Washington University in St. Louis
Washington University Open Scholarship

Engineering and Applied Science Theses & Dissertations

McKelvey School of Engineering

Summer 8-2015

CFD SIMULATION AND SHAPE OPTIMIZATION OF SUPERSONIC EJECTORS FOR REFRIGERATION AND DESALINATION APPLICATIONS

Liju Su
Washington University in St. Louis

Follow this and additional works at: https://openscholarship.wustl.edu/eng_etds

 Part of the Engineering Commons

Recommended Citation
Su, Liju, "CFD SIMULATION AND SHAPE OPTIMIZATION OF SUPERSONIC EJECTORS FOR REFRIGERATION AND DESALINATION APPLICATIONS" (2015). *Engineering and Applied Science Theses & Dissertations*. 107.
https://openscholarship.wustl.edu/eng_etds/107

This Thesis is brought to you for free and open access by the McKelvey School of Engineering at Washington University Open Scholarship. It has been accepted for inclusion in Engineering and Applied Science Theses & Dissertations by an authorized administrator of Washington University Open Scholarship. For more information, please contact digital@wumail.wustl.edu.

WASHINGTON UNIVERSITY IN ST. LOUIS
School of Engineering and Applied Science
Department of Mechanical Engineering and Material Science

Thesis Examination Committee:
Ramesh Agarwal, Chair
Kenneth Jerina
David Peters

CFD SIMULATION AND SHAPE OPTIMIZATION OF SUPERSONIC EJECTORS FOR REFRIGERATION AND DESALINATION APPLICATIONS

by

Liju Su

A thesis presented to the School of Engineering and Applied Science
of Washington University in St. Louis
in partial fulfillment of the requirements for the degree of
Master of Science

August 2015

Saint Louis, Missouri

Contents

List of Figures

List of Tables

Acknowledgments

First of all, I would like to acknowledge my advisor Dr. Ramesh Agarwal for encouraging me to devote myself in this CFD research. I have learnt a lot from him about CFD and doing research. His knowledge of engineering fundamentals and computational fluid dynamics has helped me enormously in completing my research with a great understanding.

I would also like to acknowledge many who helped me with the thesis both academically and emotionally. Thanks to Guangyu Bao for his guidance in CFD modeling and FLUENT simulations. Thanks to Tim Wray for helping me understand the intricacies of CFD. Thanks to Chris Seager for helping me with the GA methodology. Thanks to Subhodeep Banerjee for helping me in understanding the compressible aspects of the flow. Thanks to all my CFD lab colleagues for their continuous help and encouragement.

I would like to acknowledge my committee members, Dr. Peters and Dr. Jerina, for taking the time to read the thesis and attend its defense.

Liju Su

Washington University in St. Louis

August 2015

Dedicated to my parents

I would like to dedicate this thesis to my parents (Yunlong Su and Xingjuan Wang) for their lifetime support, understanding, and encouragement

Love you forever

ABSTRACT

CFD SIMULATION AND SHAPE OPTIMIZATION OF SUPERSONIC EJECTORS FOR REFRIGERATION AND DESALINATION APPLICATIONS

by

Liju Su

Master of Science in Mechanical Engineering

Washington University in St. Louis

Research Advisor: Professor Ramesh K. Agarwal

The aim of this thesis is to investigate the detailed flow field inside the supersonic ejector using numerical methods and to optimize the ejector's mixing chamber wall shape to obtain a maximum entrainment ratio (ER) in order to obtain the highest possible efficiency that can be attained by the ejector. A steam ejector applied in the cooling industry is first studied to determine the most accurate turbulence model for its supersonic jet flow field simulation with mixing with the entrained steam in the mixing chamber. A commercial Computational Fluid Dynamics (CFD) package FLUENT 14.5 along with the meshing tool ICEM 14.5 is utilized to conduct the modeling and simulation to examine the ejector performance using two different turbulence models: k- ε realizable and k- ω SST. Velocity contours, pressure plots and entrainment ratio plots obtained from FLUENT are studied to investigate the effects of several ejector operating conditions as well as to verify the turbulence model accuracy by comparing the numerical results with experimental data. Simulations for three different supersonic ejectors (ejectors for refrigeration and desalination application with different working fluids namely the steam or compressed air) are conducted to further validate the numerical solution accuracy. The turbulence model producing more accurate results is applied to all three cases. In second part of the thesis, a single objective genetic algorithm

(SOGA) is employed to optimize the mixing chamber wall shape for steam ejector for refrigeration to achieve the maximum entrainment ratio. Bezier Curves are used to generate the new wall shapes. The whole shape generation-meshing-simulation-SOGA process is repeated until the ER converges to a maximum value based on the specified convergence criteria for SOGA.

Chapter 1 Introduction

1.1 Motivation

Supersonic ejectors are widely used as a compressor in a wide range of industrial applications such as refrigeration, desalination, etc. The flow filed analysis within the ejector is relatively difficult due to the supersonic flow of the jet and its mixing with the ambient fluid which requires taking into account the compressibility effects at high Mach numbers. The experimental setups are quite expensive and can analyze only a limited number of flow conditions. Therefore, the computational fluid dynamics (CFD) technology has been commonly applied to the supersonic ejector simulations for simulation of the flow field for a range of geometries and flow conditions as well as for better visualization of the flow field. The relevant flow variables are determined, the most important among them is the entrainment ratio (ER), which plays a key role in ejector design. In this thesis, we employ the CFD technology to simulate the flow in a steam ejector used in refrigeration applications by solving the Reynolds-Averaged Navier-Stokes (RANS) in conjunction with a number of different turbulence models. The computational results are compared with available experimental data to assess the accuracy of the numerical results obtained with different turbulence models. Employing the most accurate turbulence model, additional simulations are conducted for the other 3 cases. Finally, shape optimization is applied to one of the supersonic steam ejectors using a genetic

algorithm (GA) to achieve the maximum entrainment ratio (ER), which corresponds to the highest possible efficiency for the ejector.

1.2 Background

Supersonic ejectors are widely used in a variety of industrial applications such as aerospace (in V/STOL aircraft), and for desalination and refrigeration. In this thesis, a steam ejector and an air ejector are studied for both refrigeration and desalination applications with a total of four different supersonic ejectors.

1.2.1 Ejectors for Refrigeration

In cooling industry, the systems used to remove heat from a lower temperature reservoir are driven either mechanically or thermally. Mechanical compressor still plays an important role in majority of the installations for both commercial and residential applications, but for thermally driven systems like ejector systems, much attention has been paid in the last few decades due to their environmentally friendly operating character [1]. The ejector systems can be activated by low temperature renewable energy source like solar energy. Another advantage is that water as the most environmentally friendly substance can be used in these systems. It has been recognized that the performance of the system depends largely on the performance of the ejector [2]. Hence, to improve the performance of an ejector, comprehensive understanding of the flow inside the ejector is needed. The early attempts on application of computational fluid dynamics (CFD) techniques for

ejector simulation were made in 1990s [3]. Since then, the CFD has been commonly used to investigate the complex local flow physics inside the ejector and has now become an important part for designing and optimizing the ejector performance.

A schematic view of the refrigeration cycle using an ejector is shown in Fig. 1.1. The boiler creates high- pressure saturated water vapor as the primary fluid. Then the high-pressure steam gets accelerated in the nozzle within the ejector and draws the secondary water vapor from the evaporator. The low pressure in the evaporator causes the water to evaporate, which gives the refrigeration effect.

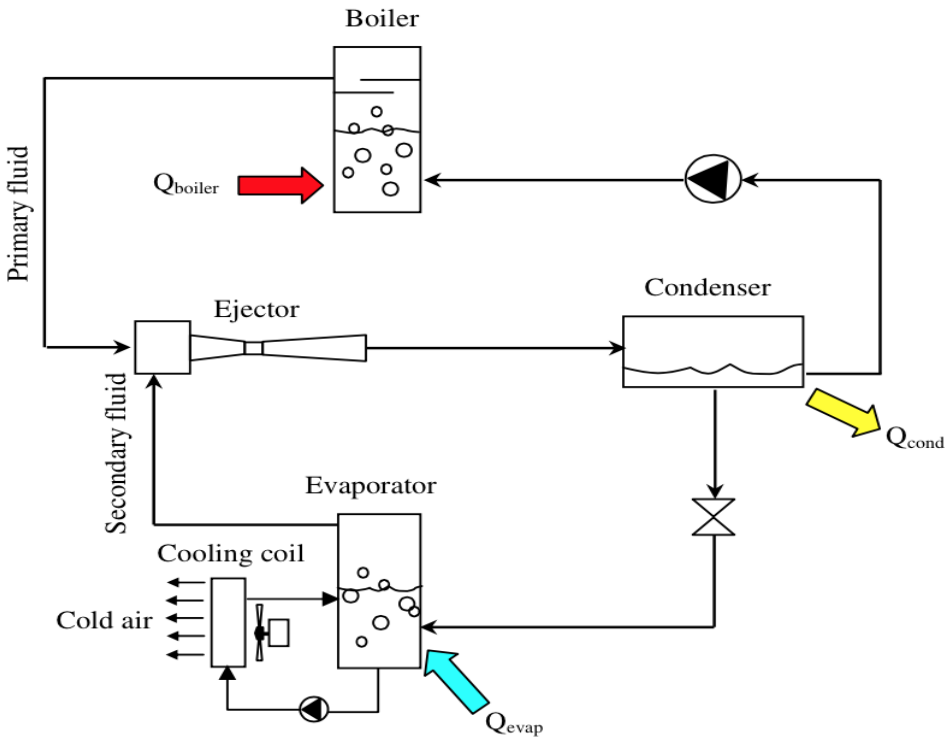

Figure 1.1: Schematic view of refrigeration cycle using an ejector

Performance of a steam ejector in refrigeration cycle is defined in terms of coefficient of performance (COP). Since the enthalpy change at the boiler is almost the same as that at the evaporator, we can assume COP ≈ ER [4], where the definition of ER is as follows:

$$\mathrm{ER} = \frac{\textbf{\textit{mass flow rate of the entrianed secondary fluid}}}{\textbf{\textit{mass flow rate of the primary fluid}}} \qquad (1)$$

A larger value of ER corresponds to a better performance of the ejector. ER is therefore used in both the verification of CFD results against the experimental data and in optimization of ejector performance.

An ejector is a simple fluid pumping device with four major parts: primary nozzle, mixing chamber, constant-area throat, and subsonic diffuser as shown in Fig. 1.2. The primary fluid gets accelerated to supersonic flow level through the converging-diverging nozzle, and outside the nozzle exit there is a low-pressure region known as supersonic jet core. Due to the considerable pressure difference between this region and the secondary fluid chamber, the secondary fluid gets sucked into the mixing chamber, where the mixing process begins. When the velocity of the secondary fluid reaches the sonic level, also known as the choked flow-condition, string mixing between the primary and secondary fluid occurs. The annular area formed at that specific position is called an effective area. The mixing continues until the velocity of both primary and secondary fluids drops to subsonic level and finally the mixture gets discharged into the condenser. Several supersonic flow phenomena like shock and expansion waves occur inside the ejector, which can be clearly visualized in the CFD simulations.

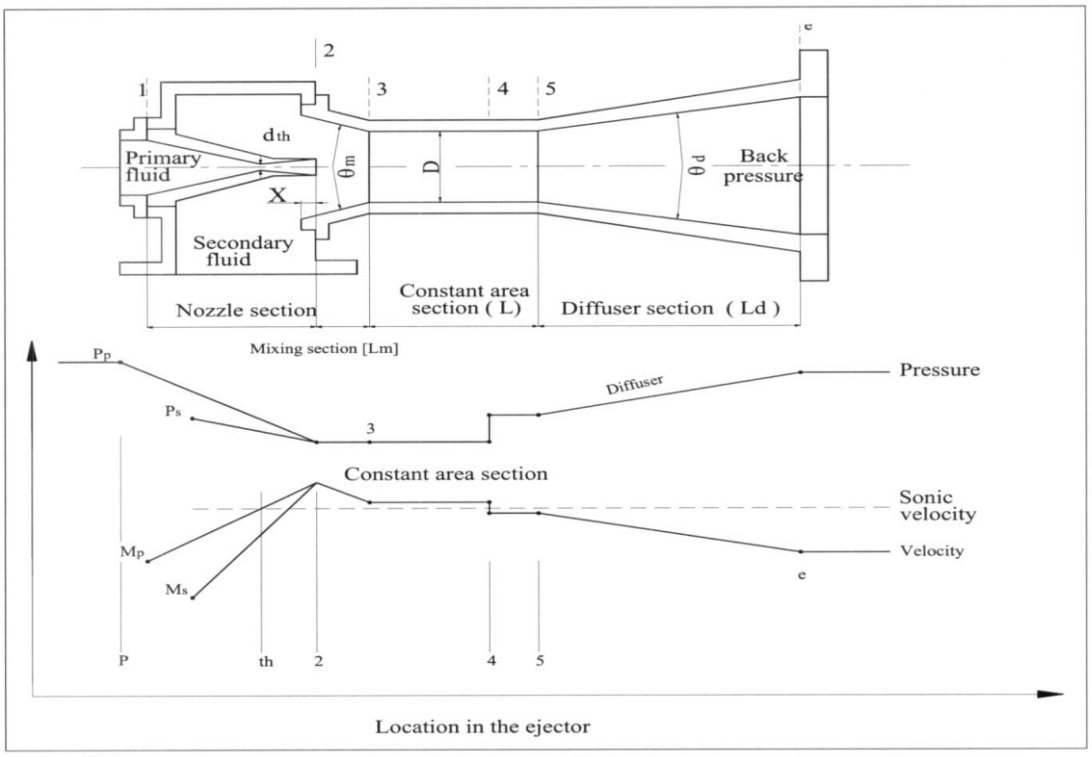

Figure 1.2: 2-D schematic of a typical steam ejector with pressure and velocity distributions along the center line [5]

1.2.2 Ejectors for Desalination

In desalination industry, multi-stage flash distillation (MSF) and reverse osmosis (RO) are the most well-known technologies for potable water production. They require large amount of energy consumption, therefore research on how to increase the energy efficiency by utilizing waste heat (e.g. the heat from diesel generator) has been conducted over several decades [9]. Figure 1.3 shows the schematic of a multi-effect evaporation with thermal vapor compression (MEE-TVC) system.

This system is widely employed in thermal desalination industry due to its lower corrosion and scaling rates, lower capital cost, longer operation life, and requirement of less pumping power [10].

