starting estimate
% C_m1=linspace(0,300); % Meridional velocity [m/s]; from [22]
%-----
% x(1)=C_m1
% x(2)=r_1h
% Calculations
C_u1=x(1)*tan(alpha1);
C1=sqrt(C_u1.^2+x(1).^2);
T_s1=T00-(C1.^2)/(2*cp);
M1=C1./sqrt(g*R*T_s1);
p_s1=p00*((T_s1/T00).^(g/(g-1)));
RO_1=p_s1./(R*T_s1);
A_f1=m./(RO_1.*x(1)*(1-B1));
r_1t=sqrt((A_f1./pi)+x(2).^2);
U_1t=2*pi*N*r_1t/60;
W_1t=sqrt(x(1).^2+U_1t.^2);
M_1t=W_1t./sqrt(g*R*T_s1);

% outlet
% Designer's choice
% Zr=15; % Number of blades [-]
% betaB2_deg=-40; % Backsweep angle [°]
% eta_stage= 0.1; % Stage efficiency [-]
eta_rotor=0.92; % Rotor efficiency [-]; from [26]
pr=2; % Stage pressure ratio

%Specific speed calculation

```

```

%Specific speed [rpm*(ft^0.75)/(s^0.5)]; it will be calculated with...
% ...an expression defined by Baljé (1962)
RO_air=1.225; %In kg/m3
%Q_00 is the volume flow in m3/s(1m3=35.3146ft3)
%volume flow converted in ft3/s
Q_00=(m/RO_air)*35.3146
%Adiabatic head in ft
%To calculate the following equation gas constant R(J/kgK)unit must be
%converted in ft.lbf/lbm.R and the temperature T (K) unit must be converted
%in °R.
%1 J/kgK=0.33456*1/1.8ft.lbf/lbm.R and 1K=1.8°R.
R_conv=R*(0.33456*1/1.8)
T00_conv=T00*1.8
H_ad=R_conv*(g/(g-1))*T00_conv*(pr^((g-1)/g)-1)
Ns=(N*sqrt(Q_00))/(H_ad^0.75)
% Ns=105.3248;
%-----
% Calculations
% x(3)=betaB2_deg
% x(4)=Zr
% x(5)=eta_stage
betaB2=x(3)*pi/180; % Backsweep angle [rad]
L2m=6.5-0.025*Ns; % Swirl parameter
sigma=1-((sqrt(cos(betaB2)))/(x(4)^0.7))
mu_2=sigma*L2m/(L2m-tan(betaB2));
Dh0_is=cp*T00*((pr^((g-1)/g))-1);
Wx=Dh0_is/x(5);
T02=T00+Wx/cp;
U2=sqrt(Wx/mu_2);
D2=60*U2/(pi*N);
r2=D2/2;
Cu2=mu_2.*U2;
Cm2=Cu2./L2m;
C2=sqrt(Cu2^2+Cm2^2);
Ts2=T02-((C2.^2)/(2*cp));
p02=p00.*((1+(eta_rotor.*Wx./(cp*T00))).^(g/(g-1)));
ps2=p02.*((Ts2./T02).^(g/(g-1)));
ro_2=ps2/(R*Ts2);
M2=C2/(sqrt(g*R*Ts2));
A2=m/(Cm2*ro_2);
b2=A2/(pi*D2);
W2=sqrt((U2-Cu2)^2+(Cm2)^2);

c = [-0.9+(1-((sqrt(cos(betaB2)))/(x(4)^0.7)));
-mu_2+0.8*L2m./(L2m-tan(betaB2));
-0.9+(W_1t./sqrt(g*R*T_s1))];

ceq = [-2+(ps2./p00);
-1.67+(W_1t/W2)];

```

E2. Preliminary inlet and outlet design with Z_r and β_{B2} as known values

Objective function

