11-9-2012

Design Optimization of Nozzle Shapes for Maximum Uniformity of Exit Flow

Karla K. Quintao
Florida International University, kquin020@fiu.edu

DOI: 10.25148/etd.FI12120505

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

DESIGN OPTIMIZATION OF NOZZLE SHAPES FOR MAXIMUM UNIFORMITY OF EXIT FLOW

A thesis submitted in partial fulfillment of

the requirements for the degree of

MASTER OF SCIENCE

in

MECHANICAL ENGINEERING

by

Karla Keldani Quintão

2012

To: Dean Amir Mirmiran
College of Engineering and Computing

This thesis, written by Karla Keldani Quintão, and entitled Design Optimization of Nozzles Shapes for Maximum Uniformity of Exit Flow, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this thesis and recommend that it be approved.

_____
Yiding Cao

_____
Cheng-Xian (Charlie) Lin

_____
George S. Dulikravich, Major Professor

Date of Defense: November 9, 2012

The thesis of Karla Keldani Quintão is approved.

_____
Dean Amir Mirmiran
College of Engineering and Computing

_____
Dean Lakshmi N. Reddi
University Graduate School

Florida International University, 2012

DEDICATION

I dedicate this thesis to my loving grandmother, who will always be in my heart and to my loving parents for their support in all the moments of my life. I owe them everything I have accomplished.

ACKNOWLEDGMENTS

I would like to thank Prof. George Dulikravich for his support during my whole Master`s program. I am profoundly grateful for all his help and dedication in teaching and guiding me always in the kindest way.

In addition, I would like to thank Prof. Yiding Cao and Prof. Cheng-Xian Lin for their advices as members of my MSc. Committee, and Prof. Cesar Levy, Prof. Andres Tremante and the Mechanical Engineering Department, for my assistantship because without it, I would not have had the opportunity to do this work.

I would like to thank my student friends who greatly helped me in my research, especially Ahmadreza Abbasi, Ricardo Gasparini and Agerneh Dagnew. I will be forever grateful.

Last, but not least, I would like to thank all the great people that I have met at FIU. Robel, Maryam Zahedi, Majid, Leon, Juan, Maryam Asghari, Abas, Kamran. All of you will be always remembered!

ABSTRACT OF THE THESIS

DESIGN OPTIMIZATION OF NOZZLE SHAPES

FOR MAXIMUM UNIFORMITY OF EXIT FLOW

by

Karla Keldani Quintão

Florida International University, 2012

Miami, Florida

Professor George S. Dulikravich, Major Professor

The objective of this study is to identify the optimal designs of converging-diverging supersonic and hypersonic nozzles that perform at maximum uniformity of thermodynamic and flow-field properties with respect to their average values at the nozzle exit.

Since this is a multi-objective design optimization problem, the design variables used are parameters defining the shape of the nozzle. This work presents how variation of such parameters can influence the nozzle exit flow non-uniformities.

A Computational Fluid Dynamics (CFD) software package, ANSYS FLUENT, was used to simulate the compressible, viscous gas flow-field in forty nozzle shapes, including the heat transfer analysis. The results of two turbulence models, k-ε and k-ω, were computed and compared.

With the analysis results obtained, the Response Surface Methodology (RSM) was applied for the purpose of performing a multi-objective optimization. The optimization was performed with ModeFrontier software package using Kriging and Radial Basis Functions (RBF) response surfaces. Final Pareto optimal nozzle shapes were then analyzed with ANSYS FLUENT to confirm the accuracy of the optimization process.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE PAGE

LIST OF TABLES

TABLE PAGE

CHAPTER I

INTRODUCTION

1.1. PROBLEM STATEMENT

Flow of gases through a converging-diverging nozzle is one of the benchmark problems used for modeling compressible flow using computational fluid dynamics algorithms.

In a converging nozzle, the highest speed that a fluid can be accelerated to is sonic speed, which occurs at the exit. The converging – diverging nozzles are used to accelerate the fluid to supersonic speeds past the throat of such a nozzle. In this case, depending on the ratio of the average exit pressure to the inlet stagnation pressure, there is a possibility of creating shock waves in the flow-field.

Exit flow from a converging-diverging nozzle often has strong gradients of pressure, temperature, density, and speed in radial and axial direction. These non-uniformities at the nozzle exit are the result of the non-uniformities in the flow-field entering the nozzle. The causes of the nozzle inlet flow non-uniformities could be as follows: inhomogeneous combustion in the combustion chamber upstream of the nozzle, injection of cooling gas tangentially or radially through the slots on the combustion chamber wall, pre-swirl that exists in the combustion chamber created by the compressor upstream of the combustion chamber, flow separation behind fuel injector and flame holders in the combustion chamber, among others.

In many practical applications, non-uniformities in the nozzle exit flow are unacceptable. An example is the use of a converging-diverging nozzle to accelerate gas flow to a hypersonic speed in order to simulate actual conditions when a hypersonic missile moves through a uniform atmosphere. In the existing high enthalpy hypersonic testing facilities the gas is heated using powerful electric arcs.

The gas heated by the electric arcs is extremely hot (up to 15,000K) in the area of the arc which is along the axis of a heating chamber. This temperature drops off to approximately 1000K close to the wall of the heating chamber. Thus, the gas has extreme radial gradient of temperature and other thermodynamic and flow-field properties as it enters the converging-diverging nozzle.

Another example is the air entering a gas turbine. The conditions within gas turbines are extreme. The pressure can be as high as 40 bar and the temperature more than 1000 K. The most extreme conditions are found in the high pressure part downstream of the combustion chamber where hot combustion gases flow through a cascade of rotors and stators. A great radial gradient of temperature increases thermal stresses on stator blades which may damage them. Because of the high rotational speeds, this can result in a rupture of the entire turbine.

The question is: how to mix this flow so that it exits the nozzle with a minimum possible radial gradient of each of these properties? This is the main motivation for this thesis.

1.2. RESEARCH OBJECTIVE

This problem presents a minimization of variations (positive or negative) of thermodynamic and flow-field property with respect to their average values at the nozzle exit. The question is: what can be used as design variables (parameters) that can be varied in order to influence the nozzle exit flow non-uniformities?

The design variables evaluated are parameters defining the shape of the converging-diverging nozzle. The parameters to be varied are: the distance between the inlet and the throat, the distance between the throat and the exit, the inlet radius, the exit radius, the inlet angle of the wall and the exit angle of the wall. The throat radius is considered constant.

The objective is to find the optimal designs that present the minimum variation of temperature, density and Mach number at the nozzle exit.

1.3. METHODOLOGY

In order to create the nozzles shapes, a FORTRAN code was used to read the design variables inputs. It uses a fifth order polynomial to find the values of nozzle radius along the axis.

A computational fluid dynamics software package was used to simulate 2D axisymmetric fluid and thermal flows through forty hypersonic nozzles. ANSYS GAMBIT was used for mesh generation and ANSYS FLUENT for flow analysis. The analyzes were made for two turbulence models. A non-uniform temperature distribution profile at the inlet was used to simulate different conditions.

With the results obtained, the optimization was performed. To significantly accelerate the entire design optimization process, response surfaces (meta-models) were generated in order to perform the virtual optimization. They are mathematical functions that replace very complicated physical models, generate correlations of experimental data and reduce the computational cost involved with the optimization process [1].

There are a lot of different methods for generating response surface models. Kriging is a semi–parametric approach that does not rely on any specific model structure, which makes it much more flexible than approaches based on parametric behavioral models. On the other hand, accurate predictions are obtained for short training sequences [2]. As this work presents only six design variables, Kriging could be applied.

A multi-objective optimization software (ModeFrontier) was used to generate the response surfaces and to perform an optimization using Multi Objective Genetic Algorithm (MOGA).

Designs evaluations with the optimal configurations were validated by real computation.

1.4. LITERATURE REVIEW

The non-uniformities in nozzle flows have been studied over the years. Some previous research in this field will be reviewed.

In 1989, Doty et al. [3] [4], showed that moderate non-uniformities in the flow properties leaving the combustor and entering a supersonic nozzle had effect on thrust produced by the nozzle. Later, Doty et al. [5] studied non-uniform profiles which were more severe and consequently had more influence on nozzle performance.

In 1991, Snelling [6] also examined the effects of non-uniform entrance flow profiles of hypersonic nozzles for scramjet-powered flight vehicles, but the studies were centered on pitching moment. A uniform and a non-uniform inlet profile were compared. The effects of non-uniformities in the flow-field were an increase in the overall vehicle thrust and a decrease in the overall vehicle moment for the inlet conditions in the single nozzle geometry used in the study. The increased thrust for the non-uniform case was due to higher compression on the nozzle wall caused by expansion waves from the nozzle wall reflecting off the non-uniformities in the flow-field as compression waves. The decreased moment for the non-uniform case was due to lower pressure on the nozzle wall than the uniform case for the aft portion of the nozzle combined with the longer moment arm canceling out the effect of the higher compression region. The lower pressure on the nozzle wall was the result of

compression waves from the nozzle wall reflecting off the non-uniformities in the flow-field as expansion waves.

In 2003, Palma et al. [7] executed some experiments in which planar laser-induced fluorescence (PLIF) of nitric oxide (NO) was used to measure vibrational and rotational temperatures. The experiments took place in a small shock-tunnel facility, which is an impulse facility that can generate the pressures and stagnation enthalpies required for simulation of hypersonic atmospheric re-entry flows [8]. Good agreement between measured rotational temperature and a non-equilibrium one-dimensional nozzle calculation was demonstrated. The measured vibrational temperatures were higher than the computed value, although they exhibited the expected vibrational freezing behavior. This disagreement was attributed to non-linearities in the imaging system and non-uniformity in the flow. The non-uniformities were assumed to be due to contamination by driver gas and were found in 32% of the images.

In 2006, O'Byrne et al. [9], in a similar experiment, used nitric oxide PLIF to visualize the flow at the exit of a hypersonic conical nozzle, to determine operating conditions that would allow more uniform nozzle flow than that of [7] and to explain the mechanism most likely to be responsible for the non-uniform flow. Two nozzle-throat inserts were fabricated: one with a converging conical end-wall, having a half-angle of 30° and the other with a flat end-wall. Possible causes for the non-uniformity were outlined and investigated and the problem was shown to be due to a small step at the nozzle throat. They postulated that the cause of the flow non-uniformity was the entrainment of cooler gas from the boundary layer into the freestream caused by flow separation at the throat. Upon modifying the nozzle throat, images were significantly more uniform and the standard deviation in average signal between tunnel runs reduced from 25% to 15%.

1.5. THESIS STRUCTURE

Chapter I introduces the study. First, the problem statement is presented. Then the research objectives and methodologies applied are explained. Some literature reviews about non-uniformities in nozzle flows are given in order to place this work in context.

Chapter II provides a brief review of the main topics of Computational Fluid Dynamics involved in this study. Governing equations, turbulence modeling, discretization methods, and grid generation are some of the covered subjects.

Chapter III presents a review of optimization, including basic concepts, main methods and Response Surfaces Methodology.

Chapter IV describes methodologies used to arrive at the solutions. The shapes and mesh generation were explained as well as the simulations in ANSYS FLUENT and the optimization using ModeFrontier.

Chapter V presents the results from the CFD simulations and the optimization. Comparisons between two turbulence models are shown. The errors between the results of real designs and virtual designs using meta-models are discussed.

Chapter VI is a summary of the thesis with concluding remarks and proposed future work.

CHAPTER II

COMPUTATIONAL FLUID DYNAMICS

CFD is dedicated to the study of fluids in motion and how the fluid flow behavior influences processes that may include heat transfer and possibly chemical reactions in combusting flows. Additionally, the physical characteristics of the fluid motion can usually be described through fundamental mathematical equations, usually in partial differential form, which govern a process of interest and are often called governing equations in CFD [10].

A complete analysis which appears in CFD codes, including ANSYS FLUENT, consists of three main elements: pre-processor, solver and post-processor. Figure 1 presents the inter-connectivity functions of these three main elements.

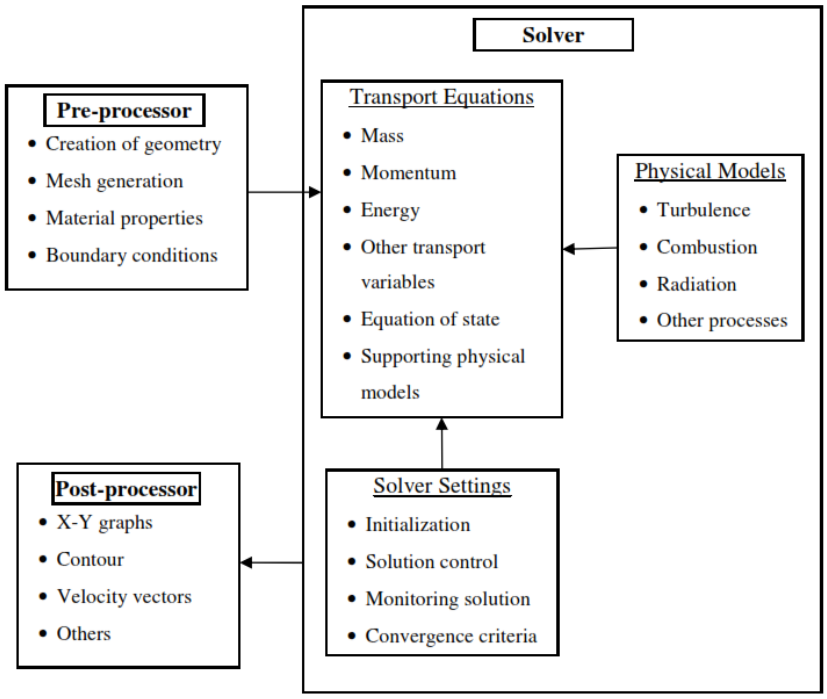

Figure 1: The inter-connectivity functions of these three main elements within a CFD analysis framework [10].

2.1. GOVERNING EQUATIONS

The governing equations are based on the conservation of mass, momentum and energy. The conservation equations are related to the rate of change in the amount of that property within an arbitrary control volume to the rate of transport across the control volume surface and the rate of the production within that volume [10].

The Navier-Stokes equations for compressible flows are examples of governing equations. The following examples include heat source ($\dot{\vec{q}}$) and body forces ($\vec{b}$).

Continuity equation: describes the conservation of mass:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \vec{V}\right) = 0 \qquad (2.1)$$

Linear Momentum Conservation:

$$\frac{\partial(\rho\vec{V})}{\partial t} + \nabla \cdot (\rho\vec{V}\vec{V} - \underline{\underline{T}}) = \rho\vec{b} \qquad (2.2)$$

Energy Conservation in total energy form:

$$\frac{\partial(e_o\rho)}{\partial t} + \nabla \cdot (e_o\rho\vec{V} - \underline{\underline{T}} \cdot \vec{V} + \dot{\vec{q}}_c + \dot{\vec{q}}_r) = \rho(\vec{V} \cdot \vec{b} + \dot{\vec{q}}) \qquad (2.3)$$

Where: ρ is the fluid density, is the flow velocity, T is the stress tensor, is the body forces, is the internal energy, is the conduction heat transfer flux and is the radiation heat transfer flux and is the heat source.

2.2. TURBULENCE MODELS

Most flows of engineering significance are turbulent in nature. Flow structure in the turbulent regime is characterized by random, three-dimensional motion of fluid particles in addition to the mean motion, which is macroscopic mixing of fluid particles from adjacent fluid layers.

The Reynolds averaged Navier-Stokes (RANS) equations is a mathematical model of turbulent flow that introduces additional terms in the governing equations that need to be modeled in order to include the turbulence effects. The RANS equations govern the transport of the averaged flow quantities, with the whole range of the scales of turbulence being modeled. The RANS-based modeling approach therefore greatly reduces the required computational effort and resources and is widely adopted for practical engineering applications.

However, it is an unfortunate fact that no single turbulence model is universally accepted as being superior for all classes of problems. The choice of turbulence model will depend on considerations such: the physics encompassed in the flow, the established practice for a specific class of problem, the level of accuracy required, the available computational resources, and the amount of time available for the simulation [11].

Among available CFD models, the RANS approach commonly based on turbulent kinetic energy (k) closure schemes is used for engineering applications. It is increasingly used in simulations of flow. The most widely used RANS models are two equation models, which solve two transport equations [10]. The k-ε model and its variants are the best known among these models, which require the solutions of (k) equation and dissipation rate equation (ε) The k-ω model and its variants, where (ω) is the specific dissipation rate are also very used. In this thesis, two models are going to be studied: the standard k-ε model and the standard k-ω model.

2.2.1. STANDARD K-Ε MODEL

The standard k- ε model is a semi-empirical model based on model transport equations for the turbulence kinetic energy (k) and its dissipation rate (ε). The model transport equation for k is derived from the exact equation, while the model transport equation for ε is obtained using physical reasoning and bears little resemblance to its mathematically exact counterpart [11].

and

In these equations, represents the generation of turbulence kinetic energy due to the mean velocity gradients. is the generation of turbulence kinetic energy due to buoyancy. represents the contribution of the fluctuating dilatation in compressible turbulence to the overall dissipation rate. , and are constants. and are user-defined source terms.

The turbulent (or eddy) viscosity, , is computed by combining κ and ε as follows:

—

The model constants , , , and have the following default values in ANSYS FLUENT:

=1.3

These default values have been determined from experiments with air and water for fundamental turbulent shear flows including homogeneous shear flows and decaying isotropic grid turbulence. They have been found to work fairly well for a wide range of wall-bounded and free shear flows [11].

2.2.2. STANDARD K-Ω MODEL

The standard k-ω model is an empirical model based on model transport equations for the turbulence kinetic energy (k) and the specific dissipation rate (ω), which can also be thought of as the ratio of ω to k [12].

— — — —

and

— — — —

In these equations, represents the generation of turbulence kinetic energy due to the mean velocity gradients. represents the generation of ω. and represent the effective diffusivity of κ and ω. and are user-defined source terms.

The turbulent viscosity, , is computed by combining κ and ω as follows:

2.3. CFD TECHNIQUES

2.3.1. OVERVIEW OF FLOW SOLVERS

ANSYS FLUENT has two solvers which use different numerical methods: pressure-based solver and density-based solver.

The pressure-based approach was developed for low-speed incompressible flows, while the density-based approach was mainly used for high-speed compressible flows. However, recently both methods have been extended and reformulated to solve and operate for a wide range of flow conditions beyond their traditional or original intent.