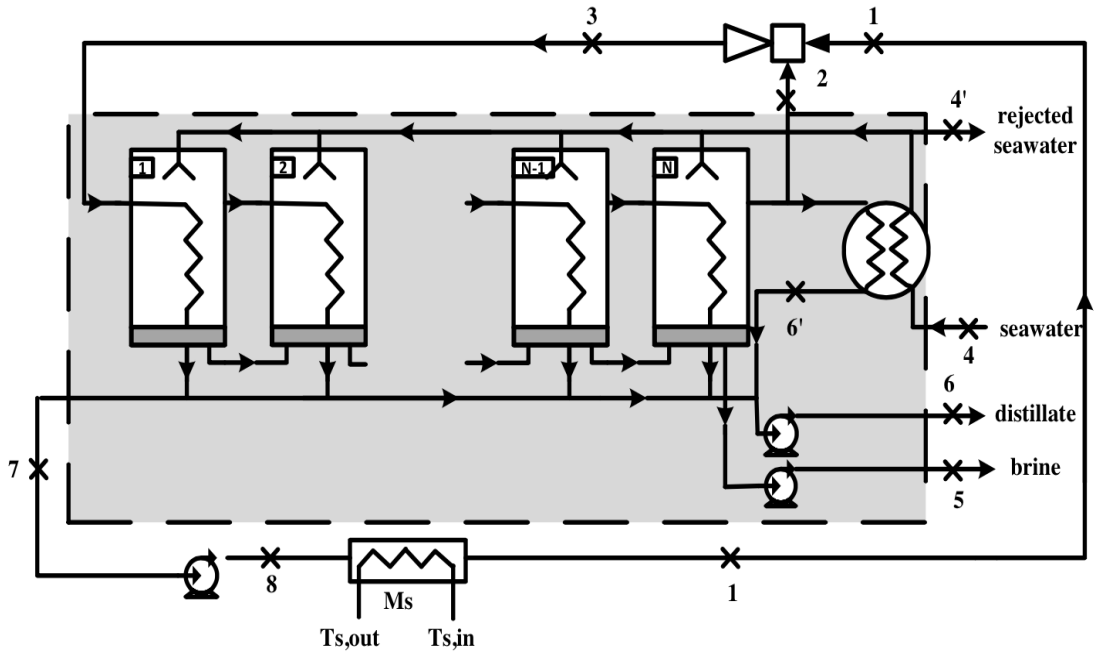

Figure 1.3: Schematic of a multi-effect evaporation with thermal vapor compressor (MEE-TVC) system [9]

In Fig. 1.3, State 1 is the high pressure saturated water vapor produced by the waste heat source. It works as the primary fluid for the thermal compressor. The secondary stream is the low pressure saturated water vapor coming from the N^{th} effect container. The compressed water vapor at Sftate 3 goes into the first effect container and condenses due to the seawater sprayed outside the tube. The seawater gets evaporated and the steam goes into the next stage while the brine drops onto the

bottom of the first container. The whole process occurs in several stages with distillate and brine being pumped out separately. An important parameter used to describe the performance of an ejector in a desalination system as thermal vapor compressor (TVC) is the entrainment ratio [8], which has the same definition given by equation (1) for an ejector in refrigeration application. Figure 1.4 shows the 2-D schematic of a typical air ejector with the velocity and static pressure distributions along the centerline. The major difference between a steam ejector and an air ejector is in their dimensions. The geometry parameters of the four ejectors investigated will be described in the following chapter.

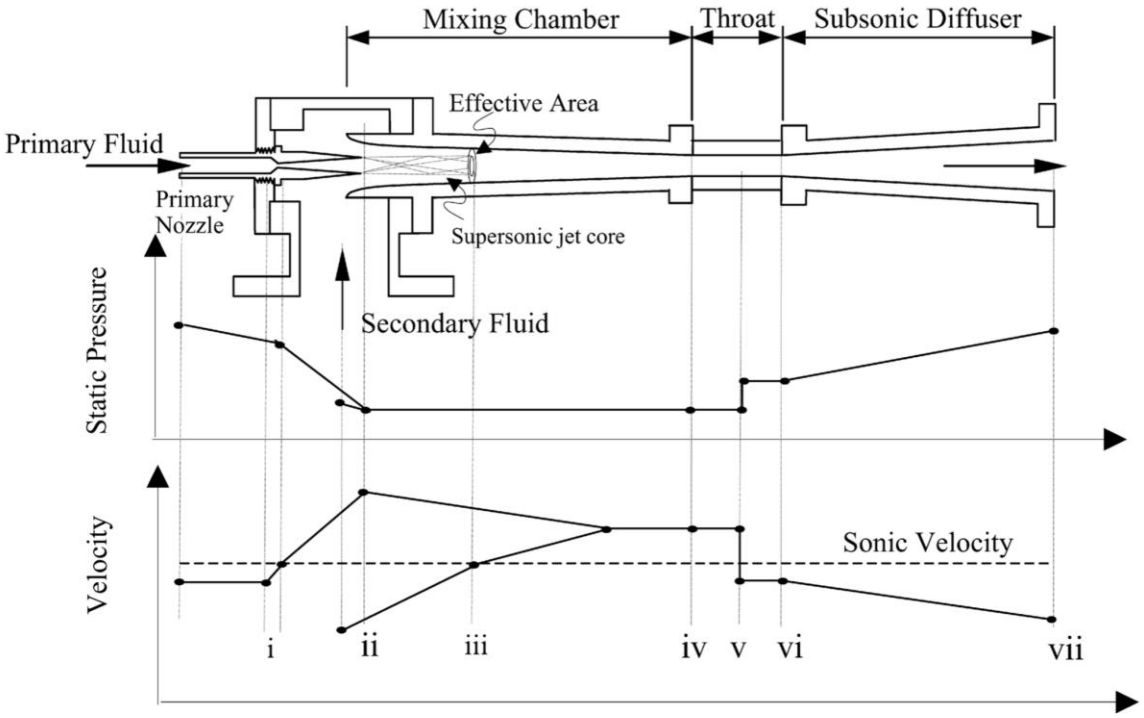

Figure 1.4: 2-D schematic of a typical air ejector used in desalination system with velocity and static pressure distributions along the centerline [11]

1.3 Goals and Objectives

The goal of this thesis is to apply the CFD technology for flow field simulation in four supersonic ejectors used in refrigeration and desalination application using steam and air as a working fluid. To get accurate numerical results, a good mesh and an accurate solver for the solution of Reynolds-Averaged Navier-Stokes (RANS) equations in conjunction with a turbulence model are needed. Several turbulence models are evaluated to determine their accuracy by comparing the computed results with experimental data. After validation of the CFD results against the experimental data, a genetic algorithm is applied to optimize the mixing chamber wall shape of one of the supersonic steam ejectors to get a maximum entrainment ratio under specific operating conditions, since increase in entrainment ratio improves the efficiency and performance of the ejector.

Chapter 2 CFD Simulations

CFD simulations are performed using the commercial software ANSYS FLUENT 14.5. Ejector geometry and mesh inside it are generated using the software ANSYS-ICEM. A solution-adaptive mesh is employed to obtain the mesh independent solutions. The two different two-equation turbulence models are tested-the k-ε realizable and k-ω SST to assess their accuracy in simulations. These two turbulence models are believed to be reasonably accurate in predicting the free shear layer/mixing flows such as the flow field of a round jet compared to other models [5][6][7]. First, T. Sriveerakul's steam ejector geometry [5] is used for simulation of refrigeration application. Wall static pressure, static pressure along the ejector centerline and the entrainment ratio are calculated using ANSYS-FLUENT and are compared with the experimental data to determine which turbulence model gives more accurate results for the ejector employed in refrigeration application. The relatively more accurate turbulence model is then employed in simulation of three additional used in refrigeration and desalination for further validation/validation; the computed results are compared either with experimental data or with some previous numerical investigations reported in the literature.

2.1 Mesh Generation

The first ejector geometry used in this study is based on the experiment configuration of Sriveerakul et al. [5]. The mesh generation software ANSYS-ICEM is used to generate the grid inside the

computational domain as shown in Fig. 2.1. A denser mesh is created in the mixing layer region and in the primary nozzle stream region to capture both the shocks and the mixing layer flow field accurately.

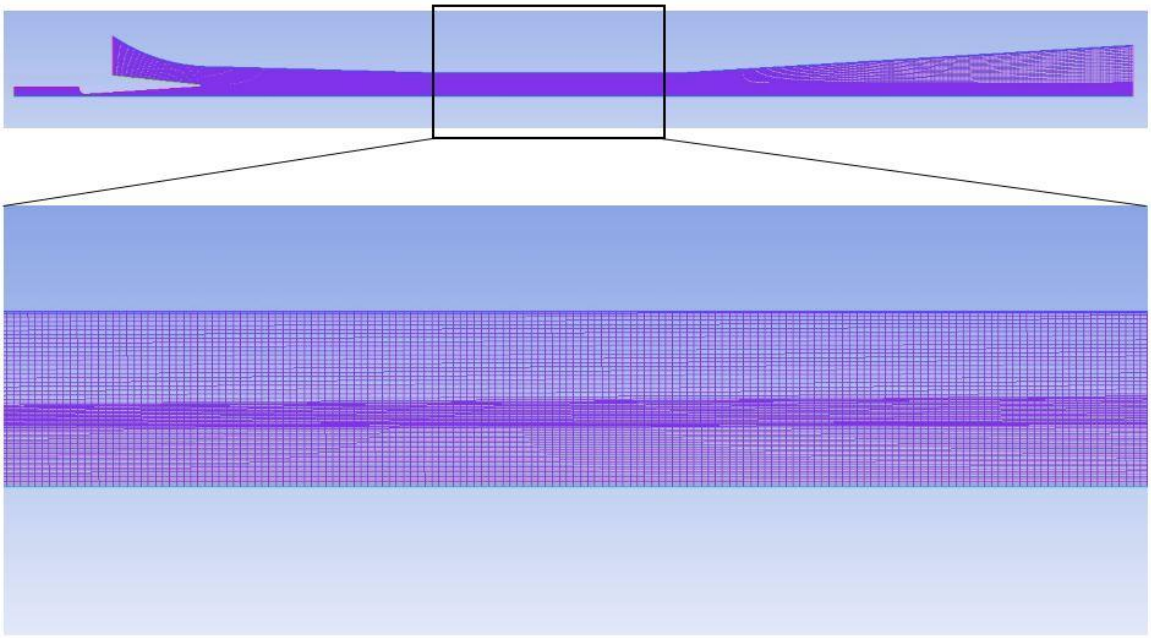

Figure 2.1: Mesh inside the steam ejector (upper) with zoomed - in view of the mesh in the throat region (lower)

The ejector geometry is created in CREO Parametric and is then imported into ICEM for mesh generation. The geometric dimensions of the ejector are listed in Table 2.1. 75 nodes are used on the vertical sides of the grid blocks while 800 nodes are used in the horizontal direction, resulting in 60000 cells in the entire 2D computational domain. Bigeometric mesh law is applied to all the meshing parameters. The spacing 2 is set to be 0.01 near the ejector wall to ensure that the desirable

y^+ value is obtained. It should be noted that additional mesh refinement may be required near the ejector wall when using the k-ω SST model since the near wall treatment of this model requires the y^+ values to be at least less than 5, whereas for the k-ε realizable model y^+ below 5 is considered acceptable [8]. To employ the mesh refinement procedure, we use the Yplus/Ystar Adaption tool under Adapt menu and type in a required threshold value of y^+.

Table 2.1: Ejector geometry specifications [5]

Specifications	Values
Primary nozzle inlet diameter	7.75 mm
Primary nozzle throat diameter	2.00 mm
Primary nozzle outlet diameter	8.00 mm
Nozzle exit position (NXP)	35.00 mm
Mixing chamber inlet diameter	24.00 mm
Mixing chamber length	130.00 mm
Throat diameter	19.00 mm
Throat length	95.00 mm
Subsonic diffuser length	180.00 mm

2.1.1 Mesh Adaption

FLUENT has the ability to adapt the mesh after a converged solution on a standard mesh is obtained. This mesh adaption feature is based on the calculated density gradient, pressure gradient or velocity gradient. It increases the mesh density in high gradient regions, which enables more accurate results in these high flow gradient regions. After the first converged solution, is obtained, 5% of the maximum density in the domain is set for the grid refinement. This adaption process is repeated two times to obtain the final mesh as shown in Fig. 2.2.

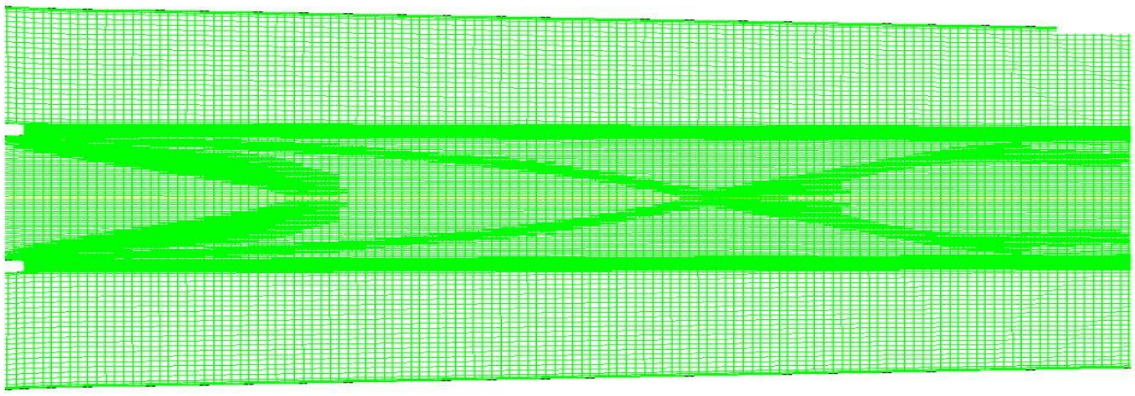

Figure 2.2: Adaptive mesh using the density gradient as a sensor

2.1.2 Mesh Independence Study for the Computed Solution

The increase in cell numbers usually leads to more accurate solution but also increases the computational time. However, beyond a certain mesh density there is no appreciable change in the

computed solution. To establish the mesh independence of the solution, two meshes are created for each turbulence model-a coarse mesh containing nearly 60,000 cells and a very fine mesh containing nearly 240,000 cells. Wall static pressure, centerline velocity and the entrainment ratio are calculated on these two meshes and are compared to determine whether the coarse mesh with 60000 cells has sufficient accuracy. It can be seen from Figs. 2.3-2.8 that the coarse mesh (the original mesh) provides sufficient resolution for obtaining accurate solutions which can be considered as mesh independent.