```
function eta_ts_stage = complete_fitness(x)
%Inlet
%-----
% Thermodynamic constants
g=1.4; % Isentropic coefficient (gamma) [-]
R=287; % Gas constant of air [J/kg/K]
cv=R/(g-1); % Specific heat of air at constant volume [J/kg/K]
cp=g*cv; % Specific heat of air at constant pressure [J/kg/K]
%-----
% Given thermodynamical
p00=101325; % Inlet total pressure [Pa]
T00=298; % Total inlet temperature [K]
m=0.005; % Mass flow rate [kg/s]
%-----
% Designer's choice
N=600000; % Rotational speed [rpm]
alpha1=0*pi/180; % Absolute inlet flow angle [deg]
% r_1h=1e-3; % Hub radius [m]
%-----
% Empirical
B1=0.02; % Inlet aerodynamical blockage factor [-]; from [26]
%-----
% Starting estimate
% C_m1=linspace(0,300); % Meridional velocity [m/s]; from [22]
%-----
% x(1)=C_m1
% x(2)=r_1h
% Calculations
C_u1=x(1)*tan(alpha1)
C1=sqrt(C_u1.^2+x(1).^2)
T_s1=T00-(C1.^2)/(2*cp);
M1=C1./(sqrt(g*R*T_s1));
p_s1=p00*((T_s1/T00).^(g/(g-1)));
RO_1=p_s1./(R*T_s1);
A_f1=m/(RO_1*x(1)*(1-B1))
r_1t=sqrt((A_f1./pi)+x(2).^2)
U_1t=2*pi*N*r_1t/60
W_1t=sqrt(x(1).^2+U_1t.^2)
M_1t=W_1t./sqrt(g*R*T_s1)
beta1=180*(atan(U_1t./x(1)))/pi
rr=r_1t/x(2)
%-----

%Outlet
%-----
% Designer's choice
Zr=9; % Number of blades [-]
betaB2_deg=-45; % Backsweep angle [°]
eta_rotor=0.92; % Rotor efficiency [-]; from [26]
pr=2; % Stage pressure ratio

%Specific speed calculation
%Specific speed [rpm*(ft^0.75)/(s^0.5)]; it will be calculated with...
```

```

% ...an expression defined by [27]
RO_air=1.225; %In kg/m3
%Q_00 is the volume flow in m3/s(1m3=35.3146ft3)
%volume flow converted in ft3/s
Q_00=(m/RO_air)*35.3146
%Adiabatic head in ft
%To calculate the following equation gas constant R(J/kgK)unit must be
%converted in ft.lbf/lbm.R and the temperature T (K) unit must be converted
%in °R.
%1 J/kgK=0.33456*1/1.8ft.lbf/lbm.R and 1K=1.8°R.
R_conv=R*(0.33456*1/1.8)
T00_conv=T00*1.8
H_ad=R_conv*(g/(g-1))*T00_conv*(pr^((g-1)/g)-1)
Ns=(N*sqrt(Q_00))/(H_ad^0.75)
%-----
% Calculations
% x(3)=eta_stage
betaB2=betaB2_deg*pi/180 % Backsweep angle [rad]
L2m=6.5-0.025*Ns; % Swirl parameter
sigma=1-((sqrt(cos(betaB2)))/(Zr^0.7)) % Wiener's slip factor
mu_2=sigma*L2m/(L2m-tan(betaB2));
Dh0_is=cp*T00*((pr^((g-1)/g))-1);
Wx=Dh0_is/x(3);
T02=T00+Wx/cp;
U2=sqrt(((U_1t*C_u1)+Wx)/mu_2)
D2=60*U2/(pi*N);
r2=D2/2
Cu2=mu_2*U2;
Cm2=Cu2/L2m;
C2=sqrt(Cu2^2+Cm2^2);
Ts2=T02-((C2^2)/(2*cp));
p02=p00*((1+(eta_rotor*Wx/(cp*T00)))^(g/(g-1)));
ps2=p02*((Ts2./T02)^(g/(g-1)));
ro_2=ps2/(R*T2);
M2=C2/(sqrt(g*R*T2))
A2=m/(Cm2*ro_2);
b2=A2/(pi*D2)
W2=sqrt((U2-Cu2)^2+(Cm2)^2)
M2_rel=W2/(sqrt(g*R*T2));
beta2m=(-acos(Cm2/W2))*(180/pi);
alfa2m=(atan(Cu2/Cm2))*(180/pi);
pr_ts=ps2/p00
eta_ts_stage=-(((pr_ts^((g-1)/g))-1)/((T02/T00)-1));
DR=W_1t/W2
%-----

```

Constraint function

```
function [c, ceq] =complete_constraint(x)