In both methods the velocity field is obtained from the momentum equations. In the density-based approach, the continuity equation is used to obtain the density field while the pressure field is determined from the equation of state. On the other hand, in the pressure-based approach, the pressure field is extracted by solving a pressure correction equation which is obtained by manipulating continuity and momentum equations.

In both methods, the governing integral equations for the conservation of mass, momentum, energy, and other scalars such as turbulence will be solved. In both cases a control-volume-based technique is used. It consists of three subjects: the division of the domain into discrete control volumes using a computational grid; the integration of the governing equations on the individual control volumes to construct algebraic equations for the discrete dependent variables such as velocities, pressure, temperature and conserved scalars; and the linearization of the discretized equations and solution of the resultant linear equation system to yield updated values of the

dependent variables. The two numerical methods employ a similar discretization process (finite-volume), but the approach used to linearize and solve the discretized equations is different [11].

The density-based solver, the one used in this study, solves the governing equations of continuity, momentum, energy, and also species transport equations simultaneously (i.e., coupled together). Governing equations for additional scalars are solved afterward and sequentially (i.e., segregated from one another and from the coupled set). Because the governing equations are non-linear (and coupled), several iterations of the solution loop must be performed before a converged solution is obtained. This solver can be implicit or explicit. The one applied in this work is the implicit solver, which allows longer time steps while preserving stability at higher Courant numbers.

2.3.2. DISCRETIZATION

There are some computational techniques that are required to solve the governing equations. The process of obtaining the computational solution consists of two stages. The first stage involves the conversion of the partial differential equations (PDE) and auxiliary (boundary and initial) conditions into a system of discrete algebraic equations. This stage is known as the discretization stage. The second stage involves numerically solving the system of algebraic equations, which can be achieved by either direct methods, such as Gaussian Elimination and Thomas Algorithm, or iterative methods, such as Jacobi and Gauss-Siedel [10].

An overview process of the computational solution procedure is presented in Figure 2.

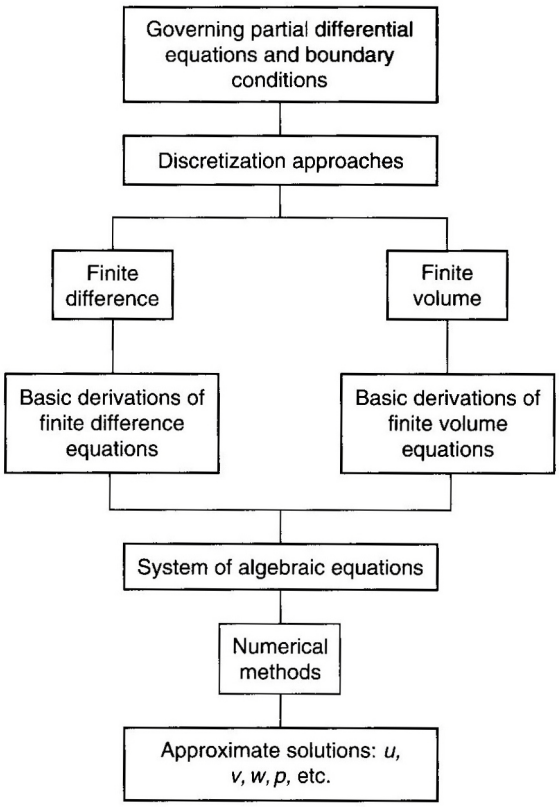

Figure 2: Overview process of the computational solution procedure [10].

Two of the most known discretizations tools are the finite difference and finite-volume method.

In the finite-difference method, at each point of the grid used to describe the fluid-flow domain, the Taylor series expansions are used to generate finite-difference approximations to the partial derivatives of the governing equations. These derivatives, replaced by finite-difference approximations, yield an algebraic equation for the flow solution at each grid point. This method generally requires a uniform distributed mesh. For a non-uniform grid distribution, some mathematical manipulation is required to transform the governing equations into a computational domain in generalized coordinates before applying the finite-difference

approximations. Figure 3 shows a representation of a one and two-dimensional uniformly distributed Cartesian grid for the finite difference method.

In the finite-volume method, the computational domain is subdivided into a finite number of contiguous control volumes. Therefore, because this method doesn`t work with grid intersection points, it has the capacity to accommodate any type of grid. Then, instead of structured grids, unstructured grids can be employed. This feature allows this method to be adopted by almost all commercial CFD codes, including ANSYS FLUENT [10]. Figure 4 shows a representation of structured and unstructured mesh for the finite-volume method.

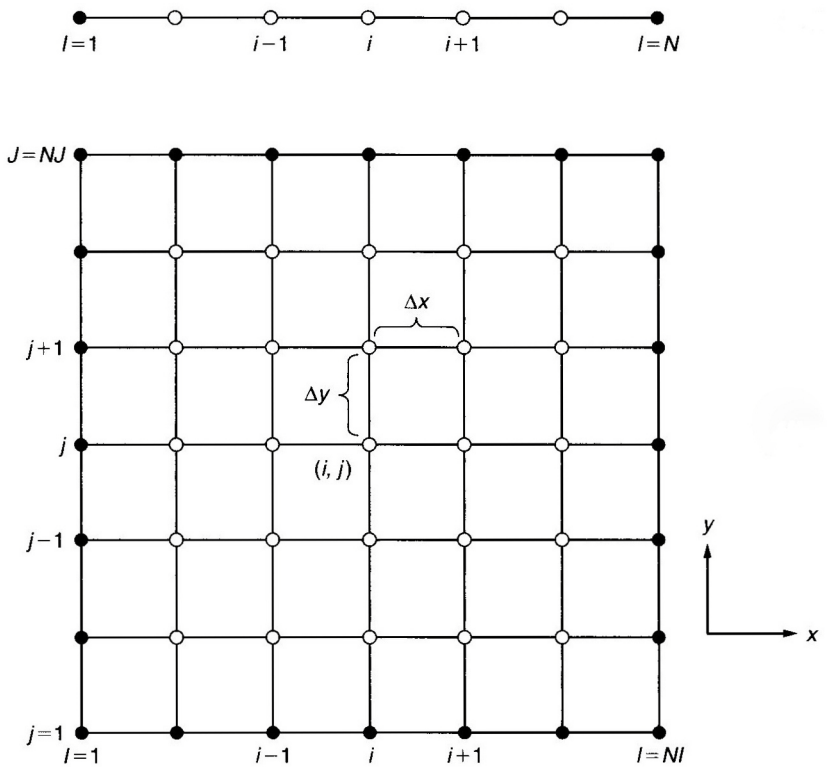

Figure 3: Representation of a one and two-dimensional uniformly distributed Cartesian grid for the finite difference method [10].

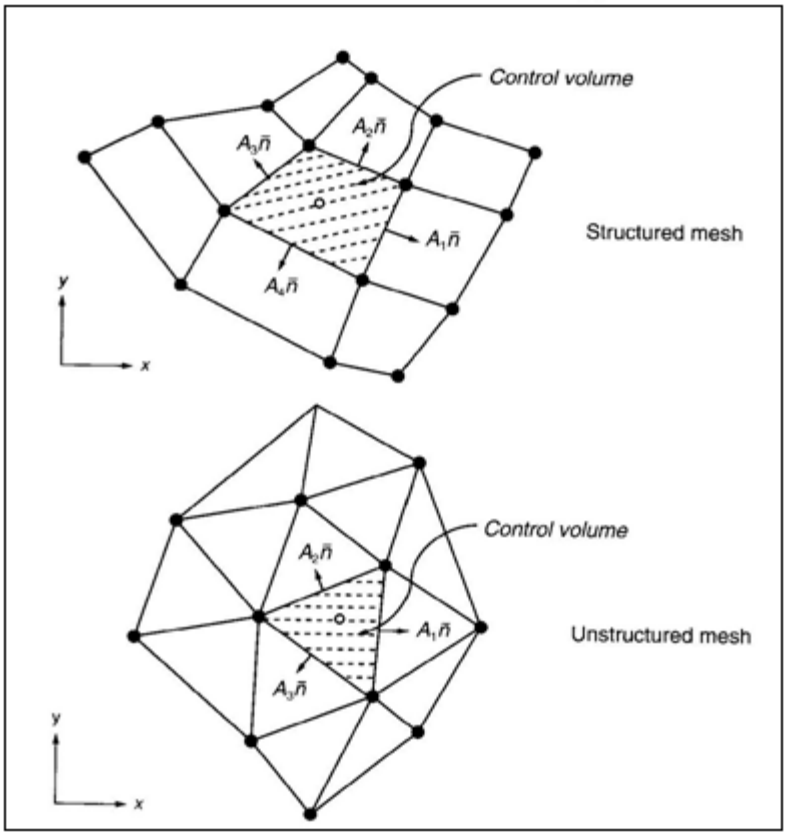

Figure 4: Representation of structured and unstructured mesh for the finite-volume method [10].

The equation governing the steady convection and diffusion process of a property in a given one-dimensional flow field u is:

By default, ANSYS FLUENT stores discrete values of the scalar at the cell centers. However, face values are required for the convection terms in the discretized transport equations and must be interpolated from the cell center values. This is accomplished using an upwind scheme. Upwinding means that the face value is derived from quantities in the cell upstream, or "upwind," relative to the direction of the normal velocity [11].

The first order upwind scheme is stable and satisfies transportiveness, boundedness and conservativeness. Although this scheme promotes numerical

stability, it is widely known to cause unwanted numerical diffusion in space. In order to reduce these numerical errors, high order approximations, such as the second-order upwind and third-order QUICK scheme are widely applied. In this work, as it consists of 2D analyses, third-order approximations were not necessary. Only first and second-order upwind schemes were used.

2.3.3. GRID GENERATION

CFD requires the subdivision of the domain into a number of smaller subdomains in order to solve the flow physics within the domain geometry that has been created. This result in the generation of a grid (or mesh) of cells (elements or control volumes) overlaying the whole domain geometry [10]. The essential fluid flows that are described in each of these cells are usually solved numerically so that the discrete values of the flow properties are determined. It is very important to create a well-constructed mesh because it will have a great influence on the solution. A mesh can be structured, unstructured or hybrid.

A single block structured mesh usually may comprise square elements (2D) or hexahedral elements (3D) which are orthogonal in i, j space (2D) or i, j, k space (3D) and follow a uniform pattern. However, it is also possible to have wedges (3D), triangles (2D) and pyramids (3D) in a structured mesh.

The connectivity on this type of mesh is straightforward because cells adjacent to a given elemental face are identified by the indices and the cell edges form continuous mesh lines that begin and end on opposite elemental faces as illustrated in Figure 5. The regularity of the connectivity allows conserving space since neighborhood relationships are defined by the storage arrangement.

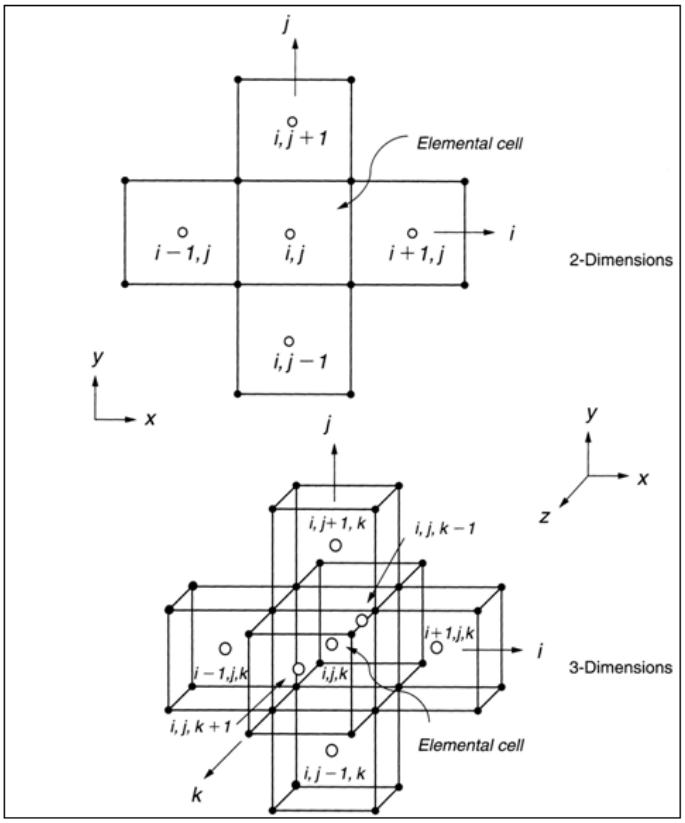

Figure 5: Nodal indexing of Elemental cells in two and three dimensions for a structured mesh [10].

An unstructured mesh does not follow a uniform pattern. It is usually comprised of triangle elements (2D) or tetrahedron (3D). The cells are allowed to be assembled freely within the computational domain. The connectivity information for each face thus requires appropriate storage in the form of a table. Compared to structured meshes, the storage requirements for an unstructured mesh can be substantially larger. Figure 6 shows the representation of four neighboring surface triangles for unstructured grids that illustrates the indexing scheme.

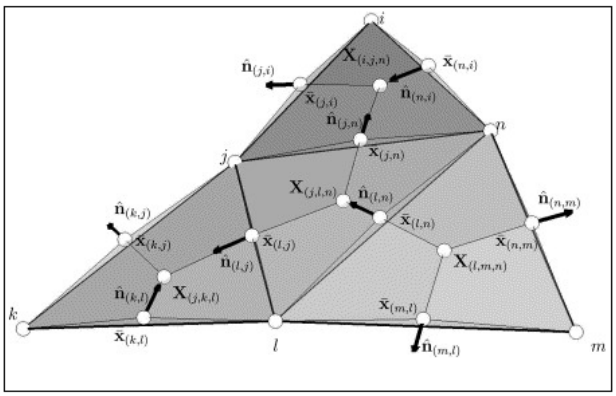

Figure 6: Representation of four neighboring surface triangles for unstructured grids that illustrates the indexing scheme [13].

A hybrid mesh is a mesh that contains structured and unstructured portions. There is also a possibility of a "mixed" mesh. The term "mixed" is usually applied to meshes that contain both elements associated with structured meshes and elements associated with unstructured meshes.

In this thesis, the designs were analyzed using structured grids.

CHAPTER III

OPTIMIZATION

Optimization is a tool to obtain the optimum value of a certain function. For a given application, several engineering projects are possible. However, due to economical costs, it is important to find the best configuration, which represents the optimum.

3.1. BASIC CONCEPTS

3.1.1. OBJECTIVE FUNCTIONS

Defined as a mathematical expression of some value to be optimized. It can be either maximized or minimized. It can be represented as:

Where are variables that must be modified in order to reach the optimal values of U.

3.1.2. CONSTRAINTS

In real engineering problems there will always be constraints. For example, constraints can be due to environmental concerns or materials limitations. There are equality and inequality constraints.

→ equality (3.2)

→ inequality (3.3)

3.2. DETERMINISTIC METHODS

These are methods with strong mathematical background, thus, it is possible to prove that the minimum of a function under certain conditions was found. Because they require the computation of the gradient of the vector, which is the vector of the first derivatives of the object function, this is also called gradient-based methods.

Steepest descent method, the conjugate gradient method, the Newton–Raphson, and the quasi-Newton method are examples of deterministic methods.

3.3. EVOLUTIONARY AND STOCHASTIC METHODS

Evolutionary methods, in contrast to the deterministic methods, do not rely, in general, on strong mathematical basis and do not make use of the gradient nor second derivative of the objective function as a direction of descent. The evolutionary optimization algorithms attempt to mimic nature in order to find the minimum of the objective function [14]. However, there is no proof of convergence to a global minimum, although they usually converge. These methods require more function evaluations than the gradient-based ones. Genetic algorithm, differential evolution, particle swarm and simulated annealing are examples of evolutionary methods.

3.3.1. GENETIC ALGORITHM

This algorithm was used in this work. Genetic algorithms are heuristic global optimization methods that are based on the process of natural selection. They use only the values of the objective function, that is, they do not use gradients of the objective function. Instead of starting from an initial guess, the optimizer starts from a randomly generated population of candidate designs and seeks to produce improved designs from one generation to the next. This is accomplished by exchanging genetic information between designs in the current population, in what is referred to as the crossover operation. Hopefully, this crossover produces improved designs, which are then used to populate the next generation [15], [16]. The size of the initial population is typically 5N to 10N, where N is the total number of design variables.

The basic genetic algorithm works with a collection or population of candidate solutions to the optimization problem. The algorithm works in an iterative manner. At each iteration, also called generation, three operators are applied to the entire

population of designs. These operators are: selection, crossover and mutation [15]. In this method, the design variable is represented as a string of chromosomes, expressed in terms of binary variables.

3.3.2. PARTICLE SWARM ALGORITHM

This algorithm was also used in this work. It is a heuristic search method whose mechanics are inspired by the swarming or collaborative behavior of biological population [14].

Particle Swarm Optimization (PSO) attempts to simulate the behavior of swarms of birds. This behavior is a combination of sociability and individuality of every member of the population. In PSO, a set of randomly generated solutions (initial swarm) propagates in the design space towards the optimal solution over a number of iterations based on large amount of information about the design space that is assimilated and shared by all members of the swarm.

Particle Swarm is similar to the Genetic Algorithm (GA) in the sense that these two evolutionary heuristics are population-based search methods. In other words, PSO and the GA move from a set of points (population) to another set of points in a single iteration with likely improvement using a combination of deterministic and probabilistic rules.

3.4. HYBRID OPTIMIZATION

A hybrid optimization is a combination of the deterministic and the evolutionary/stochastic methods, in the sense that it utilizes the advantages of each of these methods. The hybrid optimization method usually employs an evolutionary/stochastic method to locate a region where the global extreme point is located and then automatically switches to a deterministic method to get to the exact

point faster. The hybrid optimization method is quite simple conceptually, although its computational implementation is more involved [17].