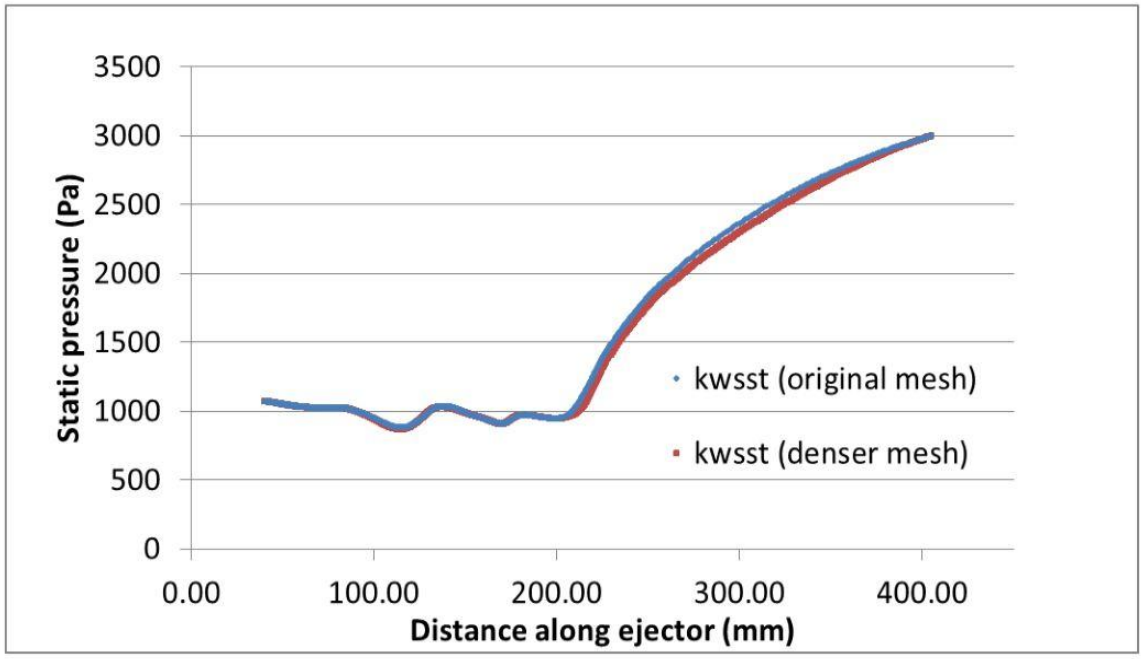

Figure 2.3: Comparison of results using coarse and fine mesh for wall static pressure (k-ω SST model)

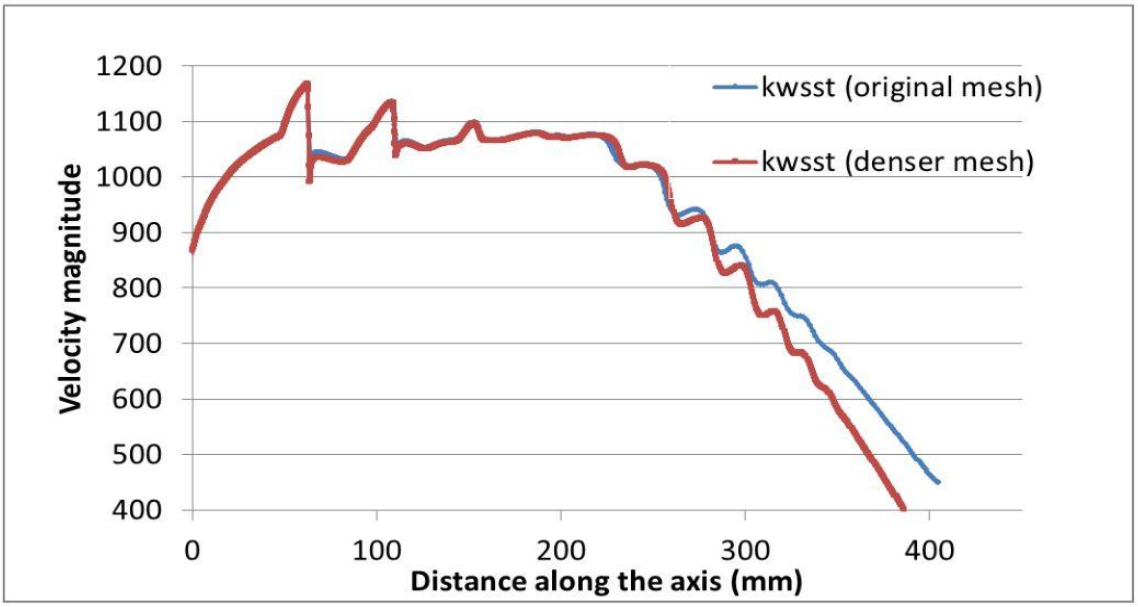

Figure 2.4: Comparison of results using coarse and fine mesh for centerline velocity (k-ω SST model)

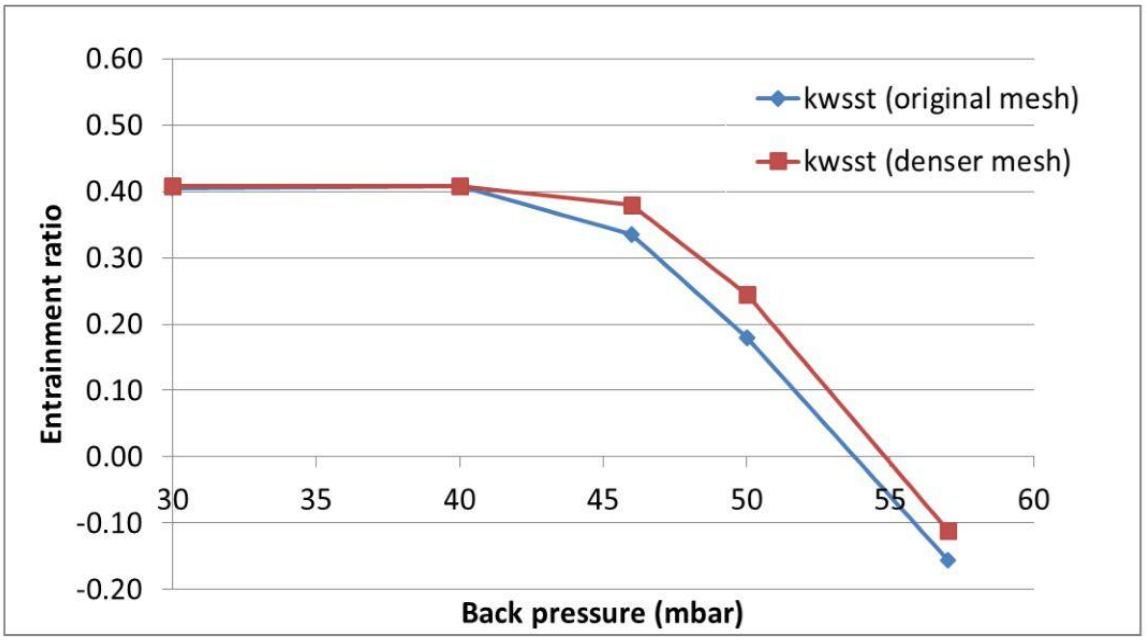

Figure 2.5: Comparison of results using coarse and fine mesh for entrainment ratio (k-ω SST model)

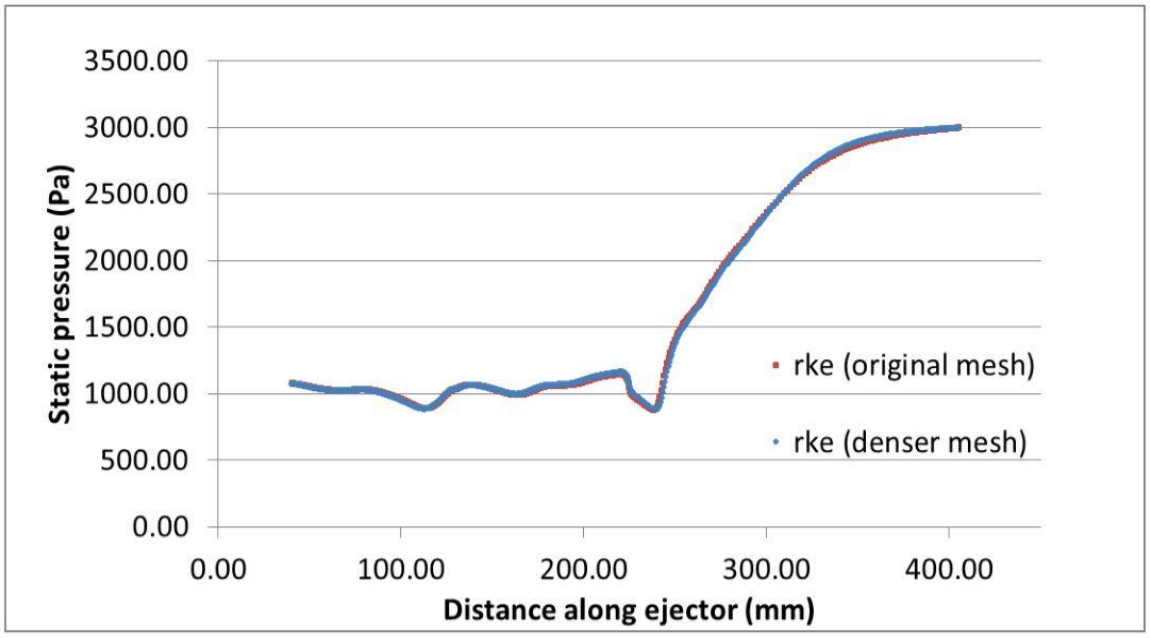

Figure 2.6: Comparison of results using coarse and fine mesh for wall static pressure using k-ε realizable model (rke)

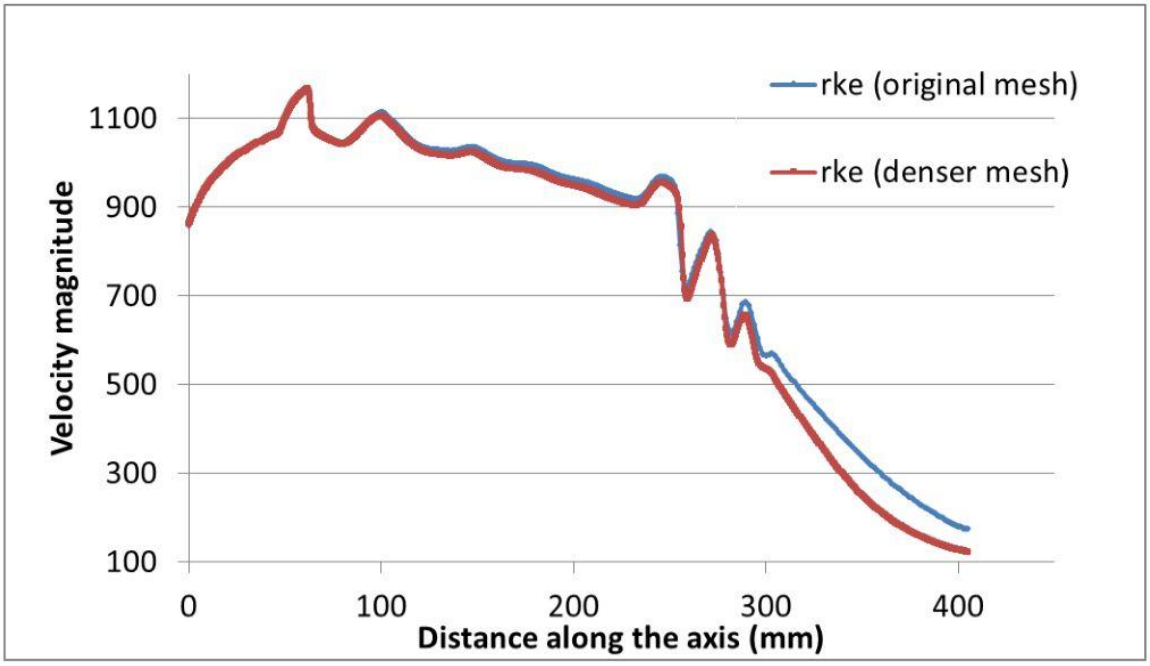

Figure 2.7: Comparison of results using coarse and fine mesh for centerline velocity using k-ε realizable model (rke)

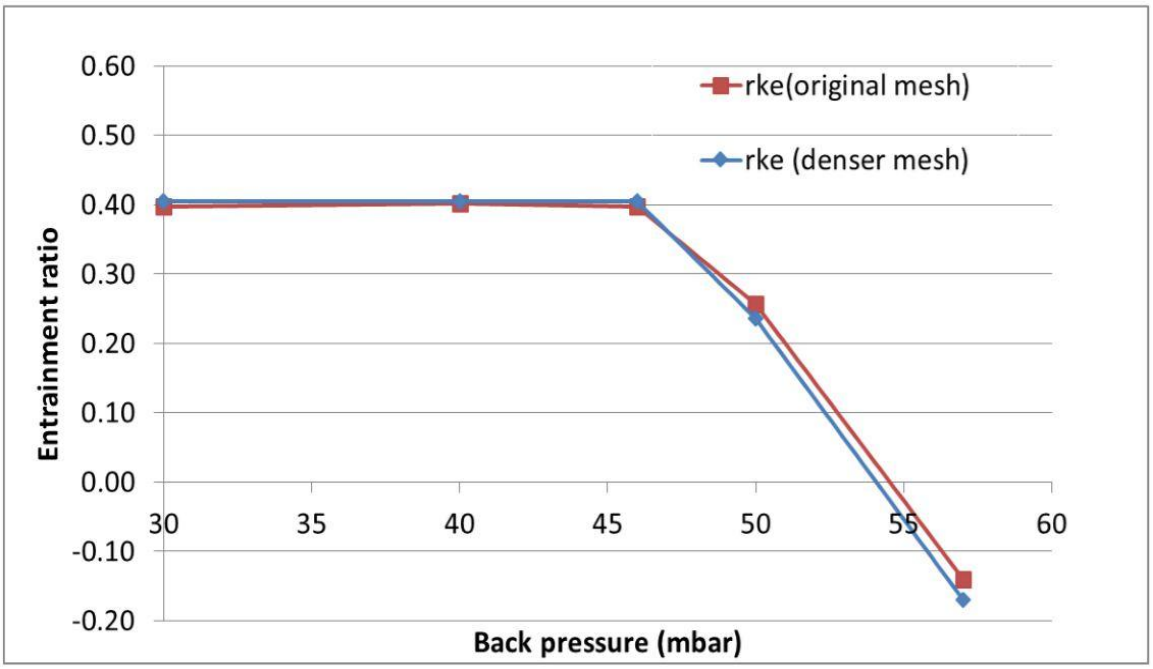

Figure 2.8: Comparison of results using coarse and fine mesh for entrainment ratio using k-ε realizable model (rke)

2.2 Flow Field Simulation Using FLUENT

2.2.1 Brief Outline of the Numerical Modeling

The turbulent flow field is modeled using the Unsteady Reynolds-Averaged Navier-Stokes (URANS) equations in conjunction with two widely used turbulence models, the k-ω SST model and the k-ε realizable model. The second-order upwind scheme is applied to numerically solve the flow equations. The density-based solver in FLUENT is chosen to account for the compressibility effects since the flow in the ejector is both supersonic and subsonic. Steady-state, axisymmetric solver is

employed. The working fluid is water-vapor which is imported from Fluent Database. It is treated as an ideal gas since for the ejector application, the operating pressure is relatively low [5]. The properties of the working fluid are listed in Table 2.2.

Table 2.2: Water-vapor properties

Properties	Values
Density	Ideal-gas Model
Cp (Specific heat)	2014 J/kg-K
Thermal conductivity	0.0261 W/m-k
Viscosity	1.34e-5 kg/m-s
Molecular weight	18.01534 kg/kgmol

2.2.2 Boundary Conditions

Boundary conditions for the two inlets (primary fluid inlet and secondary fluid inlet) are set to be pressure-inlet types, while the one at the outlet of the ejector is applied as the pressure-outlet type. The primary fluid inlet pressure and the secondary fluid inlet pressure are set at 270,280 Pa and 1230 Pa respectively with corresponding saturation temperatures of 403.15 K and 283.15 K. The operating conditions are varied by changing the discharge pressure (back pressure). Five cases of the

backpressure are considered using each turbulence model – 30 mbar, 40 mbar, 46 mbar, 50 mbar, and 57 mbar.

2.2.3 Convergence Monitors and Criteria

Three convergence monitors are employed to ensure that the final results are convergent to the desired tolerance. The residuals of all six governing flow equations including the turbulence model equations are monitored until they achieve the absolute value of 10^{-6}; at that stage the solution is considered converged. A drag coefficient monitor is employed to monitor the convergence of C_d at the ejector wall; C_d should achieve a constant value when the solution is converged. The third convergence monitor employed is that the net mass flow rate from the inlets must equal the net mass flow rate at the outlet within a tolerance of 10^{-6} kg/s. It turns out that the scaled residual values may not reach 10^{-6} for all six flow governing equations, instead, they may oscillate around 10^{-6} ~10^{-4}. Under this circumstance, we still consider the solution to be converged as long as the second and third criteria mentioned above are satisfied.