% Inlet
% Thermodynamic constants
g=1.4; % Isentropic coefficient (gamma) [-]
R=287; % Gas constant of air [J/kg/K]
cv=R/(g-1); % Specific heat of air at constant volume [J/kg/K]
cp=g*cv; % Specific heat of air at constant pressure [J/kg/K]
%-----
% Given thermodynamical
p00=101325; % Inlet total pressure [Pa]
T00=298; % Total inlet temperature [K]
m=0.005; % Mass flow rate [kg/s]
%-----
% Designer's choice
N=600000; % Rotational speed [rpm]
alpha1=0*pi/180; % Absolute inlet flow angle [deg]
% r_1h=1e-3; % Hub radius [m]
%-----
% Empirical
B1=0.02; % Inlet aerodynamical blockage factor [-]; from [26]
%-----
% Starting estimate
% C_m1=linspace(0,300); % Meridional velocity [m/s]; from [22]
%-----
% x(1)=C_m1
% x(2)=r_1h
% Calculations
C_u1=x(1)*tan(alpha1);
C1=sqrt(C_u1.^2+x(1).^2);
T_s1=T00-(C1.^2)/(2*cp);
M1=C1./sqrt(g*R*T_s1);
p_s1=p00*((T_s1/T00).^(g/(g-1)));
RO_1=p_s1./(R*T_s1);
A_f1=m./(RO_1.*x(1)*(1-B1));
r_1t=sqrt((A_f1./pi)+x(2).^2);
U_1t=2*pi*N*r_1t/60;
W_1t=sqrt(x(1).^2+U_1t.^2);
M_1t=W_1t./sqrt(g*R*T_s1);
beta1=180*(atan(U_1t./x(1)))/pi;

% outlet
% Designer's choice
Zr=9; % Number of blades [-]
betaB2_deg=-45; % Backsweep angle [°]
% eta_stage= 0.1; % Stage efficiency [-]
eta_rotor=0.92; % Rotor efficiency [-]; from [26]
pr=2; % Stage pressure ratio

%Specific speed calculation
%Specific speed [rpm*(ft^0.75)/(s^0.5)]; it will be calculated with...
% ...an expression defined by [27]
RO_air=1.225; %In kg/m3
%Q_00 is the volume flow in m3/s(1m3=35.3146ft3)
%volume flow converted in ft3/s
```

```

Q_00=(m/RO_air)*35.3146
%Adiabatic head in ft
%To calculate the following equation gas constant R(J/kgK)unit must be
%converted in ft.lbf/lbm.R and the temperature T (K) unit must be converted
%in °R.
%1 J/kgK=0.33456*1/1.8ft.lbf/lbm.R and 1K=1.8°R.
R_conv=R*(0.33456*1/1.8)
T00_conv=T00*1.8
H_ad=R_conv*(g/(g-1))*T00_conv*(pr^((g-1)/g)-1)
Ns=(N*sqrt(Q_00))/(H_ad^0.75)
%-----
% Calculations
% x(3)=eta_stage
betaB2=betaB2_deg*pi/180; % Backsweep angle [rad]
L2m=6.5-0.025*Ns; % Swirl parameter
sigma=1-((sqrt(cos(betaB2)))/(Zr^0.7))
mu_2=sigma*L2m/(L2m-tan(betaB2));
Dh0_is=cp*T00*((pr^((g-1)/g))-1);
Wx=Dh0_is/x(3);
T02=T00+Wx/cp;
U2=sqrt(Wx/mu_2);
D2=60*U2/(pi*N);
r2=D2/2
Cu2=mu_2.*U2;
Cm2=Cu2./L2m;
C2=sqrt(Cu2^2+Cm2^2);
Ts2=T02-((C2.^2)/(2*cp));
p02=p00.*((1+(eta_rotor.*Wx./(cp*T00))).^(g/(g-1)));
ps2=p02.*((Ts2./T02).^(g/(g-1)));
ro_2=ps2/(R*Ts2);
M2=C2/(sqrt(g*R*Ts2));
A2=m/(Cm2*ro_2);
b2=A2/(pi*D2)
W2=sqrt((U2-Cu2)^2+(Cm2)^2);

c = [-0.9+(1-((sqrt(cos(betaB2)))/(x(3)^0.7)));
-mu_2+0.8*L2m/(L2m-tan(betaB2));
-0.9+(W_1t./sqrt(g*R*T_s1))];