3.5. RESPONSE SURFACE METHODS

To significantly accelerate the entire design optimization process, response surfaces (metamodels) were be used in this study. They are often used to replace very complicated physical models, to generate correlations of experimental data and to reduce the computational cost involved [1]. Response surfaces are mathematical functions used to simulate the behavior of processes, experiments, and complex engineering analysis techniques. They allow optimization techniques to be feasibly applied to classes of problems outside of computer evaluated objective functions. This occurs because a properly constructed response surface that captures the behavior of a complex, computationally intense objective can be used to speed up the optimization process. Also, a properly constructed response surface can be used to optimize a process, or experimental work, where only discrete, empirical samples of the underlying system's response to process parameters can be evaluated [18]. There are a lot of different methods for generating response surface models. One of the most popular uses radial basis functions (RBFs). It was found that RBFs were able to construct an interpolation scheme with favorable properties such as high efficiency, good accuracy, and capability of dealing with scattered data, especially for higher dimension problems [14].

Another interpolation method is Kriging. This model is a response surface model that represents a relationship between objective function (output) and design variables (input) using a stochastic process. The Kriging model drastically reduces the computational time required for objective function evaluation in the optimization (optimum searching) process [19]. This method is based on the assumption that the

parameter being interpolated can be treated as a regionalized variable. A regionalized variable is intermediate between a truly random variable and a completely deterministic variable in that it varies in a continuous manner from one location to the next and therefore points that are near each other have a certain degree of spatial correlation, but points that are widely separated are statistically independent [20]. In optimization, this method is very accurate when the number of design variables is small. It was used in this research.

CHAPTER IV

METHODS OF SOLUTION

4.1. SHAPE GENERATION

In order to create the nozzle shapes, the Fortran code *Shock-GSD-newH.for* was used. This code can be found in Appendix B. Originally this code was created to find the location of a normal shock inside a quasi 1D converging-diverging nozzle. Therefore, by running the code, the shape and the mesh are generated and also other parameters are calculated, such as ratio of absolute temperatures, static pressure and gas density across the shock, Mach number computed just upstream and downstream of the shock and change in entropy. Several output files are created with all these data. However, for this study, only the information related to the nozzles' shapes were considered.

This code gives two options to the user specify the nozzle shape. The first one uses a fifth order polynomial in the x axis to find values of nozzle radius at any x location. In this case, parameters such as the inlet, throat and exit radii must be input by the user. The second one utilizes Mach number-area analytical relation. The parameters such as inlet, throat and exit Mach numbers must be specified in this option are.

The first option was the one adopted in this thesis. The relevant input parameters to generate the shapes are the design variables considered in this work, except the throat radius, which is constant. The ranges of their values are presented in Table 1, below. The throat is placed at x = 0.0. Figure 7 shows the representation of input parameters.

Table 1: Range of input parameters (design variables).

Inputs	Range of values
x- location of the inlet (Xi)	-4 to -2.5 m
x- location of the exit (Xe)	4 to 5.5 m
nozzle inlet radius (Ri)	2 to 5 m
nozzle exit radius (Re)	1.5 to 7 m
nozzle inlet wall slope (alpha_i)	-14 to -3 °
nozzle inlet wall slope (alpha_e)	3 to 15°
nozzle throat radius (Rt)	0.5 m

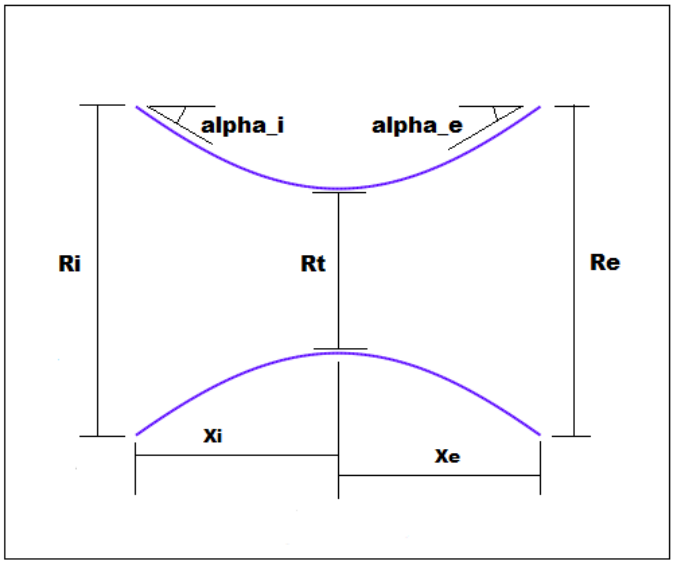

Figure 7: Representation of input parameters.

4.2. MESH GENERATION

After creating the geometries of the nozzles, they were imported by the software ANSYS GAMBIT, which was used to generate the grids.

In order to mesh geometry in ANSYS GAMBIT, first it is necessary to mesh the edges and then mesh the face.

For the four edges considered (inlet, exit, axis and wall), the grid grading scheme chosen were different. For the axis and the wall, the grading scheme was constant with 200 intervals uniformly distributed from inlet to exit of the nozzle. However, because of the need of boundary resolutions near the wall, the grids have to be more clustered in this area. Therefore a non-symmetric grading scheme was applied to the inlet and exit edges.

For each of the non-symmetric grading schemes, ANSYS GAMBIT positions mesh nodes along the edge, such that the ratio of any two succeeding interval lengths is constant [21]. That is,

—

where and are the lengths of intervals i and i+1, respectively and R is a fixed value. For any given number of intervals (*n*), the grading schemes differ from each other only with respect to the manner in which ANSYS GAMBIT determines the value of the interval length ratio (R). The Figure 8 describes the grading scheme.

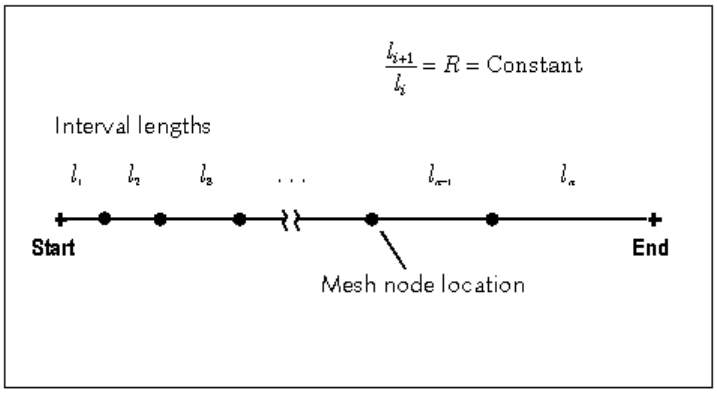

Figure 8: Edge mesh grading parameters [21].

The two grading schemes used were Successive Ratio and First Length. The Successive Ratio was applied to create the 200 uniforms intervals mentioned above. Therefore the interval length ratio (R) inputted was 1. The First Length scheme was

applied in the other 2 edges. The formula that GAMBIT uses to determine the interval length ratio (R), for this scheme is:

$$R = \frac{\text{last length}}{\text{first length}}$$

In this scheme, the parameter to be input by the user is the first length, instead of the ratio. The values used varied depending on the nozzle inlet and exit radii, but they were between 0.1 and 0.2. The number of intervals chosen was 40, 50 or 60. Therefore, the nozzzles created had 8,000, 10,000 or 12,000 cells initially. Some adaptations were necessary later in some cases. Figure 9 shows mesh nodes along the nozzle edges in ANSYS GAMBIT.

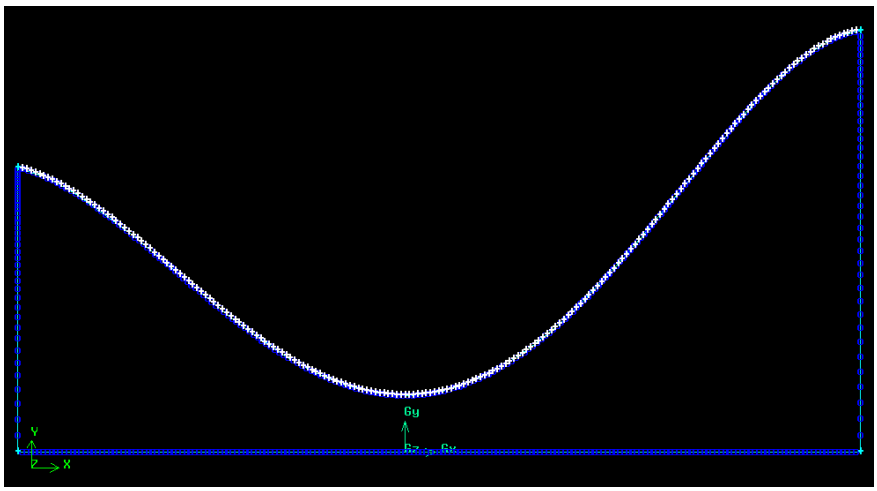

Figure 9: Mesh nodes along the edges of a CD nozzle in ANSYS GAMBIT.

After creating the mesh nodes, it is necessary to mesh the face. In order to do that, two parameters must be specified: the “Elements” defines the shape of the elements that are used to mesh the face. The “Type” defines the pattern of mesh elements on the face. The element parameter chosen was “Quad”, which specifies that the mesh includes only quadrilateral mesh elements. The type parameter chosen was

"Map", which creates a regular, structured grid of mesh elements. The Figure 10 presents an example of a nozzle mesh generated in ANSYS GAMBIT.

After those procedures, the boundary types were specified. Then, the nozzles meshes were ready to be exported to ANSYS FLUENT as a 2D mesh.

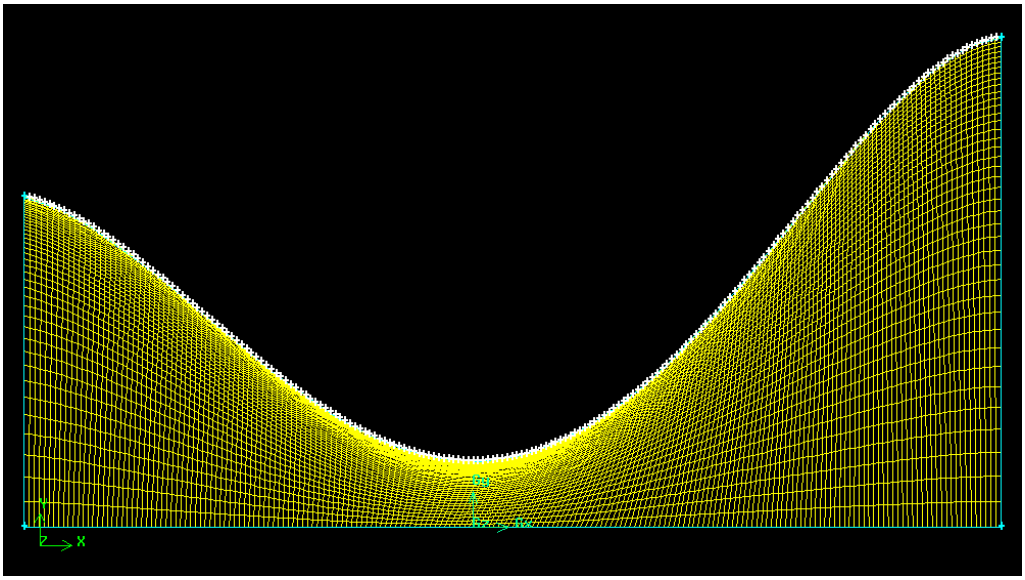

Figure 10: Example of a nozzle mesh generated in ANSYS GAMBIT.

4.3. ANSYS FLUENT

As previously mentioned, the CFD software package used to analyze the fluid flow was ANSYS FLUENT. Two turbulence models were compared.

4.3.1. Defining Models and Material

The converging-diverging nozzles were defined as 2D axisymmetric and solved using double precision. This solver gives more accurate results in cases that involve high speed and high thermal-conductivity ratios than the single-precision.

Also, the density-based solver with implicit formulation was applied, considering the high Mach number flow. It is said that both pressured-based and density-based solvers were recently reformulated in order to be able to cover wider range of flow conditions. However, in the case studies performed in this work, it was

not possible to get convergence using pressure-based solver. This probably happened because the nozzle exit Mach number of the flow can reach a value of 8.

Material properties were specified in ANSYS FLUENT. The geometric domain (nozzle) was defined as a single-phase system which consisted of air modeled as ideal-gas.

Two turbulence models were compared: standard k-ε and standard k-ω. For both models, the default constants presented in section 2.2 were used.

4.3.2. Defining Operating Pressure

It is important to set the operating pressure correctly in compressible flow calculations since the software uses it to compute the absolute pressure used in the ideal gas law.

However, the operating pressure is only important for low Mach number flows (Ma << 1), when it is influenced by the numerical roundoff. For high Mach number flows, as the case in this work, this pressure should be set to zero.

4.3.3. Defining Boundary Conditions

INLET: In order to have hypersonic flow at the exit of the nozzles, the inlet pressure applied must be very high. The value defined as inlet total pressure was 20 MPa. An initial guess had also to be input. For a subsonic inlet, as in this work, the initial guess should be the value of the static pressure. This value can be calculated from the 1D analysis. This value is updated by the code, but if it is not specified correctly, the solution will not converge. The initial guess chosen was 19.90 MPa.

So as to simulate the real condition of a flow entering a CD nozzle, instead of using a constant inlet temperature, a non-uniform profile was applied. The

temperature considered near the axis was 2200 K and it was decreasing radially at the inlet until 1000 K near the wall. One example of this profile is presented in Figure 11.

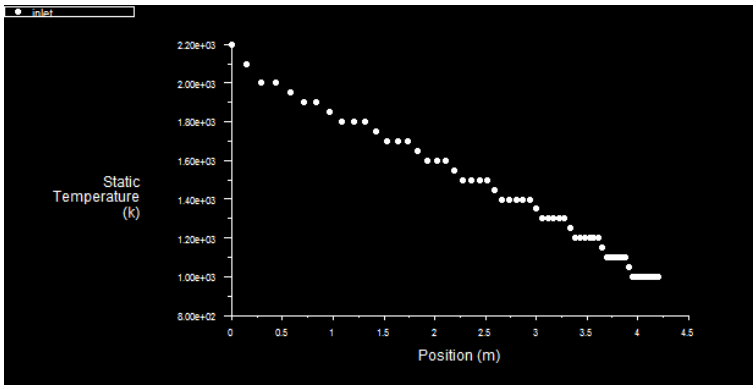

Figure 11: Example of non-uniform radial temperature profile applied at the nozzle inlet.

EXIT: The exit pressure was defined as 100 kPa. However, this value is used for subsonic flow only. Should the flow become locally supersonic, the pressure is extrapolated at the exit boundary.

WALL: The wall was considered adiabatic. Therefore, the heat flux is zero.

4.3.4. SOLVING

After applying all settings, the simulation should be initialized. However, the discretization methods and the under-relaxations factors must be defined first. The aim is to get convergence by using the second-order upwind method for flow, turbulent kinetic energy and specific dissipation rate, in order to reduce the numerical diffusion. Higher order approximations, such as the third-order QUICK scheme is not necessary. Since the cases are 2D, the second-order provides accurate results.

However, in most of the cases, it was not possible to achieve the convergence directly, using the initial guess specified in boundary conditions. Therefore, it was necessary to start with the first-order upwind scheme. Since the convergence was reached, this solution was used as initial guess for the second-order upwind.

The default under-relaxations factors were used. In ANSYS FLUENT, the default under-relaxation parameters for all variables are set to values that are near optimal for the largest possible number of cases. However, sometimes when the solution was not converging, it was necessary to decrease these factors to obtain the convergence.

Each of the 40 nozzle shape designs was solved until the residuals decreased by five orders of magnitude. The results of the 80 analyses were recorded; 40 using k-ε and 40 using k-ω turbulence models. Since the main purpose of this thesis is to optimize nozzle shapes in order to have the most uniform flow at exit, the density, temperature and Mach number distribution at the nozzle exit were computed and their standard deviations calculated by using the following formula:

$$SD = \sqrt{\frac{}{}}$$

Here, is the mean and n is the size of the sample, which in this case is the number of points computed at the nozzle exit.

4.4. MODEFRONTIER OPTIMIZATION SOFTWARE

The multi-objective optimization software package utilized in this work was ModeFrontier [22]. All input variables and the standard deviations calculated from the density, temperature and Mach number distribution at the nozzle exit computed for the 40 shapes were placed in an Excel sheet and imported by ModeFrontier through a tool called "Data Wizard." After defining in the software which parameters will be the input variables, outputs, objectives and constraints, a workflow was built and a Design Table was filled with the imported data. Table 2 presents the definition of those parameters in the software and Figure 12 shows the workflow created with these data.

Table 2: Definition of the imported data in ModeFrontier.

Parameters	Description	Symbols	Range of values
Input variables	x- location of the inlet	Xi	-4 to -2.5 m
	x- location of the exit	Xe	4 to 5.5 m
	nozzle inlet radius	Ri	2 to 5 m
	nozzle exit radius	Re	1.5 to 7 m
	nozzle inlet wall slope	Alpha_i	-14 to -3 °
	nozzle exit wall slope	Alpha_e	3 to 15°
Outputs	standard deviation of the density distribution at the nozzle exit	dens	> 0
	standard deviation of the temperature distribution at the nozzle exit	temp	> 0
	standard deviation of the Mach number distribution at the nozzle exit	Mach	> 0
Objectives	Minimize the standard deviation of the density distribution	Min_dens	-
	Minimize the standard deviation of the temperature distribution	Min_temp	-
	Minimize the standard deviation of the Mach number distribution	Min_mach	-
Constraints	The standard deviation of the density distribution has to be greater than zero.	Const_dens	Dens > 0
	The standard deviation of the temperature distribution has to be greater than zero.	Const_temp	Temp > 0
	The standard deviation of the Mach number distribution has to be greater than zero.	Conts_Mach	Mach > 0

It is obvious that the standard deviation has to be greater than zero. However, when the solutions were calculated without these constraints, some negative standard deviation appeared among them. This probably happened because of some fail on interpolation using response surfaces. In order to fix this problem, the constraints were added.