2.3 Results and Discussions

2.3.1 Selection of a Turbulence Model

The wall static pressure, static pressure along the centerline, and the entrainment ratio are computed using both the turbulence models-the k-ε realizable model and k-ω SST model. Comparisons of computations using these two models and the experimental data are made to determine which turbulence model gives more accurate numerical results. Figure 2.9 shows the wall static pressure plot; a relatively more accurate ejector wall pressure distribution is obtained using the k-ε realizable model. From Fig. 2.10, one can see that k-ε realizable model has better shock prediction capability along the ejector, especially at the end of the constant-area mixing chamber region, where the static pressure oscillates a great deal along the centerline due to shock formation. Entrainment ratio (ER) is compared in Fig. 2.11. As one can see, the CFD simulation results are not very accurate compared to the experimental data. At ER=0, there is nearly 10% error since the experimental value of the back pressure is 49 mbar, while the numerical results give a value of~54 mbar. Considering the difference between the experimental and computational ER plots, it can be observed that the critical point where the flow changes from choked to un-choked condition is more clearly delineated using the k-ε realizable model. Velocity contours for all five cases with different back pressures are shown in Fig. 2.12. The contours for computed flow inside the ejector obtained using the k-ε realizable model are in excellent agreement with the computed results from Ref. [5]. Based on these observations, it can be concluded that the k-ε realizable turbulence model is relatively a more

accurate model for computation of the steam ejector flow fields. It is therefore recommended that this model be used in simulation of other ejectors and their shape optimization to increase the entrainment ratio. Thru what follows in rest of this thesis, k-ε realizable model is employed in the computations.

As mentioned above, the effect of back pressure on the entrainment ratio (ER) is investigated by applying five different back pressure values. From Fig. 2.11, it is clear that the ER stays constant in certain range of back pressure, then drops rapidly as the back pressure is increased. This can be explained by examining the velocity contours; for the first three operating back pressures (A, B and C), flow structures stay the same before the secondary flow reaches the sonic level (choked condition), but as the back pressure continues to increase, the second shock moves further upstream and interferes with the mixing process causing the secondary flow to be no longer choked. Finally if the back pressure is too high, reverse flow occurs, which means that the fluid is forced into the secondary fluid chamber resulting in a negative ER value. For choked secondary flow, a constant entrainment ratio is always obtained even if the back pressure changes [4].

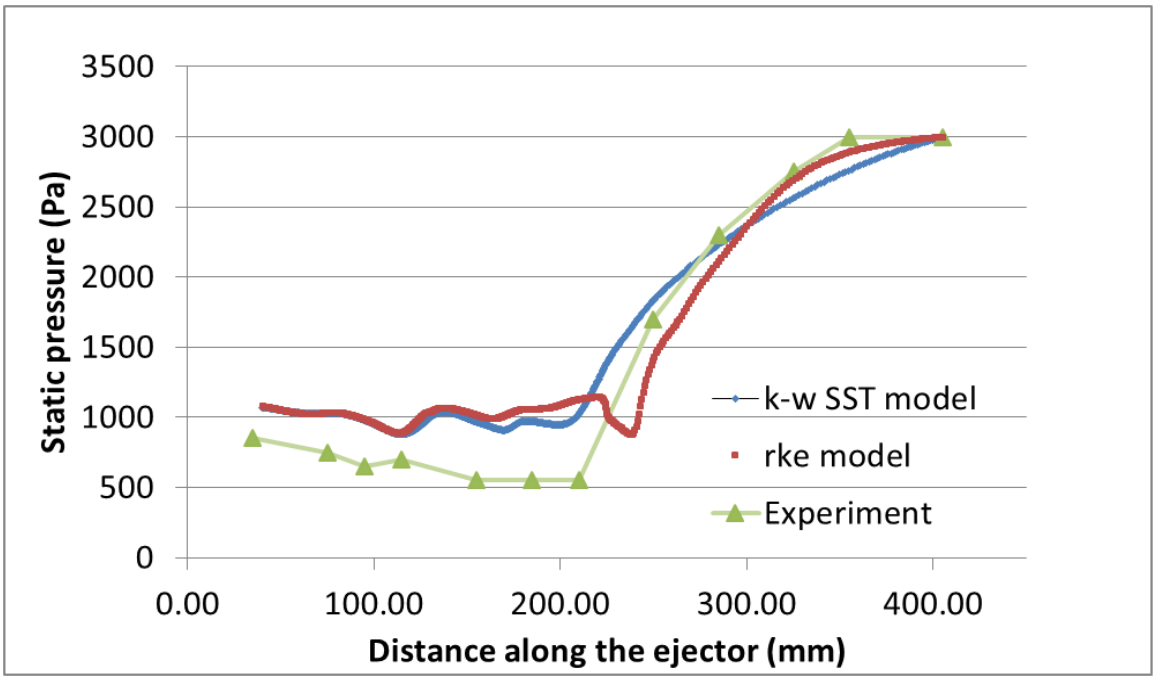

Figure 2.9: Comparison of computed static pressure along the ejector wall using the k-ω SST (kwsst) and k-ε realizable (rke) turbulence models with the experimental data [2]

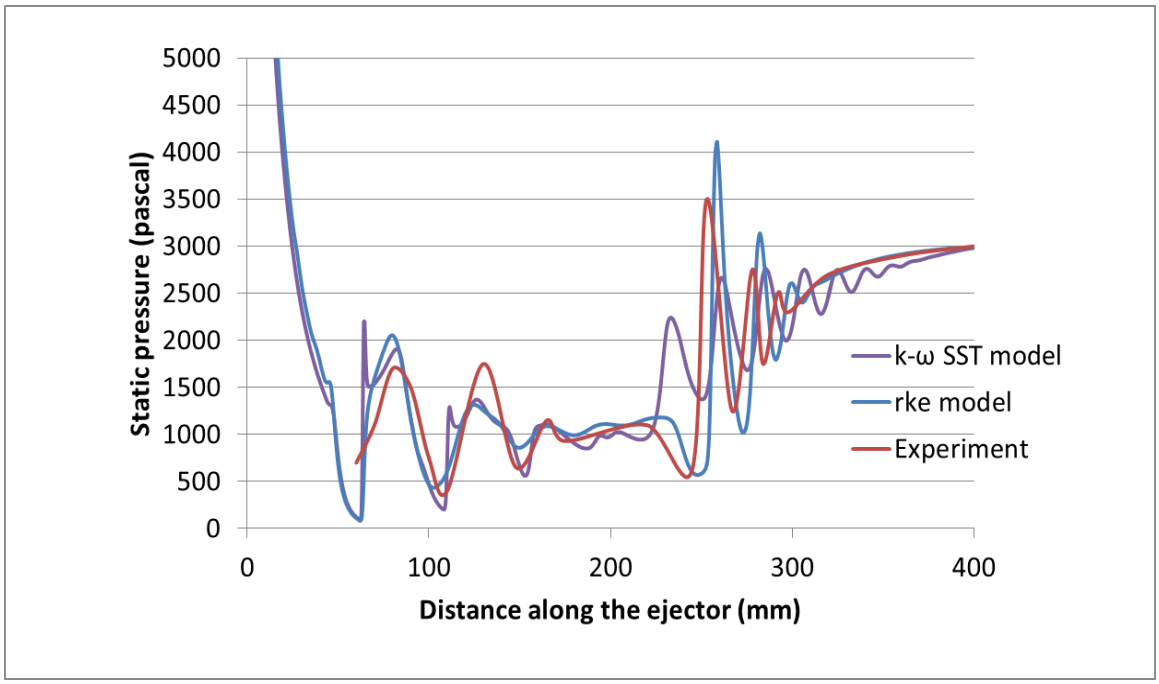

Figure 2.10: Comparison of computed static pressure along the ejector centerline using the k-ω SST and k-ε realizable (rke) turbulence models with the experimental data [2]

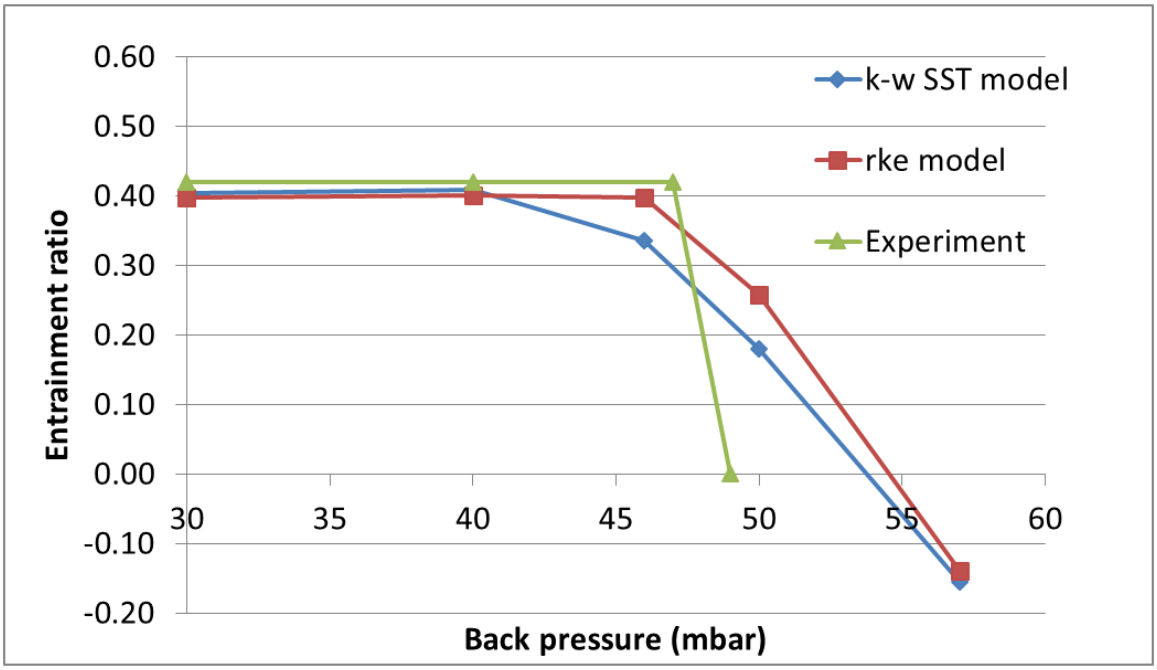

Figure 2.11: Comparison of computed entrainment ratio using the k-ω SST and k-ε realizable (rke) turbulence models with the experimental data [2]

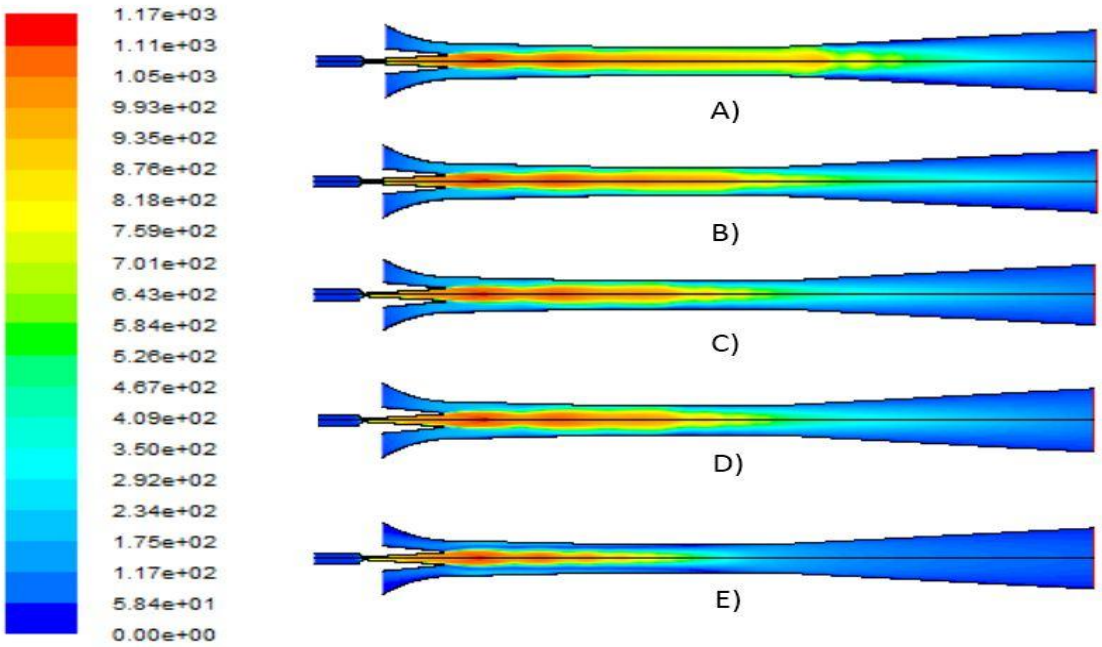

Figure 2.12: Velocity contours inside the steam ejector using the k- ε realizable turbulence model for back pressure of A) 30 mbar, B) 40 mbar, C) 46 mbar, D) 50 mbar, and E) 57 mbar showing movement of shock upstream as the outlet pressure is increased

2.3.2 General Comments on the Flow Field of an Engine Exhaust

The flow field within the ejector has a complex flow pattern. In case of a V/STOL aircraft, it usually occurs during the takeoff. Figure 2.13 shows the evenly spaced rings in the exhaust of an engine which are known as shock diamonds (Mach disks). Depending on the ambient pressure, the flow pattern in the engine exhaust can be categorized into two different types.

The first one is called overexpansion as shown in Fig. 2.14. In this case, the ambient pressure is higher than the flow pressure at the nozzle exit, which causes the flow to be compressed inward. This compression effect occurs due to oblique shock waves generated at the nozzle exit. As the flow turns and the flow becomes parallel to the centerline, a normal shock occurs. The oblique shocks emanating from the nozzle exit and the normal shocks form a Mach disk which is perpendicular to the centerline. Passing through the normal shock causes the temperature of the flow to increase, causing any excess fuel present in the engine exhaust to burn. It is this burning of the fuel that makes the Mach disk glow and become visible to create the ring pattern in the engine exhaust [12]. As the flow keeps moving downstream, it gets so compressed that the pressure of the flow exceeds the ambient pressure and the flow then expands through a series of expansion waves (expansion fans), which cause the pressure to drop below the ambient pressure. After the expansion waves reache the free boundary, they reflect back and form a compression fan. If the compression waves are strong enough, they merge into an oblique shock wave and form a new Mach disk similar to the

one near the nozzle exit [12]. The whole compression and expansion process repeats itself to form the evenly spaced rings as we mentioned above.

The second case is called the underexpansion in which the exit flow pressure is higher than the ambient pressure. The flow keeps expanding after it leaves the nozzle, forming an expansion fan instead of oblique shock waves as in the first case described above. The expansion waves then reflect at the free boundary, back into the jet to form a compression fan (oblique shock waves if strong enough). The same compression-expansion pattern occurs downstream with a series of Mach disks.

Figure 2.16 shows the pressure contours at the nozzle exit for the first operating condition of the ejector with back pressure of 30 mbar using the k-ε realizable turbulence model. As one can see, the pressure drops immediately at the exit as it expands downstream. Then, the compression fan forms to turn the flow inward towards the centerline with the occurrence of shock waves. As the flow passes through the high pressure region, it expands since the pressure is again below the ambient pressure (the pressure outside the free jet boundary). The whole process begins with expansion waves; thus the flow is underexpanded in this case. It can be observed that the second expansion-compression effect is weaker than the first one. This is due to the fact that the turbulent shear layer between the two flow streams creates a viscous damping effect that gradually dissipates the wave structure [12].