ceq = [-2+(ps2./p00);
-1.67+(W_1t/W2)];

```

References

Texts

- [1] Okechi, C and Onwubiko., (2000), *“Introduction To Engineering Design Optimization”*, Prentice-Hall, Inc., Upperside river, New Jersey
- [2] Pschenichny, B.N., (1970), *“Algorithm for general mathematical programming problems”*, Kibernetika (Russian) 5: 120 —125
- [3] Han, S.P., (1977) *“Globally convergent method for nonlinear programming”*, Journal of Optimization Theory and Applications 22: 297—309
- [4] Belegundu, A.D. and Arora, J.S., (1984), *“A recursive quadratic programming method with active set strategy for optimal design”*, International Journal for Numerical Methods in Engineering 20: 803—816
- [5] Powell, M. J. D., and Yuan, Y., (1986), *“A recursive quadratic programming algorithm that uses differentiable exact penalty function”*, Mathematical Programming 35: 265—278
- [6] Bartholomew-Biggs, M.C., (1987), *“Recursive quadratic programming methods based on the augmented Lagrangian”*, Mathematical Programming Study 31: 21—41
- [7] Davis, Lawrence., (1987), *“Genetic algorithms and simulated annealing”*, Pitman, London
- [8] Corana, A., Marchesi, M., Martini, C. and Ridella, S., (1987), *“Minimizing Multimodal Functions of Continuous Variables with the ”Simulated Annealing” Algorithm”*, ACM Trans. on Mathematical Software, Vol. 13, No. 3, pp. 262-280
- [9] S. Kirkpatrick.,(May 13, 1983), *“Optimization by Simulated Annealing”*, Science, New Series, Vol. 220, No. 4598., pp. 671-680
- [10] V. Cerny, (1985), *“ A thermodynamical approach to the travelling salesman problem”*: an efficient simulation algorithm. Journal of Optimization Theory and Applications, 45:41-51
- [11] Belegundu, A.D. and Chandrupatla, T.R., (1999), *“Optimization Concepts and Applications in Engineering”*, Prentice-Hall, Inc
- [12] Rayward-Smith, V. J., (1996), *“Modern Heuristic Search Methods”*, Wiley, Chichester; New York
- [13] Dowsland, K.A., (1995), *“Simulated Annealing In Modern Heuristic Techniques for Combinatorial Problems”*, (ed. Reeves, C.R.), McGraw-Hill
- [14] Holland, J.H., (1975), *“Adaption in Natural and Artificial Systems”*, Ann Arbor, University of Michigan Press
- [15] Rechenberg, I., (1973), *“Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution”*, Frommann-Holzboog Verlag, Stuttgart
- [16] Zbigniew Michalewicz., (1996), *“Genetic Algorithms + Data Structures = Evolution Programs”*, Third, revised and Extended edition, Springer-Verlag, Berlin Heidelberg

- [17] Goldberg, D.E., (1989), "*Genetic algorithm in Search, Optimization and Machine Learning*", Addison-Wesley, Reading, MA
- [18] de la Maza, M and Tidor, B., (1992), "*Increases flexibility in genetic algorithm: The use of variable Boltzmann selective pressure to control propagation*" New ORCA CSTS conference: Computer science and operation development in their interfaces, 425-440
- [19] Grefenstette, J.J and Baker, J., (1989), "*How genetic algorithms work: A critical look at implicit parallelism*", Proceedings of the third international conference on genetic algorithms, 20-27
- [20] De Jong, K.A., (1975), "*Analysis of the behavior of a class of genetic adaptive systems*", Ph.D. dissertation, Ann Arbor, University of Michigan
- [21] Hanlon, P.C., (2001) "Compressor handbook", McGraw-Hill, U.S.A.
- [22] Duiker, J., "*Design of a High Pressure Compressor of a Single Jet Engine of a General Aviation Airplane*", (March 2006), MSc Thesis, Delft University of Technology, Faculty of Aerospace Engineering
- [23] Meherwan P. Boyce, (2002), "*Gasturbine engineering handbook*", second edition, Butterworth-Heinemann, UK
- [24] "*The Technical Journal of Concepts ETI, Inc.*", (1998), Volume 7
- [25] Japikse, D., (1996), "*Centrifugal Compressor Design and Performance*", Concepts ETI Inc
- [26] Japikse, D. and Baines, N.C., (1994) "*Introduction to Turbomachinery*", Concepts ETI Inc. and Oxford University Press