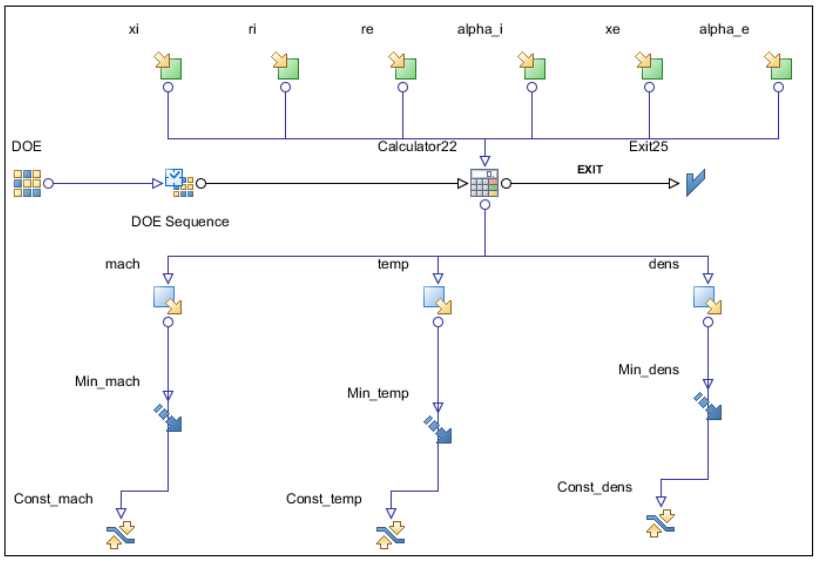

Figure 12: Workflow created in ModeFrontier.

Since the Design Table is prepared, the process to generate response surfaces can start. At this point, the method to create them was specified. The Kriging method was chosen. In general, regardless of the meta-model type, design type, or the complexity of the response, the performance tends to improve when the number of real designs evaluated increases.

After generating the surrogate models for density, Mach number and temperature standard deviations, the optimization starts. The method chosen was the Multi Objective Genetic Algorithm II (MOGA-II) designed for fast Pareto convergence. It supports geographical selection and directional cross-over and implements elitism for multi-objective search. Five hundred generations were created. Considering that the number of real analyses (shapes) imported was 40 (the design population size was 40 and kept constant), then 20,000 evaluations were made.

The general aim of a single-objective optimization is to find one global optimum design. In a multi-objective optimization such as this one, the aim is to find a set of non-dominating solutions, which are defined as those designs whose one

objective cannot be improved without compromising the values of the remaining objectives. This set is called the Pareto frontier [16].

The Pareto frontier solutions were calculated. In order to validate the results of the optimization, three randomly chosen Pareto frontier solutions were evaluated as real designs. The three shapes were generated, meshed and analyzed in ANSYS FLUENT. The results for density, temperature and Mach number standard deviation were compared to the ones obtained using meta-models.

Another optimization algorithm, Particle Swarm, was also used. The solutions calculated from both optimization algorithms were compared.

CHAPTER V

RESULTS

5.1. RESULTS FROM CFD

Initially, eighty simulations were run in ANSYS FLUENT (40 using k-ε turbulence model and 40 using the k-ω turbulence model) and their results for density, Mach number and temperature standard deviations were computed.

5.1.1. K-Ɛ MODEL

Among the 40 initial real designs, the test cases which had the minimum standard deviation for density, Mach number and temperature, respectively, for the k-ε model analysis are shown in Table 3. Case 25 presented the most uniform flow in terms of density. Case 28 was the most uniform for Mach number and case 7, for temperature.

Table 3: Designs which results in minimum standard deviation for density (case 25), Mach number (case 28) and temperature (case 7) using the k-ε model.

Case	X inlet	X exit	Inlet Radius	Exit Radius	Inlet angle	Exit angle	Standard Deviations		
							Density	Mach number	Tempe-rature
25	-3.2	5.23	3.24	4.5	-9	8	0.1557	0.9950	96.5519
28	-3	4.3	2	2.4	-6	9	0.4547	0.5436	88.3194
7	-3.9	5	3.2	6	-10	9	0.4125	2.6141	52.3236

Figure 13, Figure 14 and Figure 15 present the contour plots of density, Mach number and temperature distribution along the nozzles for cases 25, 28 and 7, respectively.

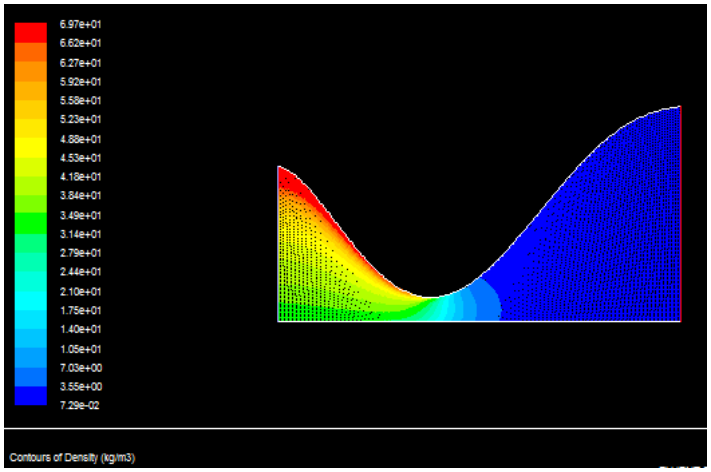

Figure 13: Contour plot of density distribution of case 25.

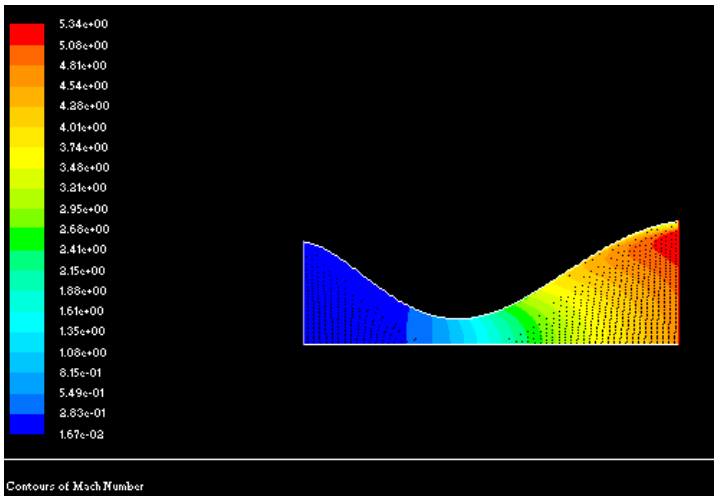

Figure 14: Contour plot of Mach number distribution of case 28.

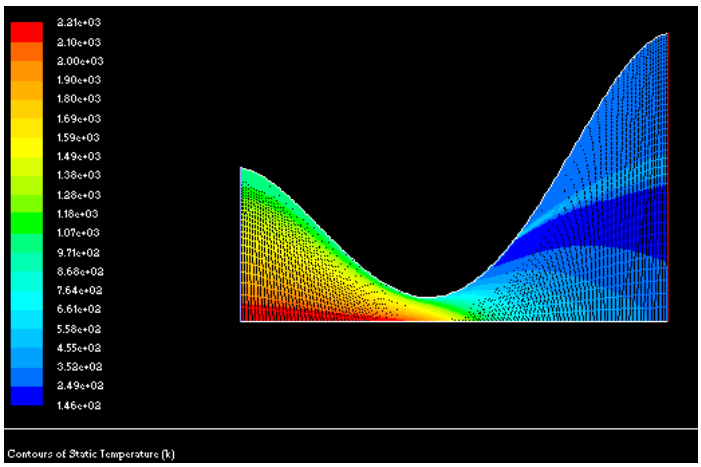

Figure 15: Contour plot of temperature distribution of case 7.

In Figure 15, the inlet temperature profile applied can be observed, and Figure 13 illustrates its influence on density. It can be observed that the fluid keeps roughly the same profile, until the throat, where the mixture occurs and it becomes more uniform downstream.

In summary, Figure 13, Figure 14 and Figure 15 show the flow behavior aimed in this work. A flow with a non-uniform inlet temperature profile and low Mach number becomes sonic at the throat, where the fluid mixes and leaves the nozzle with a very high Mach number (around 5 or higher) and more uniform. Since these are hypersonic nozzles, no shock is expected inside the nozzle. Then, the flow will keep accelerating until the exit. However, in some cases, flow separation was observed. When it occurred in a very thin layer, it did not present a great impact at the velocity at the exit. However, when recirculation occurred, especially in designs with a large exit radius, it drastically decreased the Mach number near the wall. An example of recirculation and a thin layer of flow separation are respectively presented in Figure 16.

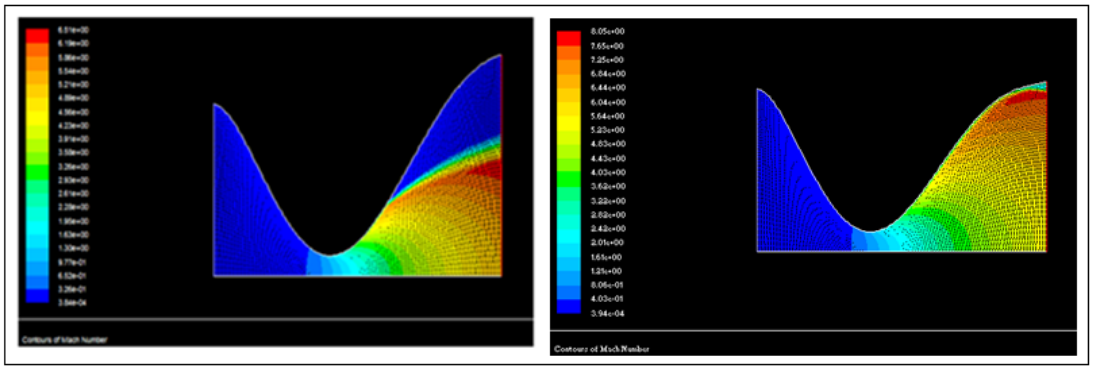

Figure 16: Examples of a large recirculation and a thin layer of flow separation in nozzles.

5.1.2. K-Ω MODEL

The same analysis was made for the k-ω model and it is presented Table 4.

Table 4: Designs which results in minimum standard deviation for density (case 25), Mach number (case 28) and temperature (case 15) using the k-ω model.

Shape	X inlet	X exit	Inlet Radius	Exit Radius	Inlet angle	Exit angle	Standard Deviations		
							Density	Mach number	Tempe-rature
25	-3.2	5.23	3.24	4.5	-9	8	0.1528	1.0611	102.5766
28	-3	4.3	2	2.4	-6	9	0.4544	0.5450	89.3406
15	-3.4	4.9	4	7	-7	6	0.4124	2.5526	52.9310

By analyzing Table 4, it can be observed that the case which presented the minimum standard deviation for density (shape 25) and the case which presented the minimum standard deviation for Mach number (shape 28) were the same as using the k-ε turbulence model. However, for the temperature, shape 15 obtained the minimum deviation.

Comparing the cases presented for the two turbulence models, it was observed that the difference between their results was very small, although k-ω is a more robust model. k-ω performed slightly better for density, while the k-ε performed better for temperature and Mach number.

5.2. RESULTS FROM OPTIMIZATION

5.2.1. RESPONSE SURFACES ANALYSES

As mentioned before, the response surface method was used for the optimization. Only the results from the k-ε turbulence model were considered. Meta-models for density standard deviation, Mach number and temperature were created using the Kriging and RBF methods. Since the response surfaces were calculated, it was necessary to check if the output computed in the Designs Table in ModeFrontier coincided with the corresponding values computed by them. Ideally, if a response surface is able to identify the behavior of the system, the designs computed by the

response surface will coincide with the designs evaluated in the original workflow (real simulations) and contained in the Designs Table. A tool called “RSM distance” plots a diagonal line of approximately 45° which shows the distance between selected response surface and some evaluated designs.

Figure 17 and Figure 18 illustrates that the meta-models created from the 40 real design evaluations by using both Kriging and RBF methods, could identify their behaviors since the results coincided.

Figure 19 shows two 3D response surfaces created by using both methods, which shows the minimum value calculated for density. In this case, it presents how the density is varying with the variables inlet radius (Ri) and exit radius (Re). When the other design variables are changed, this response surface behavior automatically changes.

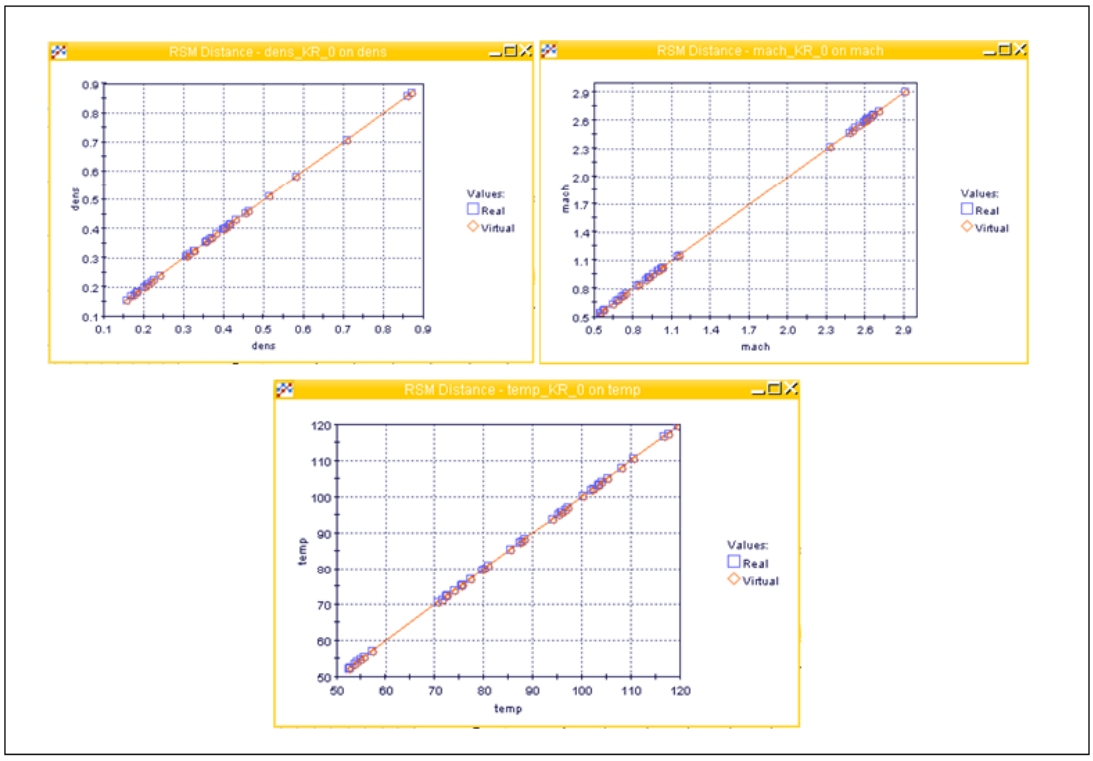

Figure 17: Comparison between k-ε model real designs evaluations and designs computed by response surfaces by using the Kriging method.

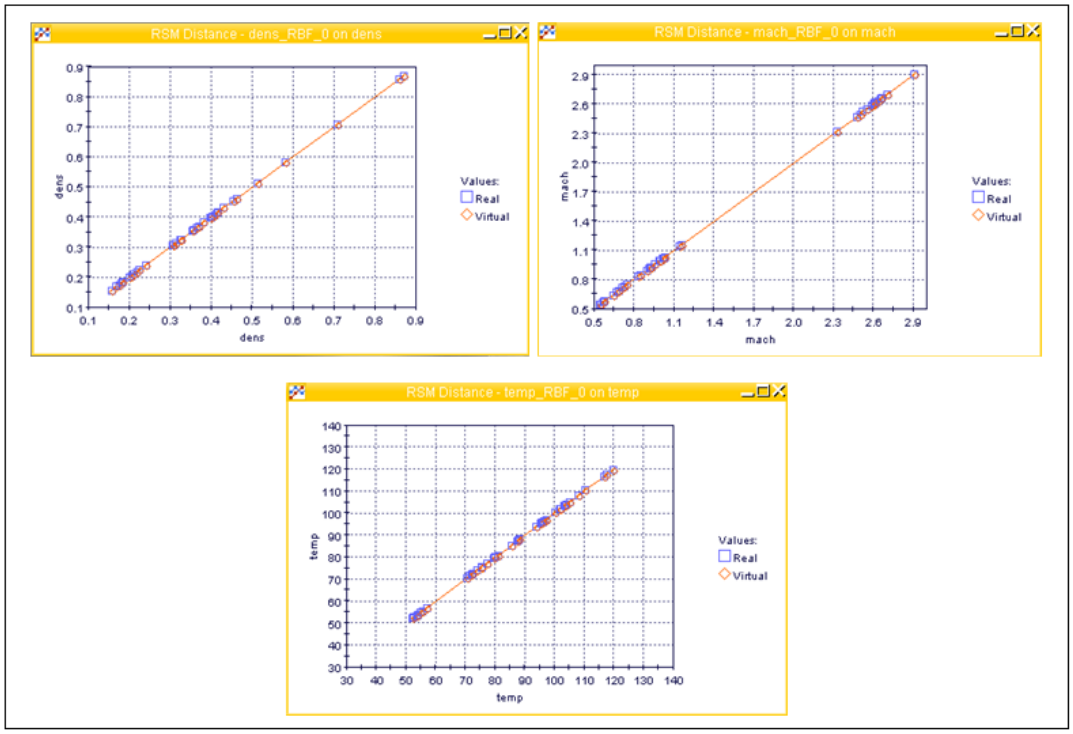

Figure 18: Comparison between k-ε model real designs evaluations and designs computed by response surfaces by using the RBF method.