Figure 2.13: Shock diamonds in the jet engine exhaust of the SR-71 Blackbird during takeoff [12]

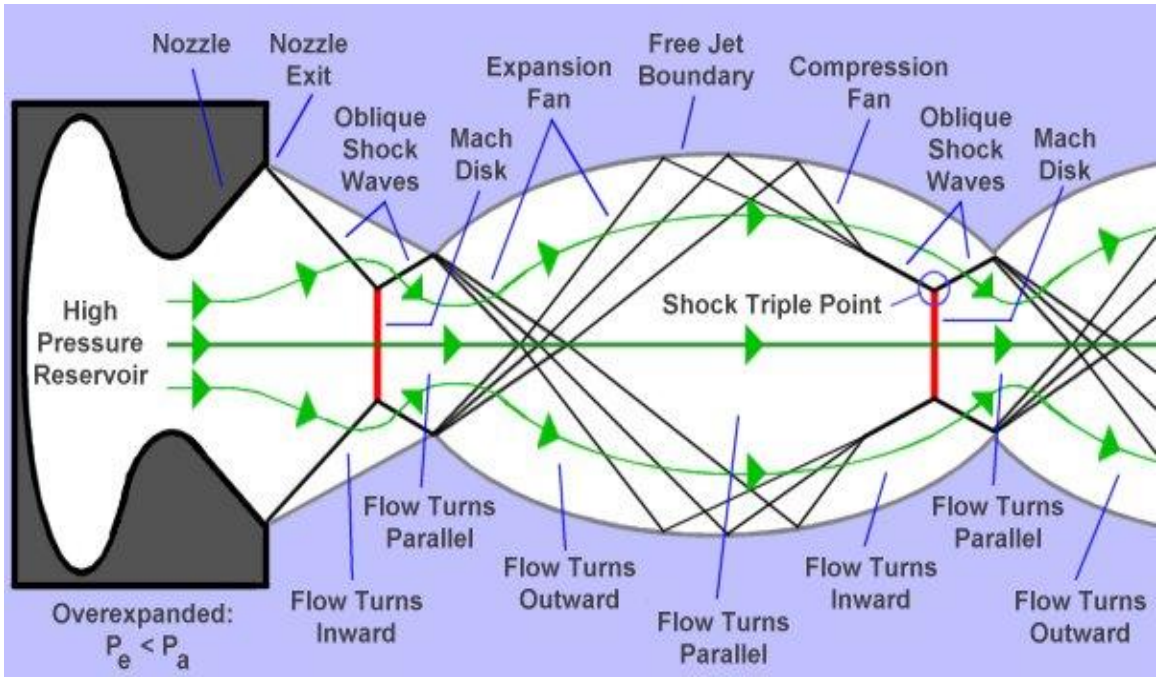

Figure 2.14: Wave structures that create shock diamonds in an overexpanded flow [12]

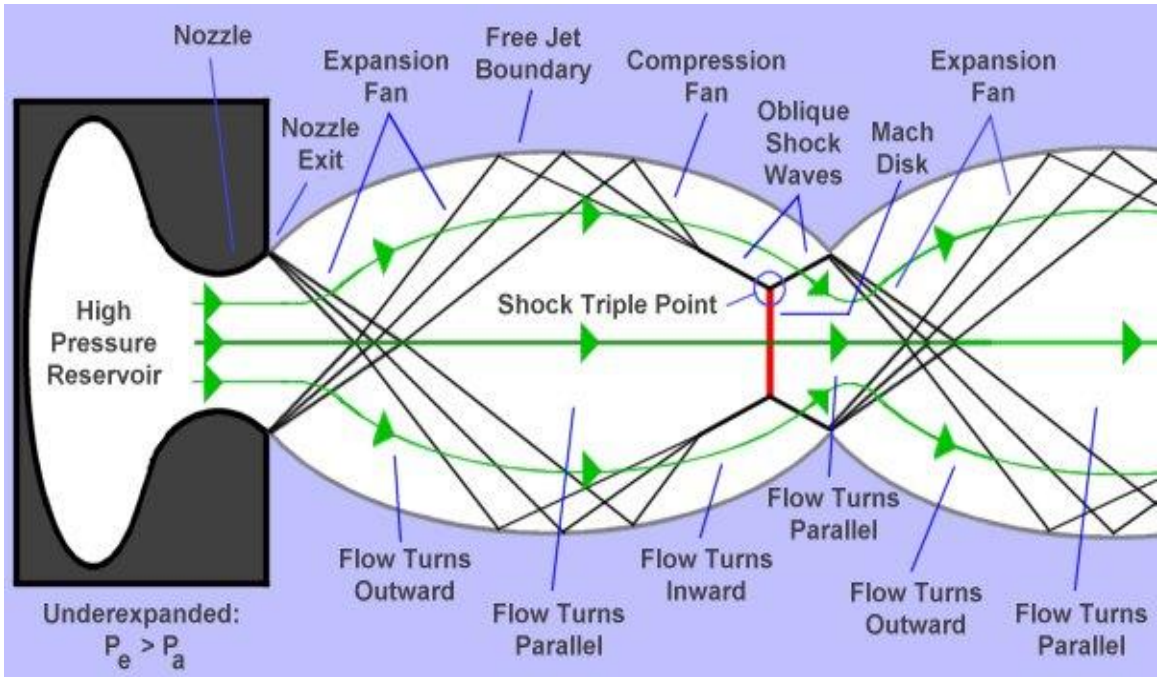

Figure 2.15: Wave structures that create shock diamonds in an underexpanded flow [12]

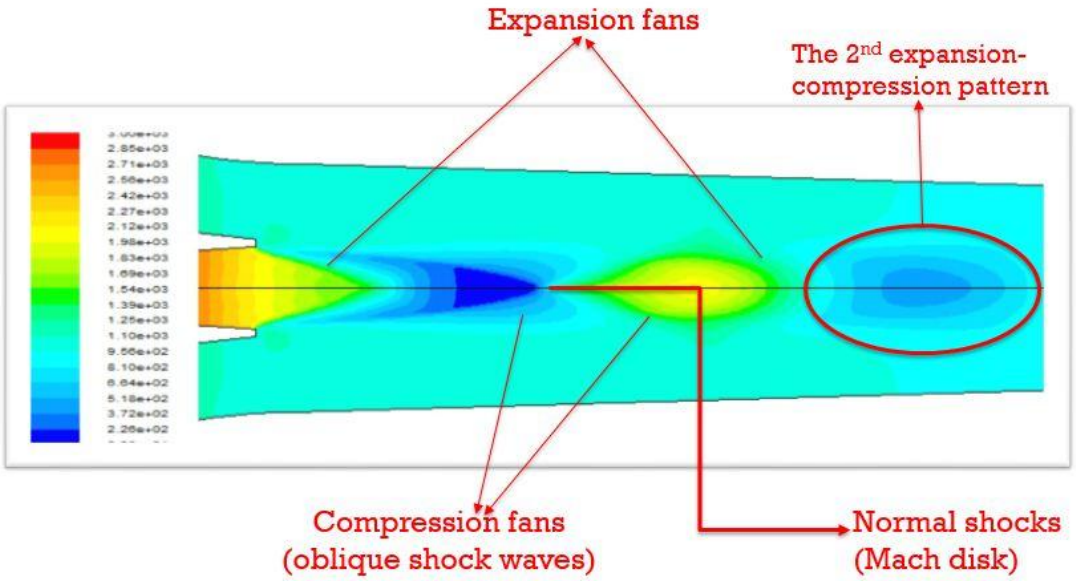

Figure 2.16: Pressure contour at the nozzle exit for the first operating condition using the k-ε realizable turbulence model (back pressure = 30 mbar)

2.4 Computations of Flow Fields of other Ejectors

2.4.1 Simulation of Air Ejector Used in Refrigeration Application

An air ejector used for refrigeration obtained from the paper of Gagan et al. [3] is simulated here to further verify the accuracy of k-ε realizable turbulence model with a different working fluid. The

geometry of the ejector and the nozzle is shown in Fig. 2.17. The same meshing procedure as described in section 2.1 is applied with a final mesh size of 45257 cells. The operating conditions of the ejector are as follows: motive pressure = 0.743 MPa, suction pressure = 0.0863 MPa, back pressure = 0.137 MPa, and air temperature at the inlet of the ejector = 20.7 °C [3]. The properties of the working fluid air are imported from the FLUENT database. Again, air is treated as an ideal gas and its properties are shown in Table 2.3. The same convergence criteria as described in section 2.2.3 are applied. The comparison of entrainment ratio (ER) between the experimental result and the CFD result is made in Table 2.4. The final mass flow rates at the ejector inlet (secondary flow) and the nozzle inlet (primary flow) are 0.0092955395 kg/s and 0.01642498 kg/s, respectively. As one can see, the error in the computed results obtained using the k-ε realizable model is very small compared to commonly accepted error in CFD simulations [8], which gives us confidence in the use of this turbulence model for accurate air ejector flow field simulations for refrigeration application.

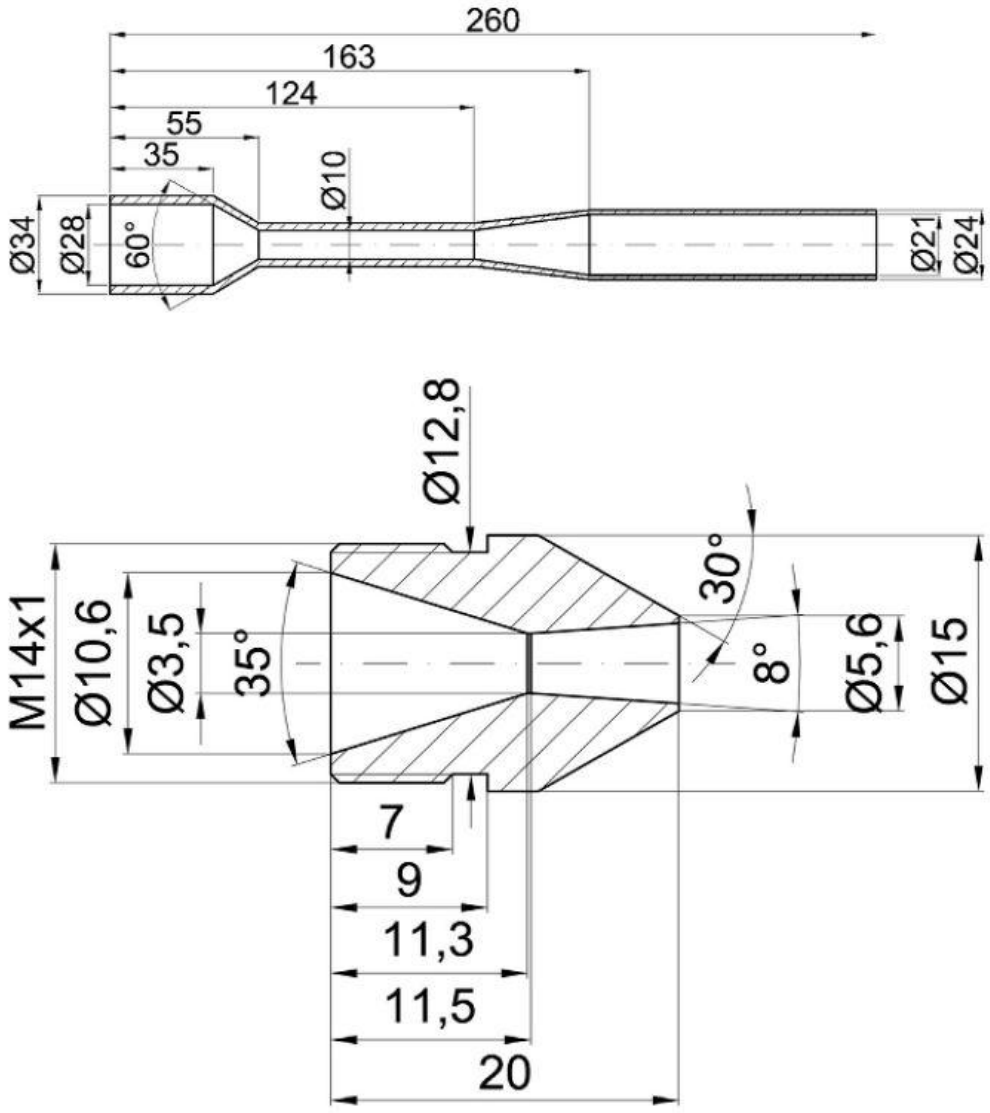

Figure 2.17: Geometry of the ejector and the nozzle

Table 2.3: Air properties

Properties	Values
Density	Ideal-gas Model
Cp (Specific heat)	1006.43 J/kg-K
Thermal conductivity	0.0242 W/m-k
Viscosity	1.7894e-5 kg/m-s
Molecular weight	28.966 kg/kgmol

Table 2.4: Comparison of CFD result and experimental result

CFD result	Experimental result	Error
ER= 0.566	ER= 0.553	2.35%

2.4.2 Simulation of Steam Ejector Used in Desalination Application

In this section, a steam ejector used for desalination described in the paper of Jeong et al. [8] is simulated. The geometric parameters et al. of the ejector and the nozzle are listed in Table 2.5. It should be noted that the important difference between the steam ejector described and considered before for the refrigeration application and the present steam ejector for desalination application is the size. For the refrigeration steam ejector discussed in the previous section, the overall length of the ejector is approximately 420 mm, while the desalination steam ejector considered in this section has an overall length of 5485.4 mm. The same meshing procedure is applied as in section 2.1 with a final mesh size of 54225 cells. The primary fluid inlet pressure and the secondary fluid inlet pressure are set at 2.66 bar and 0.16bar respectively. The operating conditions are varied by changing the discharge pressure (backpressure) as 0.2 bar, 0.25 bar, 0.26 bar, 0.275 bar, 0.3 bar and 0.31 bar. The same convergence criteria as described in section 2.2.3 are applied. The velocity contours and ER plot are analyzed. Figure 2.18 shows the velocity contours of the flow field at different operating conditions. The effect of increasing the discharge pressure is again that the movement of the 2nd shock upstream interferes with the mixing causing the secondary stream to be no longer choked, which leads to the decrease in ER as shown in Fig. 2.19. The CFD results using the k-ε realizable turbulence model are in good agreement with the experimental data within the acceptable margin of error.