- [27] Baljé, O.E., (1981), "*Turbomachines: a Guide to Design, Selection, and Theory*", John Wiley & Sons
- [28] Trazzi, P.E., (June 2004) "*Design and Development of a 70 N Thrust Class Turbojet Engine*", Engenharia Térmica (Thermal Engineering), Volume 5
- [29] Ismail, K.A.R. et al., (1998), "*Small Radial Compressors: Aerodynamic Design and Analysis*", International Journal of Rotating Machinery, Volume 4
- [30] Kang, P., Tanaka, S. and Esashi, M., (2005) "*Demonstration of a MEMS-Based Turbocharger on a Single Rotor*", Journal of Micromechanics and Microengineering, Volume 15
- [31] Kang, S. et al., (April 2006), "*Microscale Radial-Flow Compressor Impeller Made of Silicon Nitride: Manufacturing and Performance*", Journal of Engineering for Gas Turbines and Power, Volume 126
- [32] Peirs, J., Reynaerts, D. and Verplaetsen, F., (2004), "*A Microturbine for Electric Power Generation*", Sensors and Actuators, Volume 113
- [33] Olivero, M., (September 2006), "*Preliminary Design Of A Radial Compressor Of An Innovative Microturbine*", MSc Thesis, Delft University of Technology
- [34] Bonaiuti, D. et al., (2002), "*Analysis and Optimization of Transonic Centrifugal Compressor Impellers Using the Design of Experiments Technique*", Proc. of ASME

- [35] Oh, J.S., (2002), "*Investigation of Off-Design Performance of Vaned Diffusers in Centrifugal Compressors - Part II: A Low-Solidity Cascade Diffuser*", Proc. of ASME GT-2002-30388
- [36] Whitfield, A. and Baines, N.C., (1990), "*Design of Radial Turbomachines*", Longman Scientific & Technical, Harlow
- [37] Vavra, M.H., (1970), "*Basic Elements for Advanced Designs of Radial-Flow Compressors*", AGARD LS-39
- [38] Came, P.M. and Robinson, C.J., (1999), "*Centrifugal Compressors Design*", Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, Volume 213
- [39] Turton, R.K., (1995), "*Principles of turbomachinery*", Chapman & Hall
- [40] Mitsuo Gen and Runwei Cheng., (1997), "*Genetic Algorithms And Engineering Design*", John Wiley & Sons, Inc
- [41] Ludtke, K.M., (2004), "*Proces Centrifugal Compressors*", Springer-Verlag, Berlin Heidelberg
- [42] Dalbert, P. et al., (1999), "*Radial Compressor Design for Industrial Compressors*", Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, Volume 213
- [43] Harinck, J., (August 2004), "*Genetic Algorithm for Radial Turbomachinery Optimizations*", MSc Thesis, Delft University of Technology
- [44] Wilson, R. B., (1963), "A simplicial algorithm for concave programming." Ph.D. dissertation, Graduate School of Business Administration, Harvard University, Boston, Mass
- [45] Aggarwal, C.C., Orlin, J.B., and Tai, R.P., (1997), "Optimized crossover for the independent set problem", Operation research 45(2): 226-234
- [46] Peirs, J. et al., (2003) "*Development of a Micro Gas Turbine for Electronic Power Generation*", the 14th MicroMechanics Europe Workshop
- [47] Kadambi, V and Prasad, M , (1977), "*An introduction to Energy Conversion*", New Age International (P) Limited, Ansari Road, Daryaganj, New Delhi

Websites

- [48] Multi-disciplinary System Design Optimisation of a 2-Stage Axial Compressor for Aerodynamic Performance, Production Cost and Dynamic Stability
http://ocw.mit.edu/NR/rdonlyres/Aeronautics-and-Astronautics/16-888Spring-2004/B7643BD6-A5AF-47A6-A7C5-E3D2001E950F/0/perrot_dorca.pdf
- [49] A MULTI-STAGE COMPRESSOR DESIGN OPTIMIZATION USING CFD
<http://flab.eng.isas.ac.jp/member/oyama/papers/rjcf2003.pdf>

[50] Process Integration and Automated Multi-Objective Optimization Supporting Aerodynamic Compressor Design

http://opus.kobv.de/btu/volltexte/2007/48/pdf/Diss_Keskin.pdf

[51] Simulated annealing

www.cs.nott.ac.uk/~gmk/aim/notes/simulatedannealing.doc

[52] Wikipedia

www.wikipedia.com

[53] Microturbines

<http://www.powermems.be/gasturbine.html>

[54] Freepatent

<http://www.freepatentsonline.com/5145317.pdf>