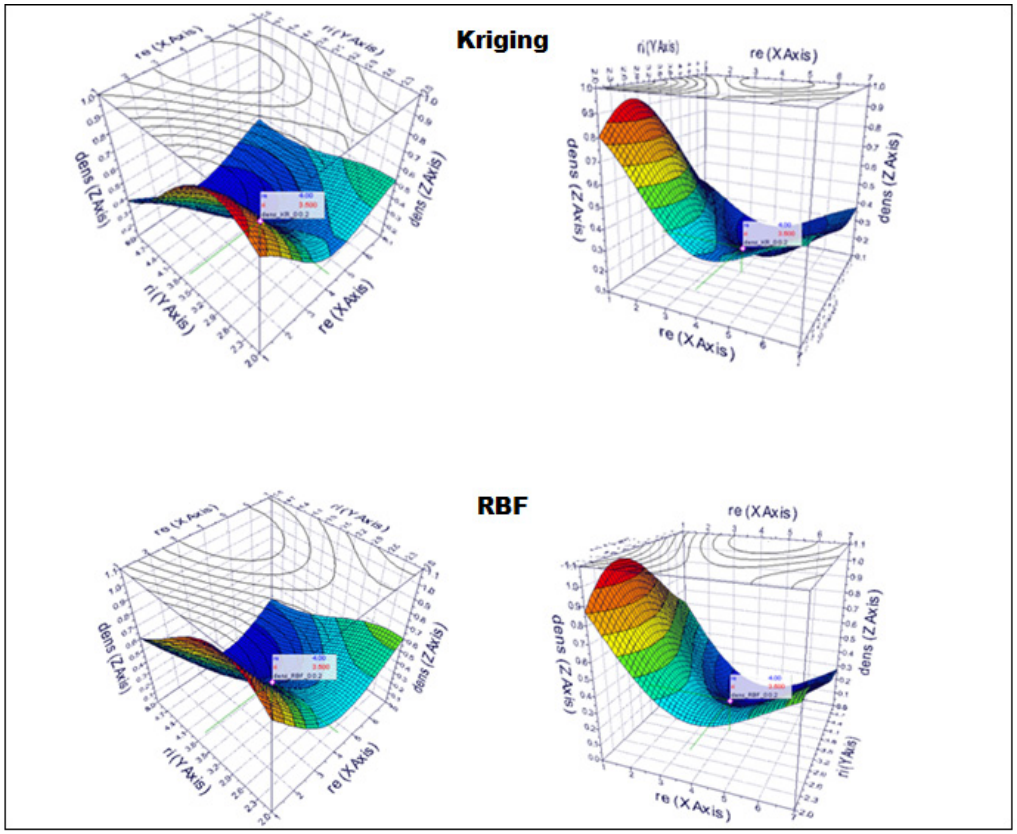

Figure 19: Response surfaces of density, created by using Kriging and RBF method, seen from two different views.

5.2.2. Optimization Results

After creating the response surfaces and running the optimization for 500 generations by using the Multi Objective Genetic Algorithm II (MOGA-II), 20,000 virtual designs were evaluated and the Pareto Frontier solutions were calculated. The Figure 20 shows a 3D scatter plot of all the solutions calculated for both Kriging and RBF interpolation methods.

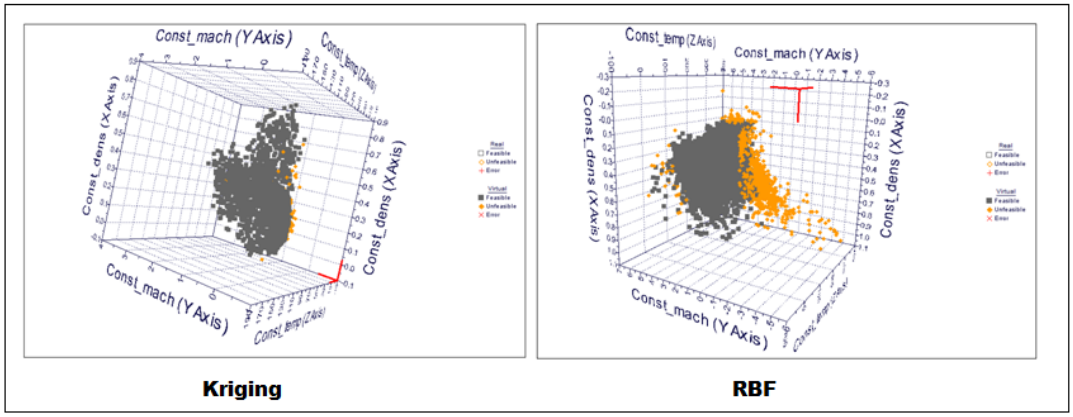

Figure 20: Scatter plot of the solutions for Kriging and RBF.

By comparing those solutions, it is possible to see that the RBF method produced more unfeasible designs (errors), in other words, designs which parameters are out of those specified by the constraints. However, this is not enough to prove that the Kriging method performed better. In order to affirm that, it was necessary to validate the results of both methods in ANSYS FLUENT. The validations are presented in the next section.

It can be observed that after each generation, the solution tends to move to the right side of the plot, which means that the values of the objectives are decreasing. This represents an improvement of the solution, since this is a minimization problem. The best designs are the rightmost points, which are also called “the Pareto Frontier solution” and are represented in two different views in Figure 21.

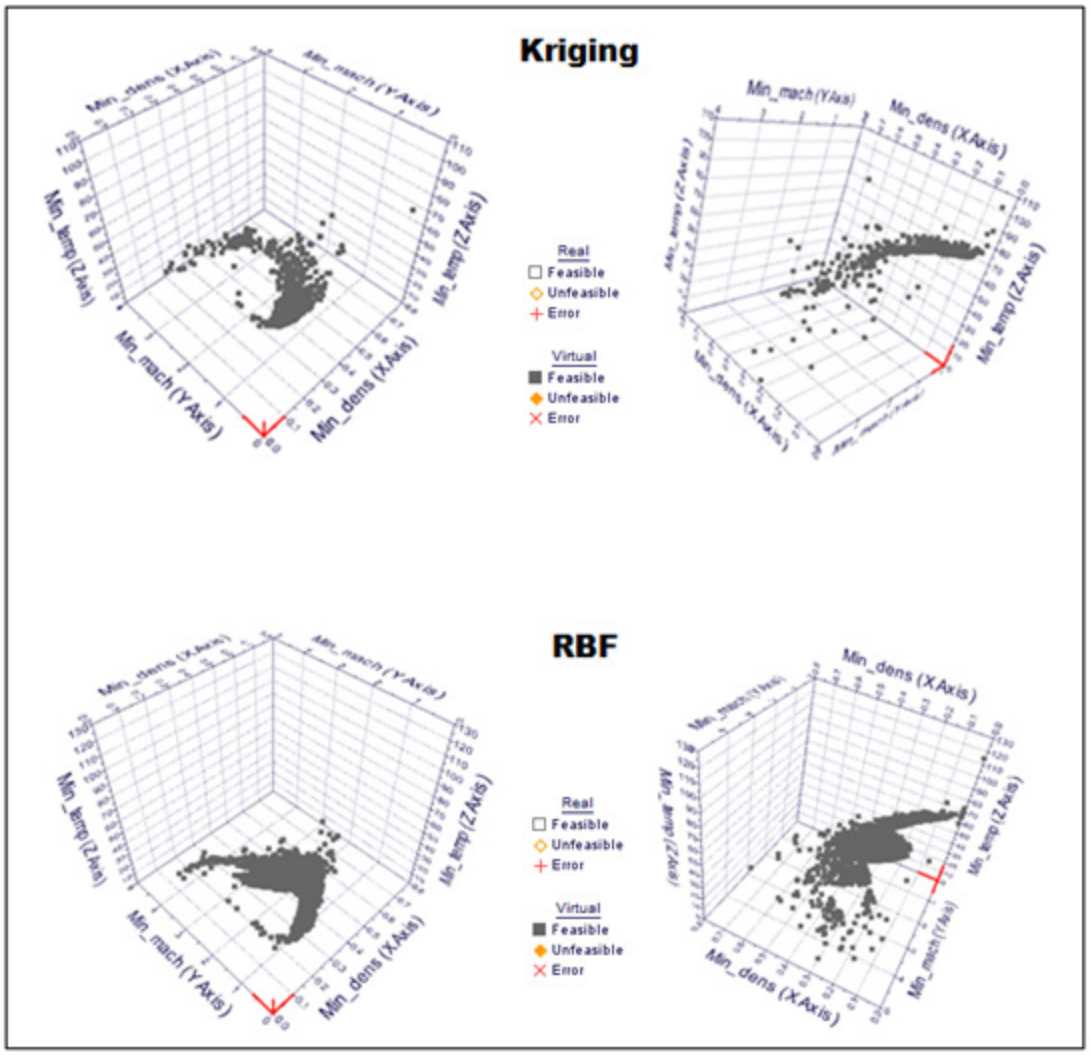

Figure 21: 3D scatter plot of the Pareto Frontier results seen by two different views.

5.3. VALIDATION

After obtaining the optimum virtual designs by using meta-models, it is necessary to check if the density, Mach number and temperature standard deviations computed correspond to the values obtained by real simulations. In order to validate these solutions, one randomly chosen Pareto design was evaluated in ANSYS FLUENT. The error between the virtual Pareto design and the real simulation is a good indicator of how accurate the response surfaces are. Table 5 presents the chosen Pareto designs and its parameters for Kriging and RBF methods and Table 6 presents the errors.

Table 5: Pareto design and their design variables for Kriging and RBF methods.

ID in MF	X inlet	X exit	Inlet Radius	Exit Radius	Inlet angle	Exit angle
962 (Kriging)	-3.4762	4.8447	3.4597	5.9169	-9.1095	8.1797
2453 (Kriging)	-2.854	4.1857	4.2	5.3958	-7.8333	1.4999
7123 (RBF)	-3.3785	4.2830	2.8065	2.9127	-13.401	14.972
11823 (RBF)	-2.9322	4.0497	4.7013	3.5526	-7.4868	1.4534

Table 6: Errors related to density, Mach number and temperature standard deviations, between the virtual Pareto design and the real simulation.

ID in MF	ANSYS FLUENT			ModeFrontier			Error		
	Standard Deviations			Standard Deviations			Standard Deviations		
	Dens	Mach	Temp	Dens	Mach	Temp	Dens	Mach	Temp
962 (Kriging)	0.1077	1.9058	132.025	0.037	2.57	42.47	65.66%	34.85%	67.83%
2453 (Kriging)	0.4374	2.7618	63.5541	0.254	1.855	65	41.9%	32.8%	2.3%
7123 (RBF)	0.2285	0.5477	97.1084	0.2833	0.2352	35.69	23.95%	57.05%	63.25%
11823 (RBF)	0.3813	2.534	51.7169	0.49	3.838	58.51	28.5%	51.46%	13.14%

By analyzing the values in table 6, it is possible to say that the response surfaces did not perform well for both methods, since the errors are large. Although the designs 2453 and 11823 presented a small error for temperature standard deviation, the other two designs presented a large error. This means that those response surfaces are not accurate and must be improved.

Besides, analyzing the errors of standard deviations, it is possible to observe that the behavior of the response surfaces is unstable. By comparing the Kriging method designs, for the 2453, the smallest error was for temperature standard deviation. However, for the 962, the temperature standard deviation presented the largest error. The same happened with the RBF designs 7123 and 11823.

Figure 22 shows the contours plots of density, Mach number, static temperature and static pressure for the design 2453 obtained by Kriging method.

By analyzing the Mach number contours plot, it is possible to see that there is recirculation at the divergent part. Therefore, although the Mach number can reach 6.5 at the exit, where there is recirculation it is less than 1. Then, this should be avoided, since it decreases the velocity of the flow at the exit. However, among the real designs evaluated, 40 percent presented recirculation. It happened when the nozzle exit radius was very large compared to the throat radius.

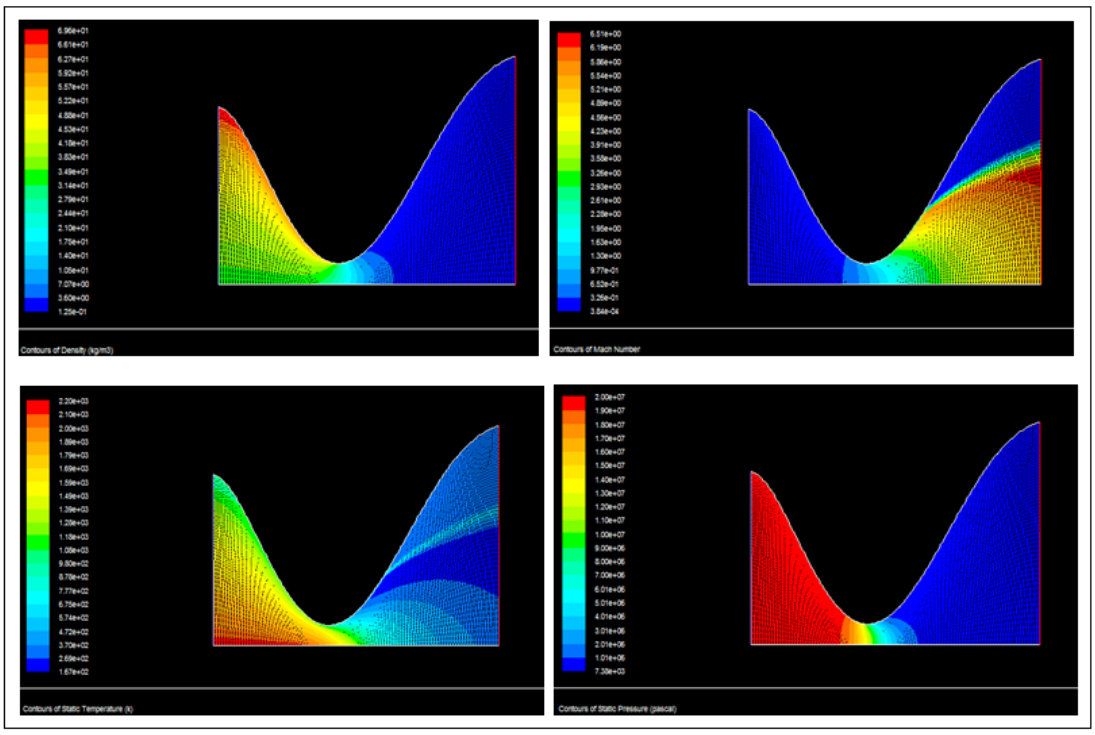

Figure 22: Contours of density, Mach number, static temperature and static pressure of the design 2453.

5.4. ADDITION OF NEW REAL DESIGNS

One way to improve the accuracy of response surfaces is by adding new real designs (increasing the number of high fidelity function evaluations). Their

performance tends to improve with the size of the design. This is especially true for Kriging method. Another way is by improving the quality of the real designs in terms of the distribution of the design variables within their own ranges. Although the range of each variable is specified, it may happen that some points be concentrated in a determined area instead of filling the entire range uniformly. Therefore, it is important to randomize the combination of each design defined by the six geometric design variables.

Figure 23 illustrates an example of the non-uniform distribution of the variables (x-location of the exit, inlet radius, exit radius, inlet angle and exit angle) related to the x-location of the inlet, in the design space. It can be observed that some areas have a high concentration of points while some spaces are empty. Most probably, this is the cause of errors in the response surfaces evaluations.

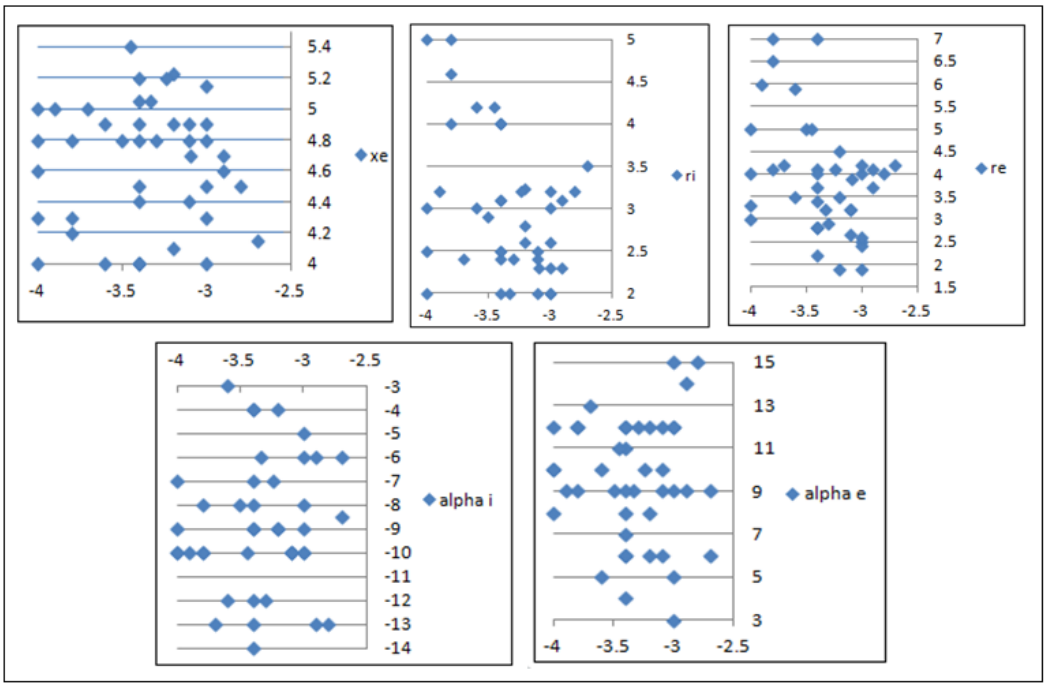

Figure 23: Non-uniform distribution of variables within the design space. The x-location of the exit (Xe), the inlet radius (Ri), the exit radius (Re), the inlet angle (alpha_i) and exit angle (alpha_e) are related to the x-location of the inlet (Xi).

In order to improve the accuracy of the response surfaces, 10 real designs were added to the 40 initial real designs. They were chosen in such a way to fill the gaps of the design spaces shown in Figure 24.

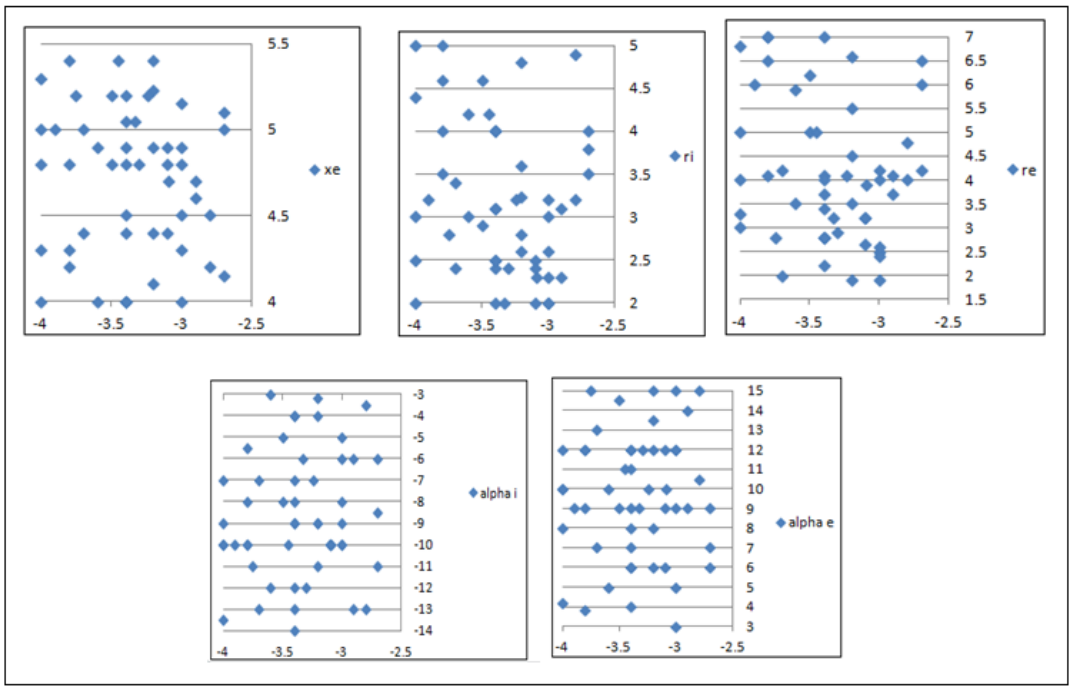

Figure 24: Non-uniform distribution of variables within the design space after the addition of 10 real designs. The x-location of the exit (Xe), the inlet radius (Ri), the exit radius (Re), the inlet angle (alpha_i) and exit angle (alpha_e) are related to the x-location of the inlet (Xi).