Table 2.5: Desalination ejector and nozzle specifications [8]

Specifications	Values
Primary nozzle inlet diameter	80 mm
Primary nozzle throat diameter	36.5 mm
Primary nozzle exit diameter	87.2 mm
Primary nozzle area ratio (A_{exit}/A_{throat})	5.71
Overall length of ejector	5485.4 mm
Secondary inlet diameter	404.4 mm
Ejector outlet diameter	436 mm

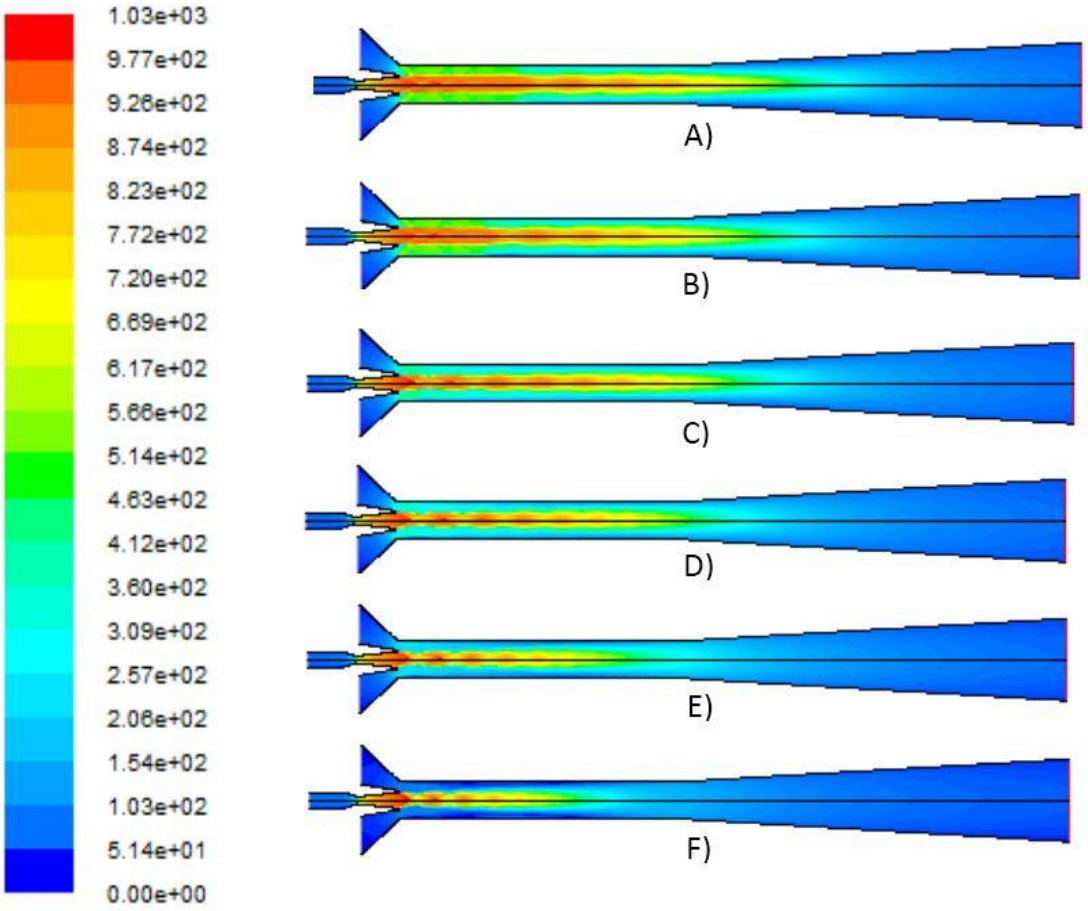

Figure 2.18: Velocity contours inside the steam ejector used in desalination application employing the k- ε realizable turbulence model with back pressure of A) 0.2 bar, B) 0.25 bar, C) 0.26 bar, D) 0.275 bar, E) 0.3 bar and F) 0.31 bar showing the movement of shock upstream

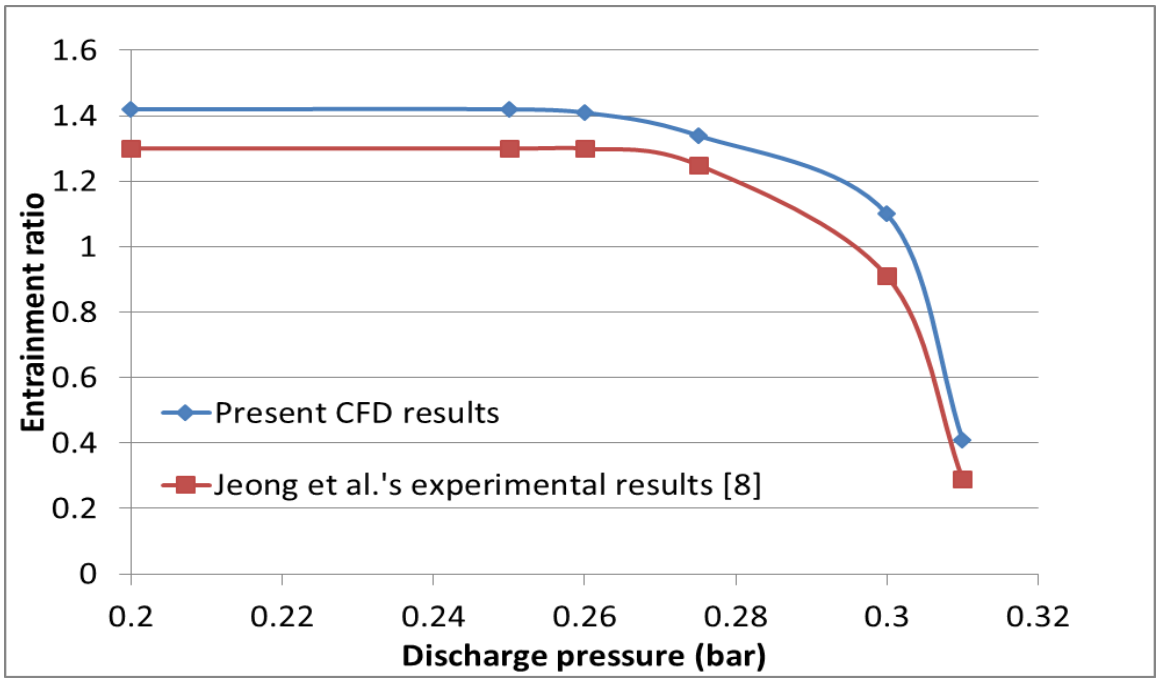

Figure 2.19: Comparison of entrainment ratio using the k-ε realizable turbulence model with the experimental results for steam ejector used in desalination application [8]

2.4.3 Simulation of Air Ejector Used in Desalination Application

Another air ejector used in desalination application, described in the paper of Aly et al. [11], is simulated for further validation and verification of the model accuracy. The geometric parameters of the ejector and the nozzle are listed in Table 2.6. The same meshing procedure as described in section 2.1 is applied with a final mesh size of 53471 cells. The primary fluid inlet pressure and the condenser outlet pressure are set at 7 bar and 0.17 bar respectively. The operating conditions are varied by changing the secondary fluid inlet pressure (suction pressure) as 0.4 atm, 0.5 atm, 0.6 atm, 0.7 atm, 0.8 atm and 0.9 atm. The same convergence criteria as described in section 2.2.3 are applied.

The numerical results are compared with the experimental data given in the paper by Aly et al. [11]. The effect of the suction pressure is investigated here by examining the velocity contours and the entrainment ratio plots shown in Fig. 2.20 and Fig. 2.21 respectively. It is clear that the entrainment ratio increases as the suction pressure is increased. When the suction pressure is increased, it can be seen from the velocity contours that the primary jet core gets compressed and becomes smaller and smaller from A) to F) in Fig. 2.20. A thinner jet core gives the largest effective area, where more secondary fluid is entrained into the ejector. The increase in the mass flow rate of the secondary flow results in a larger entrainment ratio. The accuracy of k-ε realizable turbulence model is again validated in Fig. 2.21, where the CFD results are in excellent agreement with the experimental data.

Table 2.6: Desalination air ejector and nozzle specifications [11]

Specifications	Values
Primary nozzle inlet diameter	19 mm
Primary nozzle throat diameter	3.5 mm
Primary nozzle exit diameter	5.26 mm
Nozzle exit position (NXP)	0 mm
Mixing chamber inlet diameter	16.4 mm
Throat diameter	9 mm
Throat length	67.5 mm
Diffuser length	90 mm
Ejector outlet diameter	25 mm

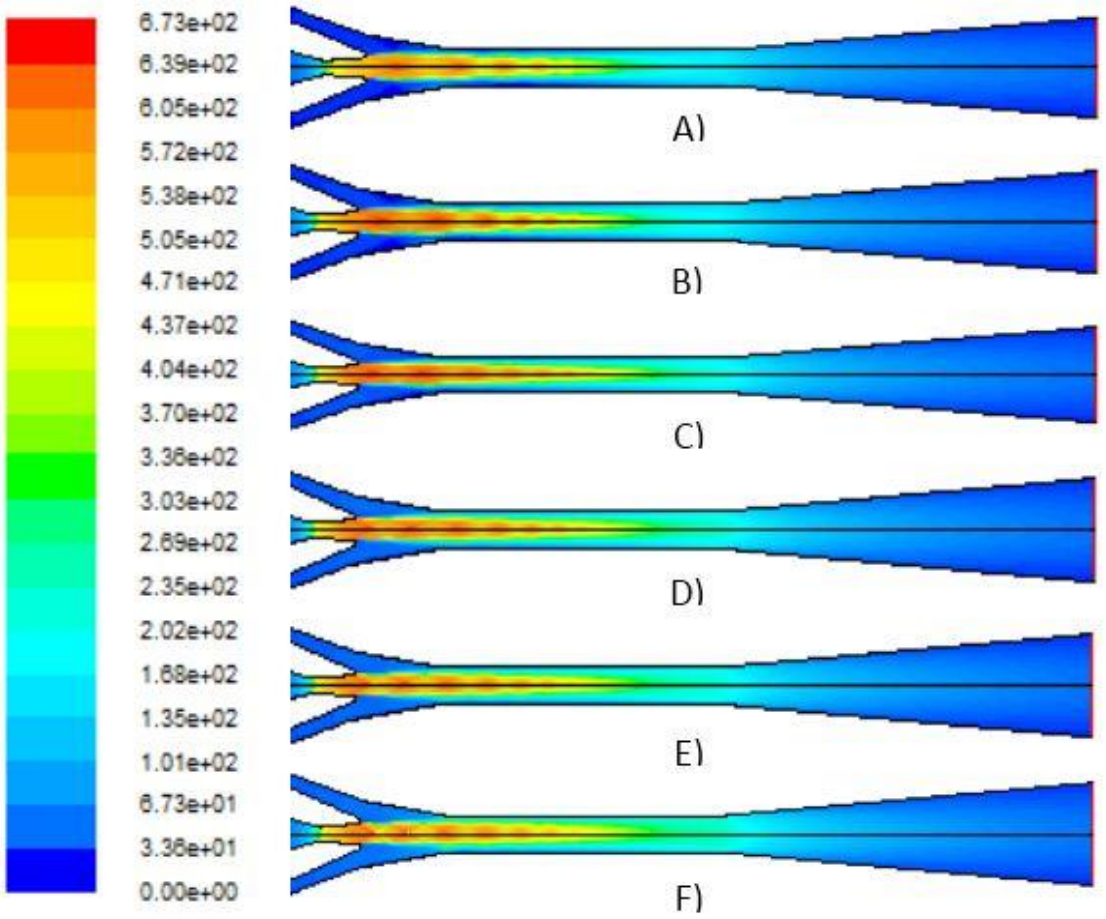

Figure 2.20: Velocity contours inside the air ejector used in desalination application employing the k- ε realizable turbulence model with suction pressure of A) 0.4 atm, B) 0.5 atm, C) 0.6 atm, D) 0.7 atm, E) 0.8 atm and F) 0.9 atm

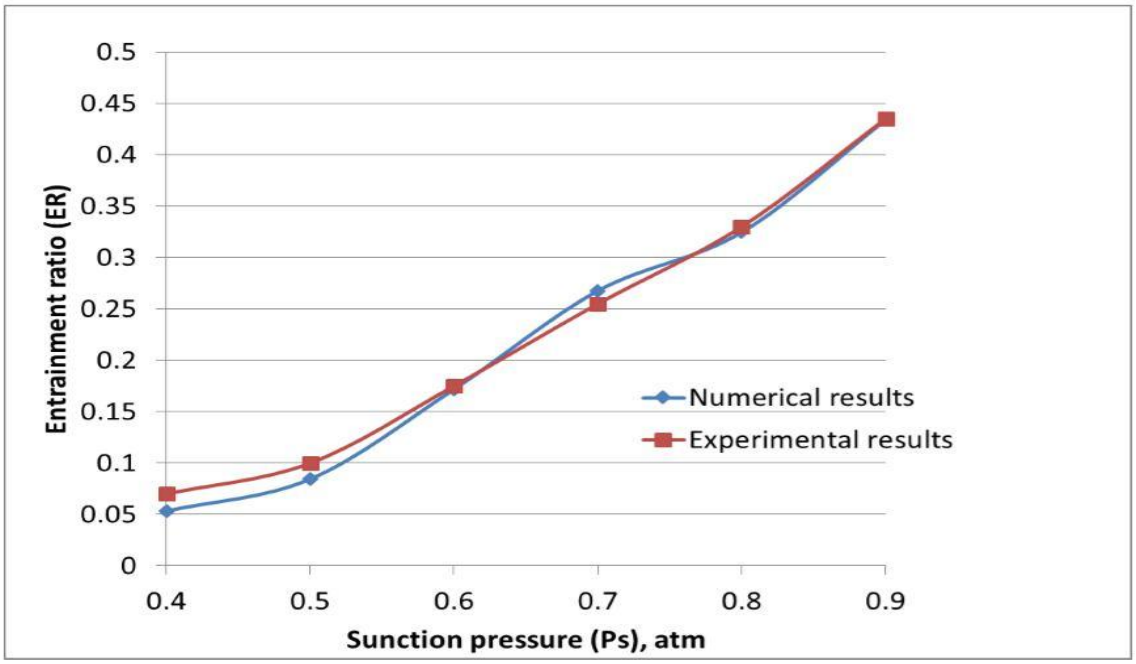

Figure 2.21: Comparison of computed entrainment ratio using the k-ε realizable turbulence model with the experimental results for air ejector used in desalination application [11]

Chapter 3 Shape Optimization of Ejectors Using a Genetic Algorithm

Optimization of a supersonic ejector has been conducted for decades by considering different geometrical parameters such as the diameter of the primary nozzle inlet, the throat length, the mixing chamber inlet diameter, etc. [13]. Different optimization methods have been employed to achieve better ejector performance e.g. the Gauss optimization method [14]. In this chapter, Genetic Algorithm based optimization method is applied to optimize the mixing chamber wall shape of the first steam ejector simulated in previous chapter in section 2.3. The section of the ejector wall that is shape optimized in this chapter is shown in Fig. 3.1. An optimal entrainment ratio is obtained using the single objective genetic algorithm (SOGA) in conjunction with the flow solver FLUENT and the mesh generation software ICEM.

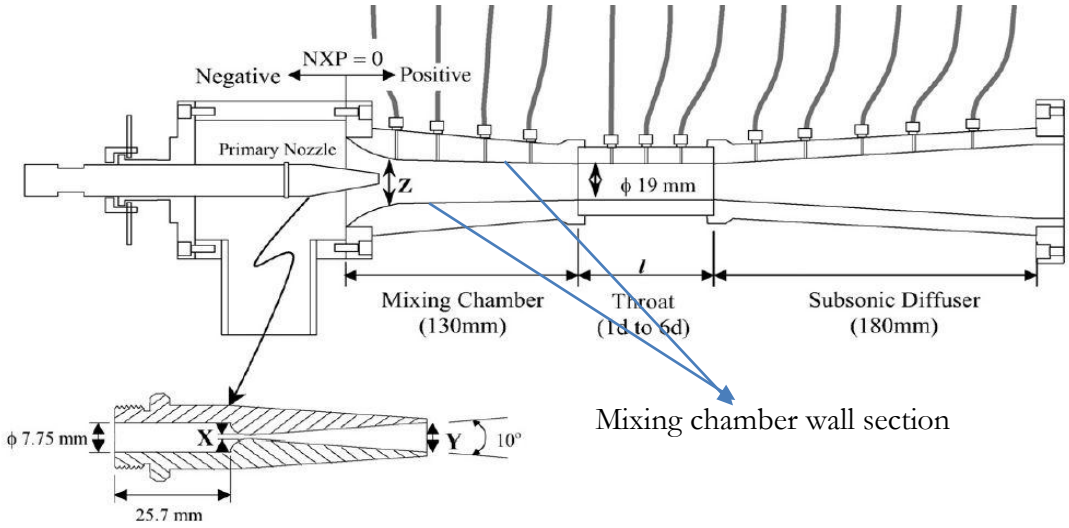

Figure 3.1: Schematic diagram of a steam ejector for refrigeration application showing the mixing chamber wall that is shape optimized to increase entrainment [5]

3.1 Overview of the Genetic Algorithm

Genetic algorithms (GAs) are a class of stochastic optimization algorithms inspired by the biological evolution [15]. They efficiently exploit historical information to speculate on new offspring with improved performance [16]. In the present, the mixing chamber wall shape of the ejector is optimized to achieve improved ejector performance (larger entrainment ratio). GAs are different from most of other optimization and search procedures in four ways [16]:

• GAs work with coding of a parameter set, not with the parameters themselves.