After creating the new response surfaces and running the optimization with the same algorithm (MOGA II) for 500 generations, 25,000 virtual designs were computed for each interpolation method (Kriging and RBF).

In order to validate the results, 2 designs from Kriging method and 1 from RBF were computed in ANSYS FLUENT. The input parameters of those designs are presented in Table 7. The comparison between the values from ANSYS FLUENT analysis and the ModeFrontier optimization are showed in Table 8.

Table 7: Pareto designs chosen and their design variables for Kriging and RBF methods.

ID in MF	X inlet	X exit	Inlet Radius	Exit Radius	Inlet angle	Exit angle
16303 (Kriging)	-2.9561	4.3217	4.1209	4.5218	-9.2470	13.7640
18394 (Kriging)	-2.7357	4.4164	3.7784	4.3676	-10.0710	14.4640
24680 (RBF)	-2.8638	4.3959	3.8301	3.9222	-9.9610	12.9330

Table 8: Errors related to density, Mach number and temperature standard deviations, between the virtual Pareto designs and the real simulations after the addition of 10 real designs.

ID in MF	ANSYS FLUENT			ModeFrontier			Error		
	Standard Deviations			Standard Deviations			Standard Deviations		
	Dens	Mach	Temp	Dens	Mach	Temp	Dens	Mach	Temp
16303 (Kriging)	0.4140	2.6952	86.2205	0.2104	2.386	55.166	49.18%	11.47%	36.02%
18394 (Kriging)	0.1502	1.3347	125.9933	0.034	1.1929	91.102	77.36%	10.63%	27.69%
24680 (RBF)	0.2151	1.2640	122.1156	0.0403	1.4544	81.851	81.29%	15.06%	32.97%

After analyzing Table 8, it is possible to observe that the errors of the standard deviations are still large. However, by analyzing the designs of Kriging method, it is notorious that the behaviors of the response surfaces are more stable with the addition of the 10 real designs. For the 3 designs, the Mach number standard deviations presented the smallest errors, while the density, presented the largest. Figure 25 presents the contours plots of design 24680.

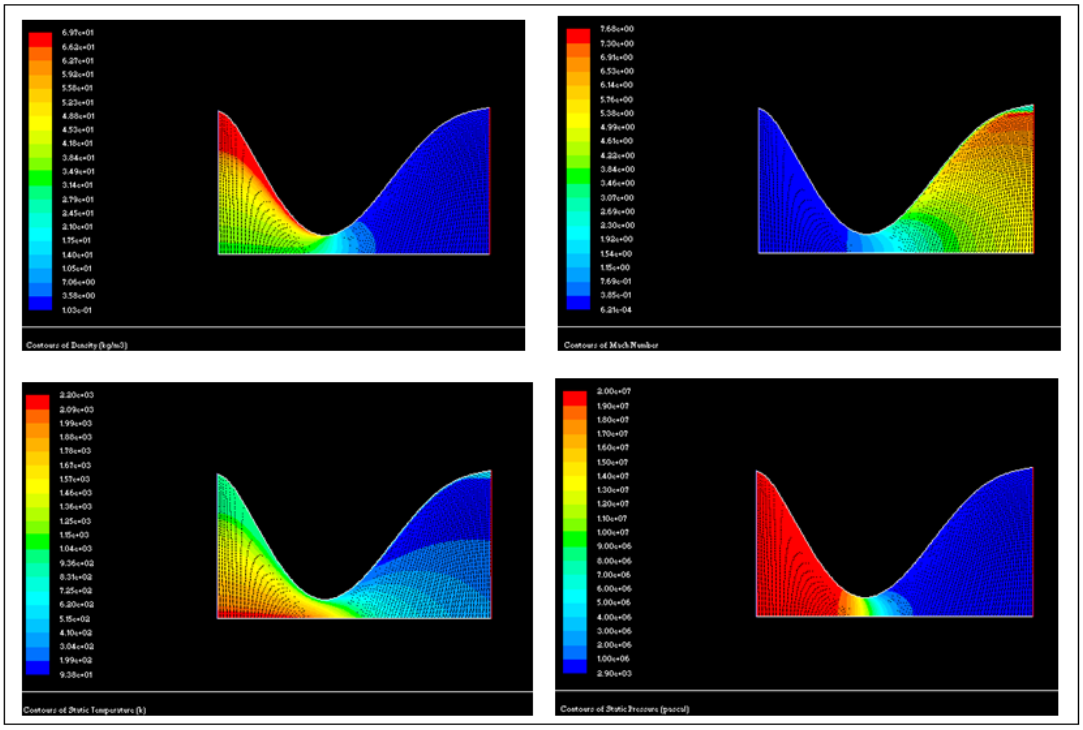

Figure 25: Contours of density, Mach number, static temperature and static pressure of the design 24680.

5.5. COMPARISON BETWEEN TWO OPTIMIZATION ALGORITHMS

In this section, the solution of the optimization of 50 real designs, using another evolutionary method, will be presented. Instead of MOGA II, the Multi-Objective Particle Swarm algorithm (MOPSO) will be utilized. The same procedure will be followed.

First, the response surfaces for the density, Mach number and temperature standard deviations were created from the 50 real designs using Kriging method (RBF was not used in this analysis). After that, the optimization was run for 500 generations, using MOPSO. The computing time in this case was much greater than in MOGA II. While in MOGA II the optimization run in around 1 minute, in the case of MOPSO it took 2 hours. The results of this optimization process are presented in Table 9 and Table 10.

Table 9: Pareto design 4343 and its design variables computed using MOPSO.

ID in MF	X inlet	X exit	Inlet Radius	Exit Radius	Inlet angle	Exit angle
4343 (Kriging)	-3.398	4.3737	3.2207	2.3775	-10.2260	11.1450

Table 10: Errors related to density, Mach number and temperature standard deviations, between the virtual Pareto design and the real simulation using MOPSO.

ID in MF	ANSYS FLUENT			ModeFrontier			Error		
	Standard Devations			Standard Devations			Standard Devations		
	Dens	Mach	Temp	Dens	Mach	Temp	Dens	Mach	Temp
4343	0.591904	0.715677	80.15203	0.699	1.4502	56.324	18.09%	50.65%	29.73%

By analyzing Table 10, it can be observed that, as in MOGA II, the errors of the standard deviations are large. However, by comparing the standard deviations calculated by ANSYS FLUENT, to the ones in Table 8, which represents the solutions using MOGA II, it can be observed that for Mach number and temperature, the standard deviations are smaller. This shows that the particle swarm algorithm is capable of converging further than a genetic algorithm, although both of them are able to find global minima.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

This research aimed to identify the optimal designs of converging-diverging supersonic and hypersonic nozzles that perform at maximum uniformity of thermodynamic and flow-field properties with respect to their average values at the nozzle exit. In order to solve this multi-objective design optimization problem, the parameters defining the shape of the nozzle were used as design variables. This work showed how the variation of such parameters influenced the nozzle exit flow non-uniformities.

Initially, 80 simulations were run, using two different turbulence models. The steps followed to find the solutions were as follows:

1- Generation of 40 real nozzle shapes using a Fortran code.

2- Generation of the computational mesh using a grid generator software package ANSYS GAMBIT.

3- Simulations of the flow-fields and thermal analysis were run in the CFD software package ANSYS FLUENT for the 40 shapes, using k-ε and k-ω turbulence models with a total of 80 simulations.

4- The standard deviations of the density, Mach number and temperature results at the exit of the nozzles were calculated and exported to the Optimization Software package, ModeFrontier.

5- Response surfaces were generated through Kriging and RBF methods, using the data from the real simulations.

6- The optimization was run using a multi-objective genetic algorithm.

7- The Pareto solutions (the optimal designs) from the optimization were validated by randomly choosing one of the virtual designs generated and using its

values for the design variables in ANSYS FLUENT and computing the solutions for density, Mach number and temperature standard deviations. In order to compare both solutions, the errors were calculated.

8- Ten more real designs were added to the 40 initial real designs population and the 50 real designs were optimized using genetic algorithm.

9- The Pareto solutions (the optimal designs) were validated and compared to the previous optimization (of 40 real designs).

10- A new optimization was run for the 50 real designs using Multi-Objective Particle Swarm algorithm (MOPSO) instead of genetic algorithm.

11- The results were validated and compared to the ones of MOGA II.

After this process, some conclusions were made. First, a comparison between the k-ε and k-ω turbulence models showed that there was not a great difference between the solutions for the two turbulence models. k-ω performed slightly better for density, while k-ε performed better for temperature and Mach number.

After that, a first analysis has shown that the response surfaces created could identify the behaviors of the real designs by using Kriging and RBF methods. However, after validating in ANSYS FLUENT four of the virtual shapes, it was observed that the errors calculated were large for density and Mach number and temperature standard deviations and the behavior of the response surfaces were unstable. One possibility to explain those large errors encountered is the fact that, in order for the response surfaces to present a good accuracy, entire range of each design variable must be well explored to fill the design space uniformly. If the opposite happens, the area which does not have a lot of points, will be badly interpolated and therefore, inaccurately represented by the response surfaces. Figure 23 showed that the distribution used was not uniform. One way to try solving this problem was to add

10 real designs to the initial population of 40 designs. However, it was shown that it did not affect the errors as much as expected, but improved the behavior of the response surfaces.

By running the optimization of the same problem, but using another optimization algorithm (particle swarm) instead of MOGA II, it could be observed that although the errors of the response surfaces were large, the values of Mach number and temperature standard deviations calculated in ANSYS FLUENT were smaller than those when optimization was done by using genetic algorithm. This can represent a better capacity of particle swarm to find the global minima.

By adding 10 real designs to the 40 initial ones, it was possible to see an improvement of the response surfaces. By adding more real designs, the accuracy of response surfaces can increase, since it will improve the quality of the interpolation and thus, the performance of the surrogate models. This can be done in a future work. It is important to emphasize that not only the numbers of real designs will influence, but also the distribution of the design variables values in the design space. It should be as uniform as possible. In order to get this uniformity, a Sobol`s algorithm [23] can be used to generate the design variables randomly. Also, other response surface algorithms (such as RBF polynomial method [14], [17]) could be tested to try to get better results. Hybrid multi-objective optimization, which was explained in this thesis although it has not been used, can also be tested. Since it is a combination of the deterministic and the evolutionary/stochastic methods, in the sense that it utilizes the advantages of each of these methods, the results of this challenging optimization problem may be better.

CHAPTER VII

REFERENCES

[1] Colaço M. J., Silva W. B., Magalhães A. C., Dulikravich G. S., "Response Surface Methods Applied to Scarce and Small Sets of Training Points – A Comparative Study", EngOpt 2008 - International Conference on Engineering Optimization, (ed: Herskovits, J.), Rio de Janeiro, Brazil, June 1-5, 2008.

[2] Costa J.P., Pronzato L., Thierry E.: A Comparison Between Kriging and Radial Basis Function Networks for Nonlinear Prediction. NSIP 1999: 726-730.

[3] Doty, John H., Performance Prediction and Design of Maximum Thrust Planar Supersonic Nozzles Using a Flux-Difference-Splitting Technique, PHD Dissertation. Purdue University, Aug 1991.

[4] Doty, John H., Thompson, H. Doyle, and Hoffman, Joe D., Optimum Thrust Two-Dimensional NASP Nozzle Study, NASP CR1069, Nov 1989.

[5] Doty, John H., Thompson, H. Doyle, and Hoffman, Joe D., Optimum Thrust Airframe Integrated Scramjet Nozzles, Paper #83, Seventh National Aero-Space Plane Technology Symposium, Oct 1989.

[6] Snelling, Sandra L., Effect of Non-Uniform Entrance Flow Profile on Hypersonic Nozzle Pitching Moment, Master Thesis. The Air Force Institute of Technology, Air University, Dec 1991.

[7] Palma, P.C., Danehy, P.M., Houwing, A.F.P.: Fluorescence imaging of rotational and vibrational temperature in a shock tunnel nozzle flow. AIAA Journal 41(9), 1722–1732 (2003).

[8] Gai, S.L.: Free piston shock tunnels: Developments and capabilities. Progress in the Aerospace Sciences 29, 1–41 (1992).

[9] O'Byrne, S., Danehy, P.M., Houwing, A.F.P: Investigation of Hypersonic Nozzle Flow Uniformity Using NO Fluorescence. Shock Waves Journal, vol. 15, no. 2, pp. 81-87 (2006).

[10] Tu, J., Yeoh, G.H., Liu, C.: Computational Fluid Dynamics: a Practical Approach, 1st edition, Butterworth-Heinemman, 2008.

[11] ANSYS, Inc, "ANSYS FLUENT 12.0, Theory Guide", April 2009.

[12] Davidson L. "An Introduction to Turbulence Models", Department of Thermo and Fluid Dynamics, Chalmers University of Technology, Goteborg, Sweden, January 2003.

[13] Stuhne, G.R., Peltier, W.R.: A robust unstructured grid discretization for 3-dimensional hydrostatic flows in spherical geometry: A new numerical structure for ocean general circulation modeling. Journal of Computational Physics Volume 213, Issue 2, 10 April 2006, Pages 704–729.

[14] Colaço, J. M. and Dulikravich, G. S., "A Survey of Basic Deterministic, Heuristic and Hybrid Methods for Single-Objective Optimization and Response Surface Generation", Chapter 10 in Thermal Measurements and Inverse Techniques, (eds: Orlande, H. R. B., Fudym, O., Maillet, D. and Cotta, R.), Taylor & Francis, May 2011, pp. 355-406.

[15] Goldberg, D.E. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, Reading, MA.

[16] Deb, K. 2002. Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, New York.

[17] Colaço M. J., Dulikravich G. S., Sahoo D., "A Response Surface Method-Based Hybrid Optimizer", Inverse Problems in Science and Engineering, Vol. 16, issue 6, 2008, pp. 717-741.

[18] Moral, R. J., Dulikravich, G.S.: "Multi-Objective Hybrid Evolutionary Optimization Utilizing Automatic Algorithm Switching." AIAA Journal, Vol. 46, No. 3, March 2008, pp 673-700.

[19] Jeong, S., Murayama, M., Yamamoto, K.: Efficient Optimization Design Method Using Kriging Model. Journal of Aircraft, Vol. 42, No. 2, March–April 2005.

[20] Davis, J.C. Statistics and Data Analysis in Geology. John Wiley & Sons, New York, 1986.

[21] FLUENT, Inc, "GAMBIT Documentation", April 2000.

[22] http://www.esteco.com/home/mode_frontier/mode_frontier.html

[23] Sobol, I. M.: Uniformly distributed sequences with an additional uniform property, USSR Computational Mathematics and Mathematical Physics (16) 236-242, 1976.

APPENDICES

A. FIFTY REAL DESIGNS INPUTS AND STANDARD DEVIATIONS CALCULATED IN ANSYS FLUENT

Shape	X inlet	X exit	Inlet Radius	Exit Radius	Inlet angle	Exit angle	Standard deviation		
							Dens	Mach Number	Temp
1	-4	5	3	5	-10	10	0.3662	2.5994	75.2607
2	-3	4	3	2.5	-5	5	0.5814	0.7320	105.1199
3	-3.8	4.2	4	6.5	-8	12	0.4158	2.6510	57.0631
4	-4	4.8	5	4	-9	12	0.2007	0.9256	107.9418
5	-3.7	5	2.4	4.2	-13	13	0.3987	2.6361	77.2453
6	-3.9	5	3.2	6	-10	9	0.4125	2.6141	52.3236
7	-3.4	4	4	2.8	-12	12	0.4160	2.6156	54.8807
8	-3.4	4	2.5	3.7	-14	8	0.3553	2.4983	79.6133
9	-3.5	4.8	2.9	5	-8	9	0.4129	2.6144	55.5327
10	-3.6	4	3	3.5	-3	5	0.3689	2.5223	72.1669
11	-3.8	4.8	5	7	-10	9	0.4035	2.4738	54.0524
12	-3.6	4.9	4.2	5.9	-12	10	0.4309	2.7005	53.5624
13	-3.2	4.1	2.6	1.9	-9	12	0.8701	0.6404	70.6060
14	-3.4	4.9	4	7	-7	6	0.4137	2.5533	52.5845
15	-4	4.3	2.5	3.3	-7	8	0.3146	2.3228	93.9024
16	-3.4	4	4	2.8	-13	11	0.5124	0.9127	119.6306
17	-3.8	4.3	4.6	4.1	-10	12	0.4070	2.6639	71.4028
18	-3.4	4.4	2.4	4.1	-9	12	0.3821	2.5850	72.5221
19	-3	4.5	2.3	1.9	-10	12	0.8587	0.7491	73.8689
20	-3.1	4.9	2.5	3.2	-10	12	0.4618	0.9310	95.7283
21	-3.1	4.8	2	3.2	-10	6	0.3073	0.6823	95.9651
22	-3.4	4.8	2.5	2.2	-8	9	0.7077	0.6709	73.9122
23	-2.7	4.15	3.5	4.2	-6	9	0.1747	1.1550	117.4955
24	-3.2	5.23	3.24	4.5	-9	8	0.1557	0.9950	96.5519
25	-2.9	4.6	2.3	4.1	-6	9	0.3980	2.6038	75.5317
26	-3.24	5.2	3.2	4.1	-7	10	0.1843	1.0161	100.2541

27	-3	4.3	2	2.4	-6	9	0.4547	0.5436	88.3194
28	-3.45	5.4	4.2	5	-10	11	0.4544	2.9048	80.0762
29	-3.4	4.5	3.1	3.4	-4	6	0.3236	0.8333	97.0224
30	-3.09	4.7	2.3	3.9	-10	10	0.2240	0.9564	101.7904
31	-3.3	4.8	2.4	2.9	-12	12	0.3052	0.5494	85.4182
32	-3.2	4.9	2.8	3.5	-4	6	0.3260	0.8423	95.1974
33	-2.9	4.7	3.1	3.7	-13	14	0.2047	0.9922	104.1040
34	-3.1	4.4	2.4	2.65	-10	9	0.4615	0.5769	87.3665
35	-3	4.8	3.2	4	-9	12	0.1818	1.0289	103.3060
36	-3.4	5.2	2	2.8	-4	7	0.3571	0.5758	87.6335
37	-2.8	4.5	3.2	4	-13	15	0.1684	1.0223	102.2231
38	-3	4.9	2	2.6	-10	15	0.3249	0.5645	80.9213
39	-3.4	5.05	3.1	4	-9	4	0.2195	0.8941	103.5500
40	-3	5.15	2.6	4.2	-8	3	0.2409	1.1435	110.4944
41	-2.7	5	4	6	-8.5	6	0.4652	2.9439	69.7593
42	-2.7	5.1	3.8	6.5	-11	7	0.4640	2.9172	63.6838
43	-3.2	4.4	4.8	5.5	-3.2	15	0.4375	2.7720	61.9109
44	-3.8	5.4	3.5	7	-5.5	3.8	0.4002	2.5283	49.0601
45	-3.7	4.4	3.4	2	-7	7	0.9856	0.8323	81.7401
46	-3.2	5.4	3.6	6.6	-11	13.5	0.4597	2.9051	59.7186
47	-2.8	4.2	4.9	4.8	-3.5	10.5	0.4379	2.7732	77.6269
48	-4	5.3	4.4	6.8	-13.5	4.2	0.4470	2.8170	56.0074
49	-3.5	5.2	4.6	6.2	-5	14.5	0.4580	2.8729	62.0783
50	-3.75	5.2	2.8	2.8	-11	15	0.3031	0.5280	81.5569