• GAs work simultaneously with multiple points, and not with a single point.

• GAs search via sampling (blind search) using only the payoff information.

• GAs search using stochastic operators, not deterministic rules, to generate new solutions.

Since a GA works simultaneously on a set of coded solutions, it has little chance of getting stuck at a local optimum when used as an optimization technique [16]. Single objective genetic algorithm is simply a specific kind of genetic algorithm with just one objective or fitness value to achieve. The basic steps in a SOGA are shown in Fig. 3.2. A fixed number of different mixing wall curves are randomly selected as the first generation (initial population); these curves are parameterized using the Bezier curve method (coding of the parameters). The entrainment ratio for each curve is calculated by numerical simulation using FLUENT and compared. After that, a certain criterion is applied to select some of the old curves to be part of the individuals of the new generation (select strings to create new mating pool). Crossover and mutation are then employed to the new mating pool based on the fitness values to generate new individuals (offspring), which together with their parent curves, creates a new generation with the same number of individuals as in the first generation. The whole process is repeated until the entrainment ratio converges to a maximum value.

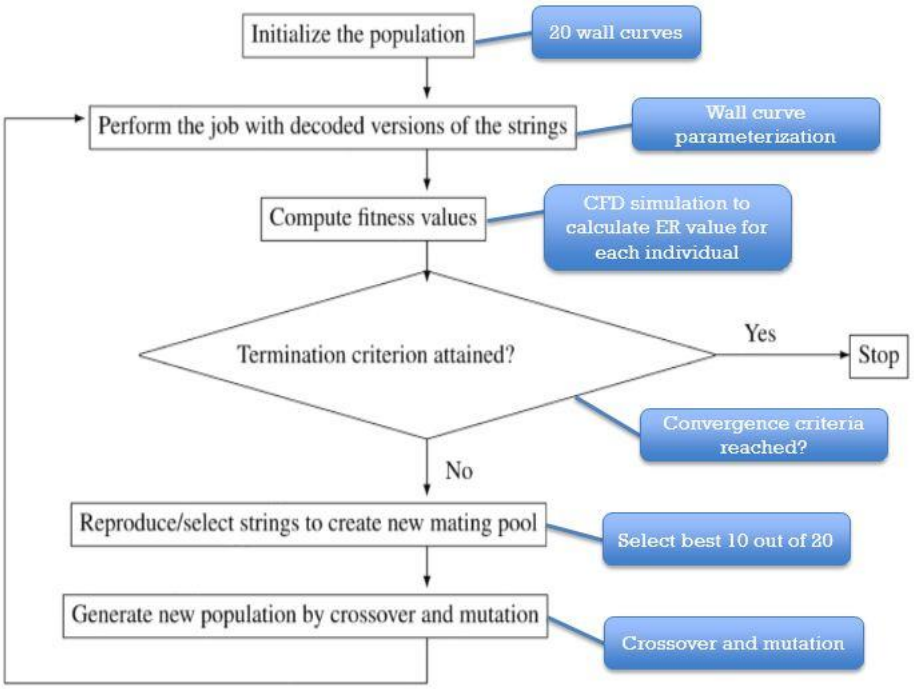

Figure 3.2: Basic steps in a single objective genetic algorithm [16]

3.2 Mixing Chamber Wall Curve Parameterization and GA Parameters

The mixing chamber wall shape curves are parameterized using Bezier curves. Bezier curves are parametric curves frequently used in computer graphics and related fields [15]. A Bezier curve is defined by a set of control points. The first and last control points are usually the end points of the curve [18]. Since we want to investigate the effect of the wall shape solely, the mixing chamber inlet diameter and the throat diameter are kept constant, which means the first and last control points are kept fixed. Three intermediate control points are employed since we only expect one or two concavities on the curve. The curve can be expressed based on the five control points using a

specific polynomial [18]. To obtain the first generation of twenty curves, the three intermediate control points of the original wall curve are varied within a box randomly. The box limit is set to be 15mm in four directions around the control point to avoid a local optimum (since the geometry of the curve is 130mm in length and 24 mm in height in x-y coordinate system).

Once the wall curve is parameterized and the initial generation is created, GA input parameters are determined [17]. Following input parameters are used in the GA optimization:

- Population size (number of individuals in each generation): 20
- Number of generations: maximum of 40 generations if convergence is not obtained
- Mutation probability: 0.5 for each point with a randomizer = $\frac{\mathbf{15}}{\boldsymbol{generation\#^{0.4}}}$, which means that the mutating range (box limit) decreases as the generation # increases
- Crossover probability: 0.5 for each point with three types of crossover:
  - Exchange the location control points (both x and y value)
  - Exchange only the y value of the control points
  - Average the location of the control points (both x and y value)
- Selection method: Roulette wheel sampling
- Convergence criteria: (assume nth generation is the final generation)
  - The best entrainment ratio in the n^{th} generation is equal to the best entrainment ratio in the $(n-1)^{th}$ generation.
  - The best entrainment ratio in the n^{th} generation has less than 1% improvement compared to the best entrainment ratio in the $(n-3)^{th}$ generation.

- The average entrainment ratio in the n^{th} generation has less than 1% of improvement compared to the average entrainment ratio in the $(n-3)^{th}$ generation.

A MATLAB code is used to perform the GA optimization as well as to automate the ICEM meshing and FLUENT flow field calculation. All GA parameters are user-defined based on the GA methodology. The optimization process is shown in Fig. 3.3. SOGA creates new control points, which are used to generate new wall shapes through the Bezier curve method. The new curves along with the rest of the ejector geometry are imported into ICEM for meshing. Then the FLUENT is run to calculate entrainment ratio of each new shape, which is then used as a fitness function in SOGA optimization to create the curves for next generation and the process is repeated until a convergence criteria is met as described above.

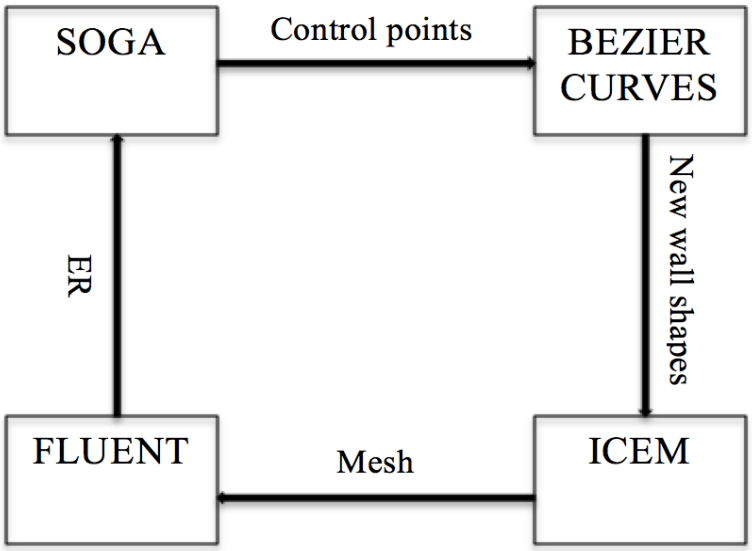

Figure 3.3: Schematic of optimizing process

3.3 Optimization Results

The GA optimizing process is run three times using three different initial generations to reduce the possibility of reaching a local optimum. Table 3.1 shows the entrainment ratios for 60 randomly generated individuals in three runs. –infinity implies that there is no ER output from that shape due to some errors occurring during the optimizing process. Some negative values of ER may also occur because of the reversed flow at the secondary fluid inlet.

Table 3.1: Entrainment ratios of the first generation for each run

Run #1		Run #2		Run #3	
Curve #	**ER**	**Curve #**	**ER**	**Curve #**	**ER**
1	0.3824	1	0.38137	1	0.06472
2	0.24255	2	0.23264	2	0.064729
3	0.39269	3	0.38423	3	0.37123
4	0.38023	4	0.37751	4	-Inf
5	0.064555	5	-0.51699	5	0.39287
6	0.3803	6	0.19757	6	0.37596
7	-Inf	7	0.005525	7	0.2344
8	0.3947	8	0.045008	8	0.39346
9	0.34925	9	0.066275	9	0.012452
10	0.28915	10	0.12543	10	0.37276
11	0.1392	11	0.39573	11	-Inf
12	0.39005	12	0.1054	12	-Inf
13	0.38312	13	0.38451	13	0.40107
14	0.03778	14	-Inf	14	0.31452
15	0.1651	15	0.37558	15	0.16932
16	0.3807	16	0.19404	16	0.38137
17	-Inf	17	0.3939	17	0.23264
18	-Inf	18	0.39667	18	0.38423
19	0.096033	19	0.29777	19	0.37751
20	0.18072	20	-1.4208	20	-0.51699

Figure 3.4 shows the optimized mixing chamber wall curve for the first run after 14 generations with a final ER value = 0.40721. The plot of best and average ER for each generation is shown in Fig. 3.5 for convergence check. The best curve is compared to the original curve in Fig. 3.5, and four zoomed-in views of Fig. 3.6 at different sections of the mixing chamber are shown in Figs. 3.7 ~ 3.10. As one can see, the optimized curve changes more gradually than the original curve as it begins from the starting point to the end point. More specifically, it crosses the original curve at around x = 25.4 mm point, which means that the cross-sectional area of the original wall is larger than the optimized wall before this position (x = 25.4), and is smaller after this position (x =25.4). Intuitively, one can conclude that since the new curve changes more gradually due to that the variation in flow is more smooth, at the same time, 4/5 part of the cross-sectional area of the mixing chamber of the new wall curve is larger than the original cross-sectional area, which results in larger entrainment of secondary fluid.

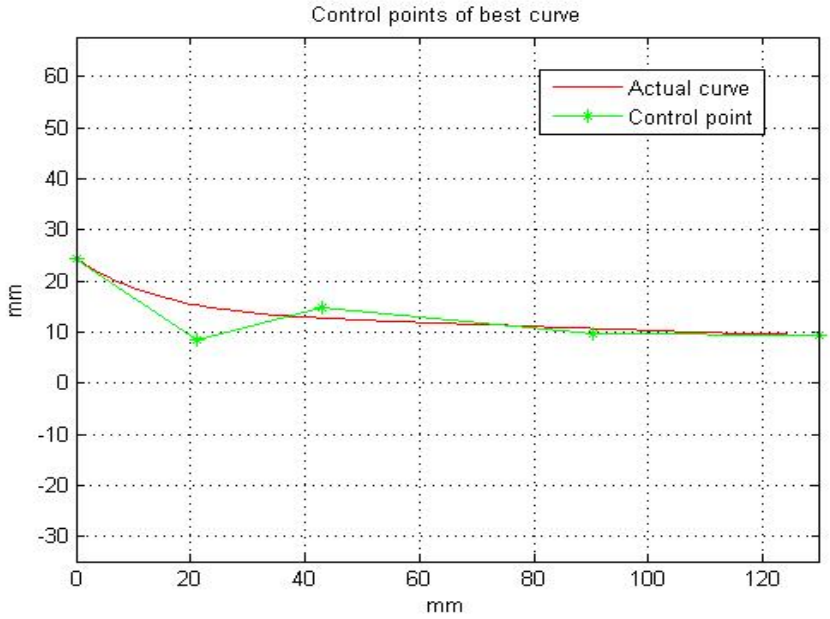

Figure 3.4: Plot of the best curve after optimization run #1 and its Bezier curve control points

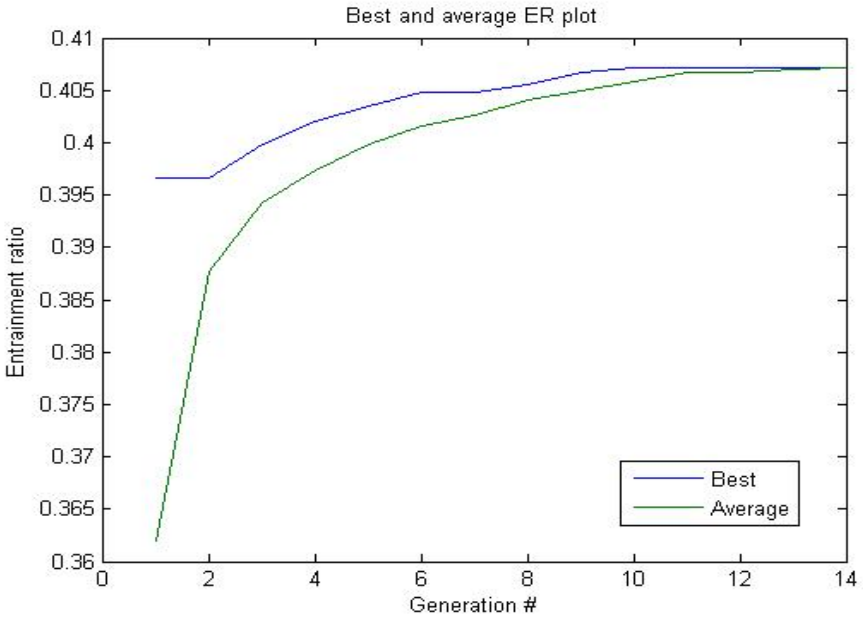

Figure 3.5: Plot of the best and average entrainment ratio for each generation beginning from run #1

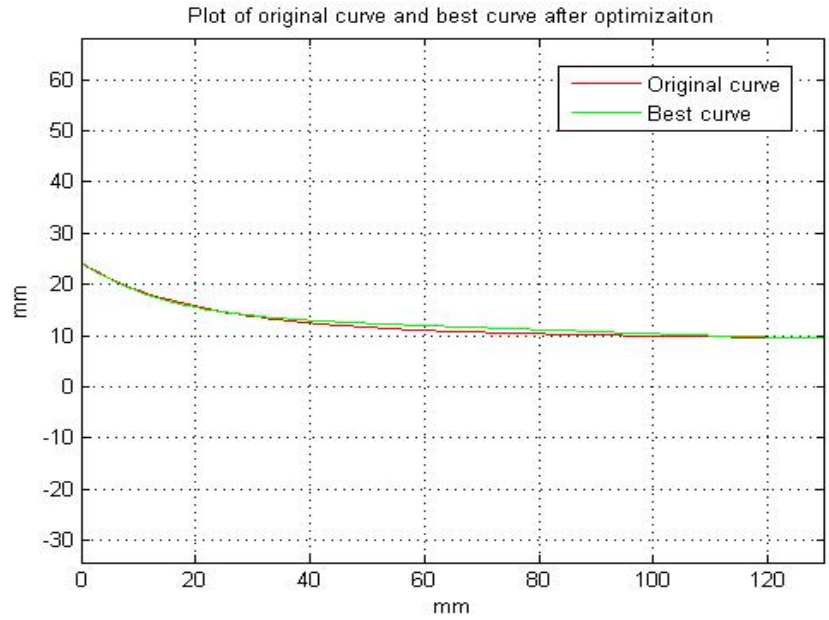

Figure 3.6: Plot of the original curve and the best (optimal) curve after optimization run #1