B. FORTRAN CODE USED TO CREATE THE NOZZLE SHAPES

```
parameter (idm=201)
    IMPLICIT REAL*8 (A-H,O-Z)
    dimension mat(4,4),b(4),coeff(6),XX(idm),RR(IDM),AA(idm)
    dimension xmach(idm),temperature(idm),density(idm),pressure(idm)
    dimension radius(idm)
     real*8 k,mat
C
C.... OPEN OUTPUT FILE FOR FINAL COMPUTATIONAL VALUES
     OPEN(UNIT =12,FILE = 'Q1D-output.dat',status='unknown')
     OPEN(UNIT =30,FILE = 'Q1D-X-coord.dat',status='unknown')
     OPEN(UNIT =31,FILE = 'Q1D-Radius.dat',status='unknown')
     OPEN(UNIT =32,FILE = 'Q1D-Area.dat',status='unknown')
     OPEN(UNIT =34,FILE = 'Q1D-Machisent.dat',status='unknown')
     OPEN(UNIT =36,FILE = 'Q1D-TT01isent.dat',status='unknown')
     OPEN(UNIT =38,FILE = 'Q1S-RR01isent.dat',status='unknown')
     OPEN(UNIT =40,FILE = 'Q1D-PP01isent.dat',status='unknown')
     OPEN(UNIT =42,FILE = 'Q1D-Machshock.dat',status='unknown')
     OPEN(UNIT =44,FILE = 'Q1D-TT01shock.dat',status='unknown')
     OPEN(UNIT =46,FILE = 'Q1D-RR01shock.dat',status='unknown')
     OPEN(UNIT =48,FILE = 'Q1D-PP01shock.dat',status='unknown')
    OPEN(UNIT =50,FILE = 'Q1D-
SHOCKDATA.DAT',STATUS='UNKNOWN')
    OPEN(UNIT =51,FILE = 'Q1D-x-r-grid.DAT',STATUS='UNKNOWN')

     write(*,*)' Finding location of a shock in a choked diffuser for'
    write(*,*)' a specified exit static press./inlet stagnat. press.'
    write(*,*)'Program put together by Prof. George S. Dulikravich'
    write(*,*)'August 4, 2012,  Florida International University'
     write(*,*)
     write(12,*)' Finding location of a shock in a choked diffuser for'
    write(12,*)' a specified exit static press./inlet stagnat. press.'
    write(12,*)'Program put together by Prof. George S. Dulikravich'
    write(12,*)'August 4, 2012,  Florida International University'
     write(12,*)

     write(*,*)' Enter specific heat ratio "k" for the gas'
     read(*,*)   k
     write(12,*)' Specific heat ratio: ',k

    pi   = acos(-1.0000000000000000)
    twopi= 2.00000000000000*pi
    gp1  = k + 1.0
    gm1  = k - 1.0
    gp1h = gp1/2.0
    gm1h = gm1/2.0
    EX   = -gp1h/gm1
    BLA1 = 2.0/(K + 1.0000000)
```

```
    GEX  = (K + 1.0000)/(K - 1.00000)

    write(*,*) 'Input for a 2D conv./diverg. nozzle shape generator:'
    write(*,*) 'COORDINATE ORIGIN IS AT THE CENTER OF THE
THROAT'
    write(*,*)'Number of grid cells desired along the nozzle (ICELLS)'
     write(*,*) '(important:  ICELLS < idm)'
    read(*,*)   ICELLS
     write(*,*) 'x-location of the inlet (must be negative) = '
     read(*,*) xin
     write(*,*) 'x-location of the exit (must be positive) = '
     read(*,*) xex
c------ echo
    write(*,*) 'Input for a 2D conv./diverg. nozzle shape generator:'
    write(*,*) 'COORDINATE ORIGIN IS AT THE CENTER OF THE
THROAT'
    write(*,*)'Number of grid cells desired along the nozzle (ICELLS)'
     write(*,*) '(important:  ICELLS < idm)'
    write(*,*)   ICELLS
     write(*,*) 'x-location of the inlet (must be negative) = '
     write(*,*) xin
     write(*,*) 'x-location of the exit (must be positive) = '
     write(*,*) xex

    If ((icells+1) .gt. idm) then
    write(*,*) 'You must make IDM larger than ICELLS'
    stop
    endif

     XTH    = 0.0000000000000
     CELLS1 = ICELLS*((XTH - XIN)/(XEX - XIN))
    ICELLS1= CELLS1
    ICELLS2= ICELLS - ICELLS1
    IMAX1  = ICELLS1 + 1
    IMAX2  = ICELLS2 + 1
    IMAX   = ICELLS  + 1
C  WRITE(*,*) 'IMAX1, IMAX2, IMAX', IMAX1,IMAX2,IMAX
         dx1 = (xth - xin)/float(IMAX1-1)
     do 10 i = 1,IMAX1
    XX(I) = xin + (i-1)*dx1
10    continue
         dx2 = (xex - xth)/float(IMAX-IMAX1)
     do 20 i = IMAX1,IMAX
    XX(I) = xth + (i-IMAX1)*dx2
20    continue

     write(*,*) 'Enter index choosing nozzle shape creation method:'
    write(*,*) 'GGG=1 use 5th order polynomial to fit input values'
    write(*,*) 'GGG=2 use area-Mach number relation to create shape'
```

```
      read(*,*) GGG
      write(12,*) 'Enter index choosing nozzle shape creation method:'
     write(12,*) 'GGG=1 use 5th order polynomial to fit input values'
     write(12,*) 'GGG=2 use area-Mach number relation to create shape'
      write(12,*) GGG

C----- determine coefficients of 5th order polynomial defining nozzle
      IF(GGG.EQ.1) then
      write(*,*) 'inlet radius of the nozzle = '
      read(*,*) radin
      write(*,*) 'throat radius of the nozzle = '
      read(*,*) radth
      write(*,*) 'exit radius of the nozzle = '
      read(*,*) radex
      write(*,*) 'inlet angle(deg) of the wall (negative and <|75|)'
      read(*,*) angin
      write(*,*) 'exit angle(deg) of the wall (positive and <|75|)'
      read(*,*) angex
c--echo of the inputted data
      write(12,*) radin
      write(12,*) 'throat half-width of the nozzle = '
      write(12,*) radth
      write(12,*) 'exit half-width of the nozzle = '
      write(12,*) radex
      write(12,*) 'inlet angle(deg) of the wall (negative and <|75|)'
      write(12,*) angin
      write(12,*) 'exit angle(deg) of the wall (positive and <|75|)'
      write(12,*) angex

            tanin = tan((angin/180.0000000000000)*pi)
        tanex = tan((angex/180.000000000000)*pi)

c   solves for the 5th order equation describing the nozzle shape
            coeff(1) = radth
            coeff(2) = 0.0
c   Load mat
            mat(1,1) = xin*xin
            mat(2,1) = xex*xex
            mat(3,1) = 2.0*xin
            mat(4,1) = 2.0*xex
            mat(1,2) = xin**3
            mat(2,2) = xex**3
            mat(3,2) = 3.0*(xin*xin)
            mat(4,2) = 3.0*(xex*xex)
            mat(1,3) = xin**4
            mat(2,3) = xex**4
            mat(3,3) = 4.0*(xin**3)
            mat(4,3) = 4.0*(xex**3)
            mat(1,4) = xin**5
```

```
                mat(2,4) = xex**5
                mat(3,4) = 5.0*(xin**4)
                mat(4,4) = 5.0*(xex**4)
c   Load b
                b(1) = radin - coeff(1)
                b(2) = radex - coeff(1)
                b(3) = tanin
                b(4) = tanex
C------ solve a 5x5 matrix to find coefficients of 5th order polynomial

    CALL GAUSSJ(mat,4,4,b,1,1)

                coeff(3) = b(1)
                coeff(4) = b(2)
                coeff(5) = b(3)
                coeff(6) = b(4)

     do 40 i = 1,imax
        x = xx(i)
                r = 0.0000000000
    do 30 n=1,6
                r = r + coeff(n)*(X**(n-1))
30    continue
    rr(i)    = r
     AA(i)    = r*r*pi
40    continue
       endif

C------use area-Mach number relation to find nozzle shape
     IF(GGG.EQ.2) then
C-------------------------------------------------------------------------
     write(*,*) 'inlet Mach number'
     read(*,*) EMin
     write(*,*) 'exit Mach number = '
     read(*,*) EMex
    write(*,*) 'inlet-to-throat clustering amplitude (0.0 < 0.9)'
    read(*,*) AMP1
    write(*,*) 'throat-to-exit clustering amplitude (0.0 < 0.9)'
    read(*,*) AMP2
c--echo of the inputted data
     write(*,*) 'inlet Mach number = '
     write(*,*) EMin
     write(*,*) 'exit Mach number = '
    write(*,*) EMex
    write(*,*) 'inlet-to-throat clustering amplitude (0.0 < 0.9)'
    write(*,*) AMP1
    write(*,*) 'throat-to-exit clustering amplitude (0.0 < 0.9)'
    write(*,*) AMP2
```

```
      BLA1= 2.0/(K + 1.0000000)
       GM1H= (K - 1.0000000)/2.000000000
      GEX = (K + 1.0000)/(K - 1.00000)
      EMTH= 1.00000000000
         DO 45 I=1,IMAX1
      XBAR = (XX(I) - XIN)/(XTH - XIN)
      BLA  = XBAR - (AMP1/TWOPI)*SIN(TWOPI*XBAR)
      EM   = EMIN + (EMTH - EMIN)*BLA
      EM2  = EM*EM
      A2AT2= (BLA1*(1.0 + GM1H*EM2))**GEX
C        WRITE(*,*)
'I,XX(I),XBAR,BLA,EM,A2AT2',I,XX(I),XBAR,BLA,EM,A2AT2
      A    = SQRT(A2AT2/EM2)
      RR(I)= SQRT(A/PI)
      AA(I)= A
45    CONTINUE
         DO 50 I=IMAX1,IMAX
      XBAR = (XX(I) - XTH)/(XEX - XTH)
      EM   = EMTH + (EMEX - EMTH)*(XBAR-
(AMP2/TWOPI)*SIN(TWOPI*XBAR))
      EM2  = EM*EM
      A2AT2= (BLA1*(1.0 + GM1H*EM2))**GEX
C        WRITE(*,*) 'I,XX(I),EM,A2AT2',I,XX(I),EM,A2AT2
      A    = SQRT(A2AT2/EM2)
      RR(I)= SQRT(A/PI)
      AA(I)= A
50    CONTINUE
         endif

       AINLET = AA(1)
      ATHROAT= AA(IMAX1)
      AEXIT  = AA(IMAX)
      WRITE(*,*) 'AINLET  =',AINLET
      WRITE(*,*) 'ATHROAT =',ATHROAT
      WRITE(*,*) 'AEXIT   =',AEXIT
      WRITE(12,*) 'AINLET  =',AINLET
      WRITE(12,*) 'ATHROAT =',ATHROAT
      WRITE(12,*) 'AEXIT   =',AEXIT
      write(12,*)'There are',IMAX,' points along the nozzle'
      write(12,*) '     xnozzle                rnozzle  '
       WRITE(30,*) IMAX,IMAX
      WRITE(31,*) IMAX,IMAX
      WRITE(32,*) IMAX,IMAX

       DO 55 I=1,IMAX
       write(12,*) XX(I),RR(I),I
      write(30,*) XX(I)
      write(31,*) RR(I)
      write(32,*) AA(I)
```

```
55    continue

      write(*,*)' Enter minimum desired exit pressure ratio pB1/p01'
     write(*,*)'(0.0 < pB1/p01 < 1.0; typically 0.2 to 0.9)'
      read(*,*)   pB1P01
     write(12,*)'Minimum desired exit pressure ratio PB1/P01=',PB1P01
      write(*,*)' Enter maximum desired exit pressure ratio pB2/po1'
     write(*,*)'(0.0 < pB2/p01 < 1.0; typically 0.2 to 0.9)'
     write(*,*)'NOTE: pb2/p01 must be grater than pB1/p01'
     read(*,*)   pB2P01
      write(12,*)'Maximum desired exit pressure ratio PB2/P01=',PB2P01
     write(*,*)' Enter increments in exit pressure ratio to analyze'
      read(*,*)   DpBP01
      write(12,*)'Increments in exit pressure ratio to analyze=',DPBP01
C
      write(12,*)
     write(12,*)'ONE OF THE FOLLOWING MESSAGES MIGHT
EVENTUALLY APPEAR'
      write(12,*)' XSHOCK < XIN means nozzle is entirely subsonic'
      write(12,*)' (that is, back pressure is too high to choke nozzle)'
      write(12,*)
      write(12,*)' XSHOCK > XEX means fully supersonic nozzle case'
      write(12,*)' (that is, shock is outside of nozzle)'
      write(12,*)
      write(*,*)
     write(*,*)'ONE OF THE FOLLOWING MESSAGES MIGHT
EVENTUALLY APPEAR'
      write(*,*) ' XSHOCK < XIN means nozzle is entirely subsonic'
      write(*,*) '(that is, back pressure is too high to choke nozzle)'
      write(*,*)
      write(*,*) ' XSHOCK > XEX means fully supersonic nozzle case'
      write(*,*) ' (that is, shock is outside of nozzle)'
      write(*,*)
     WRITE(12,*)'XSHOCK = CONVERGED LOCATION OF THE NORMAL
SHOCK'
     WRITE(12,*)'PBP01= EXIT STATIC PRESSURE/INLET STAGNATION
PRESSURE'
     WRITE(12,*)'T2T1=RATIO OF ABSOLUTE TEMPERATURES ACROSS
THE SHOCK'
     WRITE(12,*)'R2R1=RATIO OF GAS DENSITIES ACROSS THE SHOCK'
     WRITE(12,*)'P2P1=RATIO OF STATIC PRESSURES ACROSS THE
SHOCK'
     WRITE(12,*)'EM1=MACH NUMBER COMPUTED JUST UPSTREAM OF
THE SHOCK'
     WRITE(12,*)'EM2=MACH NUMBER COMPUTED JUST DOWNSTREAM
OF THE SHOCK'
     WRITE(12,*)'EMEX=MACH NUMBER COMPUTED AT THE NOZZLE
EXIT'
```

```
    WRITE(12,*)'DELSR=CHANGE IN ENTROPY ACROSS THE NORMAL
SHOCK/
    # GAS CONSTANT'

     write(12,*)
     WRITE(12,*) '       CALCULATED VALUES ALONG THE NOZZLE'
    WRITE(12,*) 'USING AREA - MACH NUMBER (ISENTROPIC
RELATIONS ONLY)'
    WRITE(12,*) ' NOTICE: THIS WILL GIVE A LOWER ENVELOPE P/P01
CURVE'
     WRITE(12,60)
60
FORMAT(5x,1HI,7x,1HX,8x,6HRADIUS,4x,4HAREA,9x,4HMACH,6x,5HT/T
01,6x,
    #       5HR/R01,6x,5HP/P01)
C234567890123456789012345678901234567890123456789012345678901234
6789012
C-----------------------------------------------------------------------
C------ Use second order Newton (or Housdorfer) iterative method to find
C------ local Mach number values from the local cros-sectional nozzle area
C------ values (from Area - Mach number relation). NOTICE: This assumes
C------ no shock in the nozzle and isentropic flow throughout
    write(34,*) imax,imax
    write(36,*) imax,imax
    write(38,*) imax,imax
    write(40,*) imax,imax

     emc    = 0.1

      do 100 i =1,imax
     X      = xx(i)
    R      = RR(I)
    A      = AA(I)
     AC     = A/ATHROAT
    if(x .ge. 0.0001) emc = 1.1*emc

     DO 65 ITR=1,1000
     BLA    = 1.0D+00 + gm1h*EMC*EMC
     BLAD   = 1.0D+00 + gm1h
    G     = (BLAD/BLA)**EX
       F     = G/EMC - AC
     H     = gp1h/BLA - 1.0D+00/(EMC*EMC)
       FP    = G*H
    HH    = (gp1h/(BLA*BLA))*EMC*((1.0D+00 - K)+gp1h)
    HHH   = -
(gp1h/BLA)*(1.0D+00+1.0D+00/EMC)+2.0D+00/(EMC*EMC*EMC)
      FPP   = G*(HH+HHH)
    FNC   = F/FP
    FNCP  = 1.0D+00 -(F*FPP)/(FP*FP)
```

```
      DMC   = FNC/FNCP
C     if(x .gt. 0.0001) then
      FNCP  = 1.0D+00 - fnc*(fpp/(2.0d+00*fp))
      DMC   = FNC*FNCP
C     endif
      EMCO  = EMC
      EMC   = EMC - DMC
c     WRITE(12,49) ITR,F,FP,DMC,EMC
c49    FORMAT(4HITR=,I5,4H  F=,E15.6,5H  FP=,E15.6,6H  DMC=,
c     #       E15.6,5H EMC=,F12.6)
C       if ((Dabs(f).lt.0.00000001).or.(Dabs(fp).lt.0.00000001)) goto 75
       IF(DABS(DMC).LT.0.00000001) GO TO 75
65     CONTINUE
75     continue
      tt01  = 1.0000000/(1.00000000000000000 + gm1h*emc*emc)
      rr01  = tt01**(1.00000000000000/gm1)
      pp01  = rr01**k
       xMach(i)       = emc
       Temperature(i) = tt01
       Density(i)     = rr01
       Pressure(i)    = pp01
       WRITE(12,110) I,X,R,AC,EMC,TT01,RR01,PP01
100    continue
110    FORMAT(1X,I5,7f11.4)
111    format(1x,f15.6)

c------ Save subsonic-to supersonic isentropic flow along the nozzle
      write(34,111) (xMach(i),i=1,imax)
      write(36,111) (temperature(i),i=1,imax)
      write(38,111) (density(i),i=1,imax)
      write(40,111) (pressure(i),i=1,imax)

       write(12,*)
       write(12,*) 'Following values were computed at',imax,' locations'
      write(12,*) 'from inlet to exit of the entire nozzle for the'
      write(12,*) 'case of ISENTROPIC flow and saved in these files:'
      write(12,*) 'Q1D-Machisent.dat (local Mach number)'
      write(12,*) 'Q1D-TT01isent.dat (local Temp./inlet stagnat. temp.)'
      write(12,*) 'Q1D-RR01isent.dat (local Dens./inlet stagnat. dens.)'
      write(12,*) 'Q1D-PP01isent.dat (local Pres./inlet stagnat. pres.)'

C*************************************************************
****C
C23456789012345678901234567890123456789012345678901234567890123456
6789012
c$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$

C----- main loop starts here TO FIND A POSSIBLE SHOCK IN THE NOZZLE
```

```
    WRITE(12,*)
     WRITE(12,*)'Main loop starts here TO FIND A POSSIBLE SHOCK'
    WRITE(*,*)
     WRITE(*,*) 'Main loop starts here TO FIND A POSSIBLE SHOCK'
    WRITE(50,120)
120
FORMAT(2X,4HITER,3X,6HXSHOCK,2X,5HPBP01,3X,4HP2P1,4X,4HR2R1,
    #       4X,4HT2T1,5X,3HEM1,5X,3HEM2,5X,4HEMEX,3X,5HDELSR)
C     Guess a location at which the shock is located

        do 500 pBP01 = pB1P01,pB2P01,dpBP01

     XSHOCK = 0.001
     EM1    = 1.01

     write(12,125) pbp01
125   format(//,'CASE WHEN EXIT PRESS/INLET STAGNATION
PRESS=',F10.4)
    write(12,126)
126   format(1x,4Hiter,9x,3Hem1,11x,3Hem2,11x,4Hemex,9x,5Hpep01,
    #       8x,6Hxshock)
     iter   = 0
     nsign  = 0
     step   = 0.01

150    continue
     AD = 1.000000000

       IF(GGG.EQ.1) THEN
              r = 0.0000000000
    do 160 n=1,6
              r = r + coeff(n)*(XSHOCK**(n-1))
160   continue
       ENDIF

       IF(GGG.EQ.2) THEN
     do 165 i=imax1+1,imax
    if(xx(i).lt.xshock) go to 164
    ishock = i
    go to 166
164   continue
165   continue
166   continue
     rbar = (xshock - xx(i-1))/(xx(i) - xx(i-1))
    r = rr(i-1) + rbar*(rr(i)-rr(i-1))
      endif
     AC  = r*r*pi/ATHROAT
     EMD = 1.0000000000000
     EMC = EM1
```

```
c      write(*,*) 'rbar,r,i',rbar,r,i
C      Calculate Mach number ahead of the guessed shock point (eq.159)
       ICNT1 = 1
170    continue
       BLA   = 1.0D+00 + gm1h*EMC*EMC
       BLAD  = 1.0D+00 + gm1h*EMD*EMD
      G     = (BLAD/BLA)**EX
        F     = AD*G*(EMD/EMC) - AC
       H     = gp1h/BLA-1.0D+00/(EMC*EMC)
        FP    = AD*G*EMD*H
      HH    = (gp1h/(BLA*BLA))*EMC*((1.0D+00 - K)+gp1h)
      HHH   = -(gp1h/BLA)*(1.0D+00 +
1.0D+00/EMC)+2.0D+00/(EMC*EMC*EMC)
       FPP   = AD*G*EMD*(HH+HHH)
      FNC   = F/FP
      FNCP  = 1.0D+00 -(F*FPP)/(FP*FP)
      DMC   = FNC/FNCP
       if ((Dabs(f).lt.0.00000001).or.(Dabs(fp).lt.0.000000001)) goto 175
      FNCP  = 1.0D+00 - fnc*(fpp/(2.0d+00*fp))
      DMC   = FNC*FNCP
      EMCO  = EMC
      EMC   = EMC - DMC
c      write(12,177) ICNT1,F,FP,FPP,FNC,FNCP,DMC,EMC
      ICNT1  = ICNT1 + 1
        goto 170
175    continue
c      write(*,*) 'icnt1,emc=',icnt1,emc
       EM1 = EMC
      em12= em1*em1
177   FORMAT(I5,7E14.5)
C      Find the Mach number after a guessed shock point with the initial
C      Mach number, M1.
c      EM2 = sqrt((EM12+2.00000/gm1)/(2.00000*k*EM12/gm1-1.00000))
       em2 = sqrt((1.00000 + gm1h*em12)/(k*em12-gm1h))
C      Find stagnation pressure, p02, (after the shock)
C      as a ratio to p01, (the initial stagnation pressure)
       G     = (2.00000*k*EM12/gp1-gm1/gp1)**(1.00000/gm1)
       H     = ((1.00000 + gm1h*EM12)/(gp1h*EM12))**(k/gm1)
       P02P01= 1.00000/G/H

C      Find the Mach number at the exit of the nozzle
       AD  = AC
       AC  = AEXIT
       EMD = EM2
       EMC = 0.95*EM2
c      write(*,*) 'em2,p02p01,emc',em2,p02p01,emc
C------ Second order Newton-Raphson iterative method to find M-exit
       ICNT2 = 1
200    continue
```

```
      BLA   = 1.0D+00 + gm1h*EMC*EMC
      BLAD  = 1.0D+00 + gm1h*EMD*EMD
     G     = (BLAD/BLA)**EX
       F     = AD*G*(EMD/EMC) - AC
      H     = gp1h/BLA-1.0D+00/(EMC*EMC)
       FP    = AD*G*EMD*H
     HH    = (gp1h/(BLA*BLA))*EMC*((1.0D+00 - K)+gp1h)
     HHH   = -(gp1h/BLA)*(1.0D+00 +
1.0D+00/EMC)+2.0D+00/(EMC*EMC*EMC)
      FPP   = AD*G*EMD*(HH+HHH)
     FNC   = F/FP
     FNCP  = 1.0D+00 -(F*FPP)/(FP*FP)
     DMC   = FNC/FNCP
      if ((Dabs(f).lt.0.00001).or.(Dabs(fp).lt.0.0000001)) goto 250
c   FNCP  = 1.0D+00 - fnc*(fpp/(2.0d+00*fp))
c   DMC   = FNC*FNCP
     EMCO = EMC
     EMC  = EMC - DMC
     ICNT2= ICNT2 + 1
c      write(12,177) ICNT2,F,FP,FPP,FNC,FNCP,DMC,EMC
       goto 200
250    continue
      EMEX = EMC
C     Find the pressure at the nozzle exit as a ratio to
C     p01, the initial stagnation pressure, using eq. (149)
      PEP01= P02P01*(1.0d+00 + gM1h*EMEX*EMEX)**(k/(1.0d+00-k))
C       write(12,*) 'iter,xshock,em1,em2,emex,pep01'
C  write(12,255) iter,xshock,em1,em2,emex,pep01
C255    format(i5,5f15.6)

      iter = iter + 1
c      write(12,177) iter,Xshock,pEp01,pBp01,EM1,EM2,EMex
C     Compare this pressure ratio, p03, to the given back pressure
C     ratio, pB.  If the two are not equal, repeat the entire
C     procedure starting with a new location for the shock.
      if (Dabs(PEP01-pBP01).lt.0.000001) goto 275
      Esign = nsign
      if (pEp01.gt.pBP01) then
        nsign = 1
      else
        nsign = -1
      end if
      if (nsign.eq.Esign*(-1)) step = step/10.0
      XSHOCK = XSHOCK + nsign*step

    write(12,251) iter,em1,em2,emex,pep01,xshock
251   format(i5,5F14.4)
```

```
      if (XSHOCK.gt.XEX) then
      write(*,*)  'When pbp01 =',pbp01,'SHOCK BLOWN OUT THE NOZZLE
EXIT'
      write(12,*) 'When pbp01 =',pbp01,'SHOCK BLOWN OUT THE NOZZLE
EXIT'
     DXSHOCK=2.0D+00
        goto 499
      end if
      if (XSHOCK.lt.0.00001) then
      write(*,*)  'When pbp01 =',pbp01,' ENTIRE NOZZLE FLOW IS
SUBSONIC'
      write(12,*) 'When pbp01 =',pbp01,' ENTIRE NOZZLE FLOW IS
SUBSONIC'
     DXSHOCK=-2.0D+00
        goto 499
      end if

      goto 150

275   CONTINUE
C------ RATIO OF STATIC PRESSURES ACROSS THE SHOCK
      P2P1    = (1.0+K*EM1*EM1)/(1.0+K*EM2*EM2)
C------ RATIO OF ABSOLUTE TEMPERATURES ACROSS THE SHOCK
      T2T1    = (1.0+gm1h*EM1*EM1)/(1.0+gm1h*EM2*EM2)
C------ RATIO OF DENSITIES ACROSS THE SHOCK
      R2R1    = P2P1/T2T1
C------ ENTROPY JUMP ACROSS THE SHOCK (DIVIDED BY SPECIFIC
GAS CONSTANT)
     DELSR = (K/gm1)*LOG(T2T1)-LOG(P2P1)
      write(*,120)
     write(*,277)
iter,XSHOCK,PBP01,P2P1,R2R1,T2T1,EM1,EM2,EMEX,DELSR
     write(12,120)

write(12,277)iter,XSHOCK,PBP01,P2P1,R2R1,T2T1,EM1,EM2,EMEX,DELSR
     write(50,120)

write(50,277)iter,XSHOCK,PBP01,P2P1,R2R1,T2T1,EM1,EM2,EMEX,DELSR
277   format(1x,i4,1x,9(f8.3))
c------ find out variation of Mach number, temperature/T01, density/D01,
c------ pressure/P01 from shock location until nozzle exit and save them
      ishock = 1
      do 280 i=1,imax
       if(xx(i) .gt. xshock) then
     ishock=i
      go to 281
      endif
280   continue
281   continue
```

```
      write(42,*) imax,imax
      write(44,*) imax,imax
      write(46,*) imax,imax
      write(48,*) imax,imax
       emc = 0.9*em2
         do 400 i=ishock,imax
       AD = aa(ishock)
       AC = aa(i)
       EMD = EM2
       EMC = 0.9*EMc
C------ Second order Newton-Raphson iterative method to find local Mach
290    continue
       BLA = 1.0D+00 + gm1h*EMC*EMC
       BLAD= 1.0D+00 + gm1h*EMD*EMD
      G  =(BLAD/BLA)**EX
          F=AD*G*(EMD/EMC) - AC
       H= gp1h/BLA-1.0D+00/(EMC*EMC)
          FP=AD*G*EMD*H
      HH=(gp1h/(BLA*BLA))*EMC*((1.0D+00-K)+gp1h)
      HHH= -(gp1h/BLA)*(1.0D+00 + 1.0D+00/EMC) + 2.0D+00/(EMC**3)
        FPP=AD*G*EMD*(HH+HHH)
      FNC = F/FP
      FNCP= 1.0D+00 -(F*FPP)/(FP*FP)
      DMC = FNC/FNCP
       if ((Dabs(f).lt.0.00001).or.(Dabs(fp).lt.0.0000001)) goto 300
      FNCP  = 1.0D+00 - fnc*(fpp/(2.0d+00*fp))
      DMC   = FNC*FNCP
      EMCO = EMC
      EMC  = EMC - DMC
        goto 290
300    continue
      tt01  = 1.0000000/(1.00000000000000000 + gm1h*emc*emc)
      rr01  = tt01**(1.00000000000000/gm1)
      pp01  = rr01**k
       xMach(i)       = emc
       Temperature(i) = tt01
       Density(i)     = rr01
       Pressure(i)    = pp01
400   continue
      write(42,111) (xMach(i),i=1,imax)
      write(44,111) (temperature(i),i=1,imax)
      write(46,111) (density(i),i=1,imax)
      write(48,111) (pressure(i),i=1,imax)
c------
       write(12,*) 'Following values were computed at',imax,' locations'
      write(12,*) 'from inlet to exit of the entire nozzle for the'
      write(12,*) 'case of SHOCKED flow for pbp01 =',pbp01, 'and saved'
      write(12,*) 'in these files:'
      write(12,*) 'Q1D-Machshock.dat (local Mach number)'
```

```
    write(12,*) 'Q1D-TT01shock.dat (local Temp./inlet stagnat. temp.)'
    write(12,*) 'Q1D-RR01shock.dat (local Dens./inlet stagnat. dens.)'
    write(12,*) 'Q1D-PP01shock.dat (local Pres./inlet stagnat. pres.)'
c------
    EMC    = EM1
    XSHOCK = XSHOCK - DXSHOCK*STEP
499   continue
500   CONTINUE
      write(12,*)
      WRITE(12,*) 'SHOCK LOCATIONS AND JUMP CONDITIONS FOR
DIFFERENT'
    WRITE(12,*) 'EXIT PRESS/INLET STAG. PRESS SAVED ON Q1D-
SHOCKDATA'
c------ generate a 2D grid
    write(*,*)'Give number of grid cells from axis to wall (JCELLS)'
     write(*,*) '(important:  JCELLS < idm)'
    read(*,*)   JCELLS
    write(*,*) 'Give axis-to-wall clustering amplitude (0.0 < 0.9)'
    read(*,*)   AMPJ
    write(12,*)'Give number of grid cells from axis to wall (JCELLS)'
     write(12,*) '(important:  JCELLS < idm)'
    write(12,*)  JCELLS
    write(12,*) 'Give axis-to-wall clustering amplitude (0.0 < 0.9)'
    write(12,*)  AMPJ
    ampjpi = ampj/pi
     jmax = jcells + 1
    dr = 1.000000000000000/float(jmax-1)
       do 600 j=1,jmax
    rbar = (j-1)*dr
       do 550 i=1,imax
    r = rr(i)*(rbar + ampjpi*sin(pi*rbar))
    write(51,560) xx(i),r,i,j
550   continue
560   format(2f15.6,2i6)
600   continue
      write(*,*)  'x,r,i,j written on file: QID-x-r-grid.data'
      write(12,*) 'x,r,i,j written on file: QID-x-r-grid.data'
      stop
      end

      SUBROUTINE GAUSSJ(a,n,np,b,m,mp)
C**********************************************
C******* From Numerical Recipes in Fortran
C**********************************************
    IMPLICIT REAL*8 (A-H,O-Z)
              parameter (idm=201)
              real*8 a(np,np), b(np,mp)
              integer indxc(idm),indxr(idm),IPIV(idm)
C---------------------------------------------------------------------
```

```
   do 11 j=1,n
            ipiv(j)=0
11          enddo
   do 22 i=1,n
            big=0.0000000000
   do 13 j=1,n
            if(ipiv(j).ne.1)then
   do 12 k=1,n
            if(ipiv(k).eq.0)then
            if(Dabs(a(j,k)).ge.big)then
            big=Dabs(a(j,k))
            irow=j
            icol = k
            endif
            endif
12    enddo
            endif
13    enddo
            ipiv(icol)=ipiv(icol)+1
            if(irow.ne.icol) then
            do 14 l=1,n
            dum = a(irow,l)
            a(irow,l)=a(icol,l)
            a(icol,l)=dum
14    enddo
     do 15 l=1,m
            dum = b(irow,l)
            b(irow,l)=b(icol,l)
            b(icol,l)=dum
15    enddo
            endif
            indxr(i)=irow
            indxc(i)=icol
            if(a(icol,icol).eq.0.000) pause 'singular matrix'
            pivinv = 1.0D+00/a(icol,icol)
            a(icol,icol)=1.0D+00
            do 16 l=1,n
            a(icol,l)=a(icol,l)*pivinv
16    enddo
     do 17 l=1,m
            b(icol,l)=b(icol,l)*pivinv
17    enddo
            do 21 ll=1,n
            if(ll.ne.icol) then
            dum=a(ll,icol)
            a(ll,icol)=0.0D+00
            do 18 l=1,n
            a(ll,l) = a(ll,l)-a(icol,l)*dum
18    enddo
```

```
      do 19 l=1,m
            b(ll,l) = b(ll,l)-b(icol,l)*dum
19    enddo
            endif
21    enddo
22    enddo
            do 24 l=n,1,-1
            if(indxr(l).ne.indxc(l))then
            do 23 k=1,n
            dum = a(k,indxr(l))
            a(k,indxr(l)) = a(k,indxc(l))
            a(k,indxc(l)) = dum
23    enddo
            endif
24    enddo
            return
            end
```