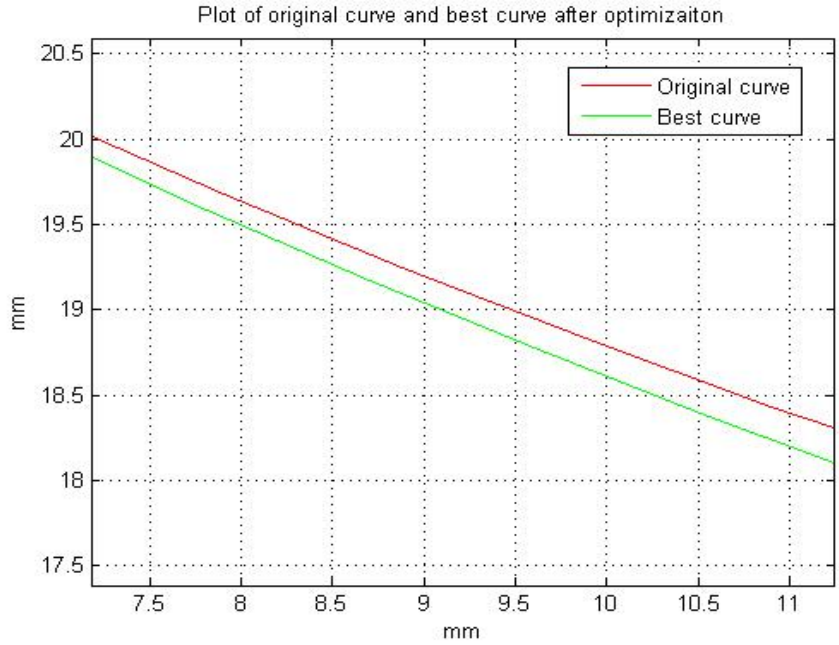

Figure 3.7: Zoomed-in view of Fig. 3.6 part (A)

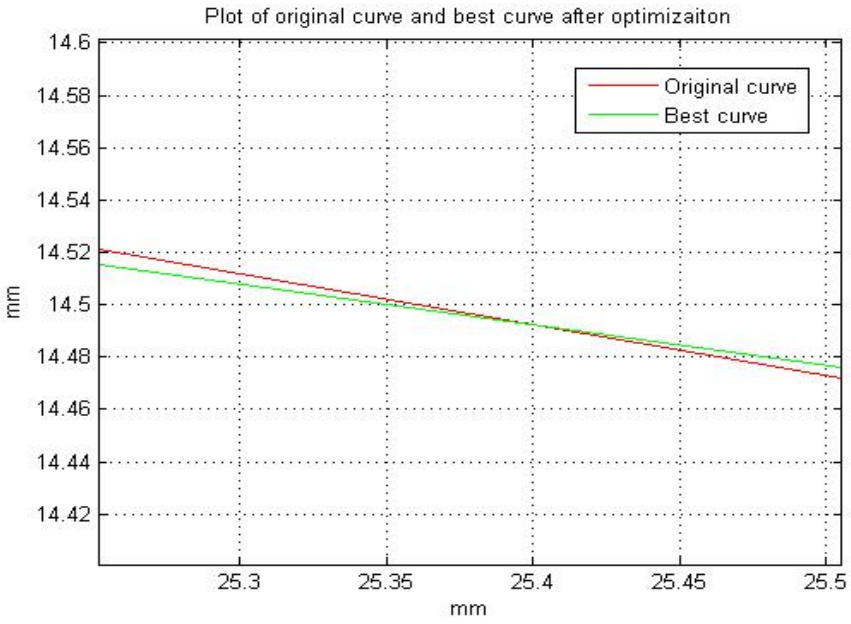

Figure 3.8: Zoomed-in view of Fig. 3.6 part (B)

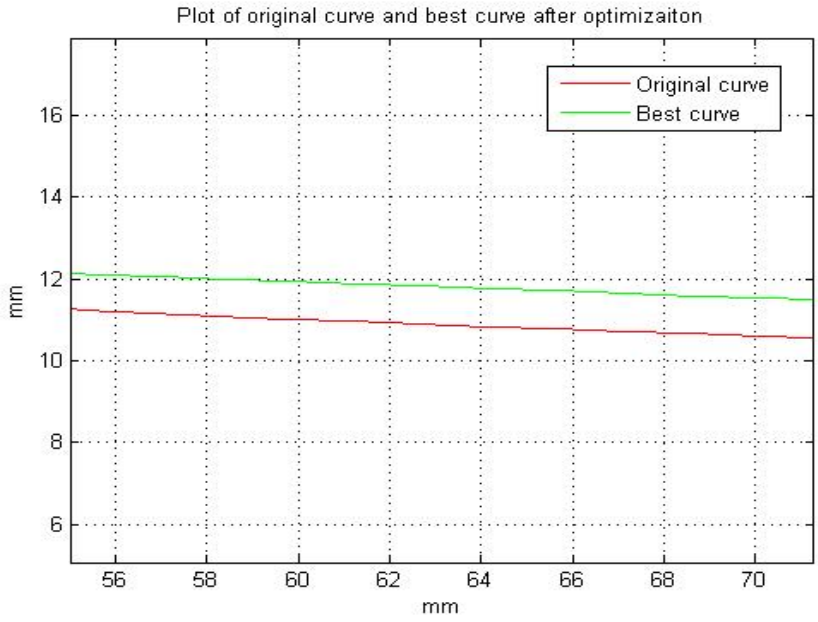

Figure 3.9: Zoomed-in view of Fig. 3.6 part (C)

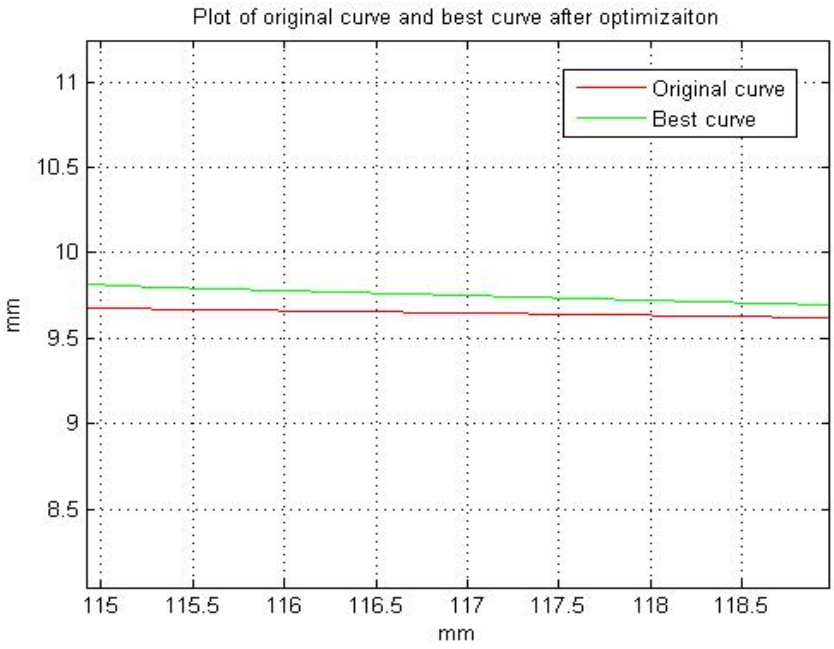

Figure 3.10: Zoomed-in view of Fig. 3.6 part (D)

Figure 3.11 shows the optimized mixing chamber wall curve for the second run after 17 generations with a final ER value = 0.40723. The plot of the best and the average ER for each generation is shown in Fig. 3.12 for convergence check. The best curve is compared to the original curve in Fig. 3.13, and four zoomed-in views of Fig. 3.6 at different sections of the mixing chamber are shown in Figs. 3.14 ~ 3.17. Once again, the optimized curve changes more gradually than the original curve as it begins from the starting point to the end point, however in this case the cross-sectional area of the new wall is larger than the original wall all the entire wall. Considering these two aspects, one can explain the increment in ER intuitively as was done for run #1.

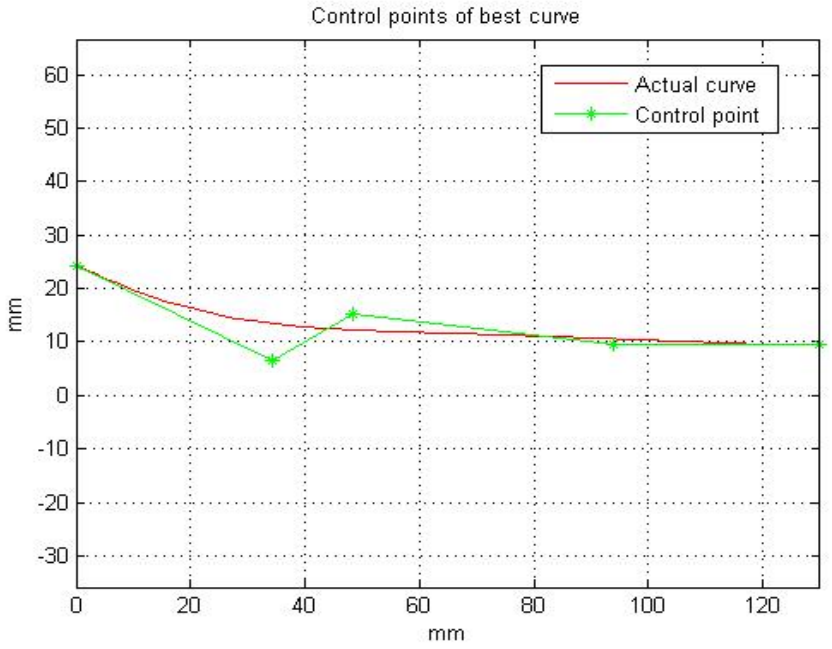

Figure 3.11: Plot of the best curve after optimization run #2 and its Bezier curve control points

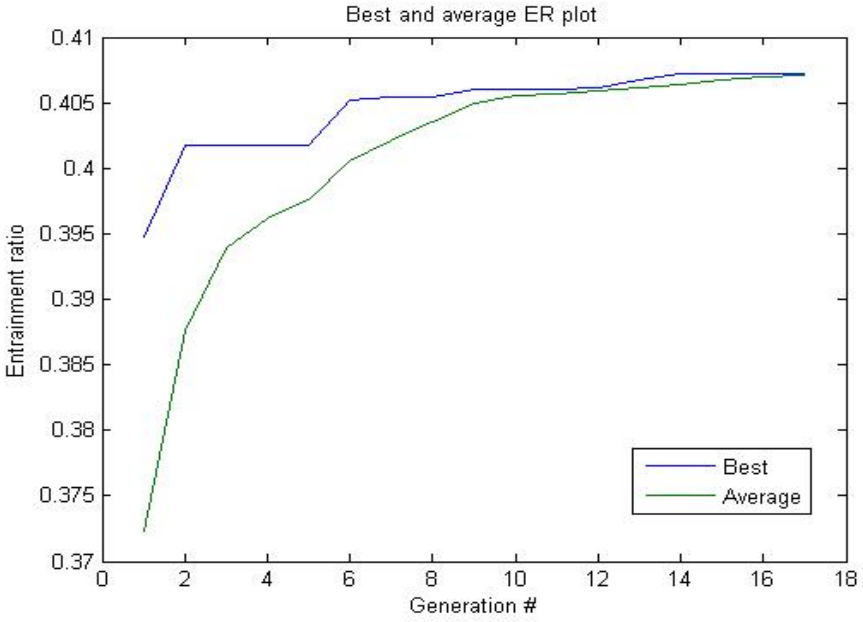

Figure 3.12: Plot of the best and average entrainment ratio for each generation for run #2

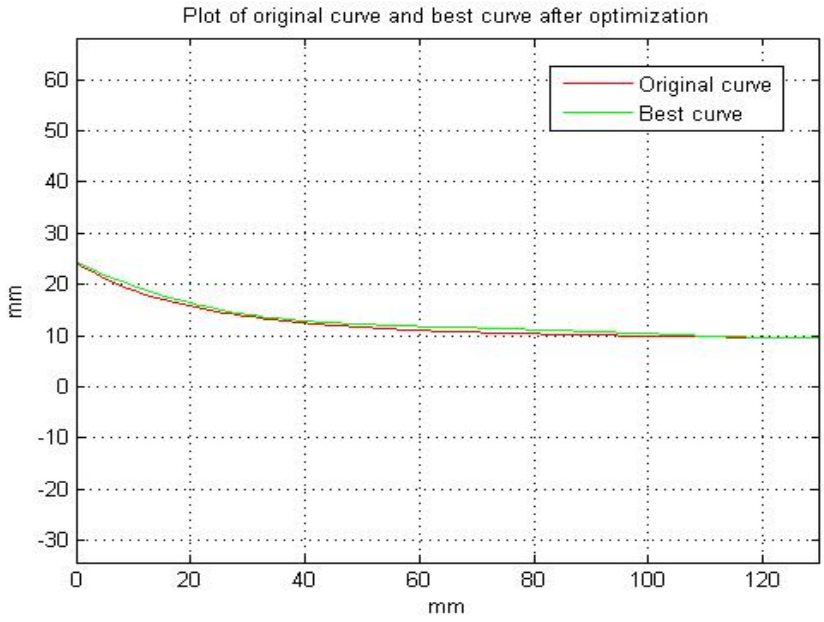

Figure 3.13: Plot of the original curve and the best (optimal) curve after optimization run #2

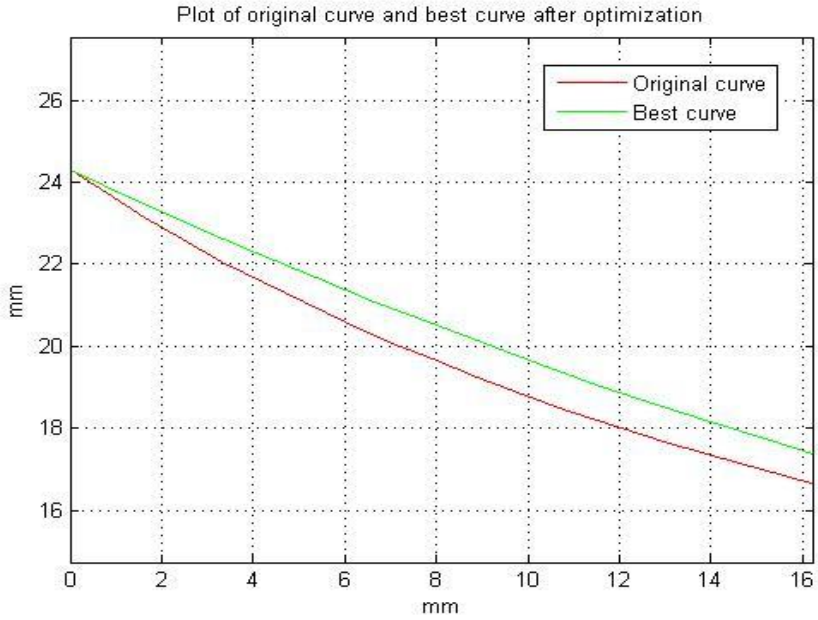

Figure 3.14: Zoomed-in view of Fig. 3.13 part (A)

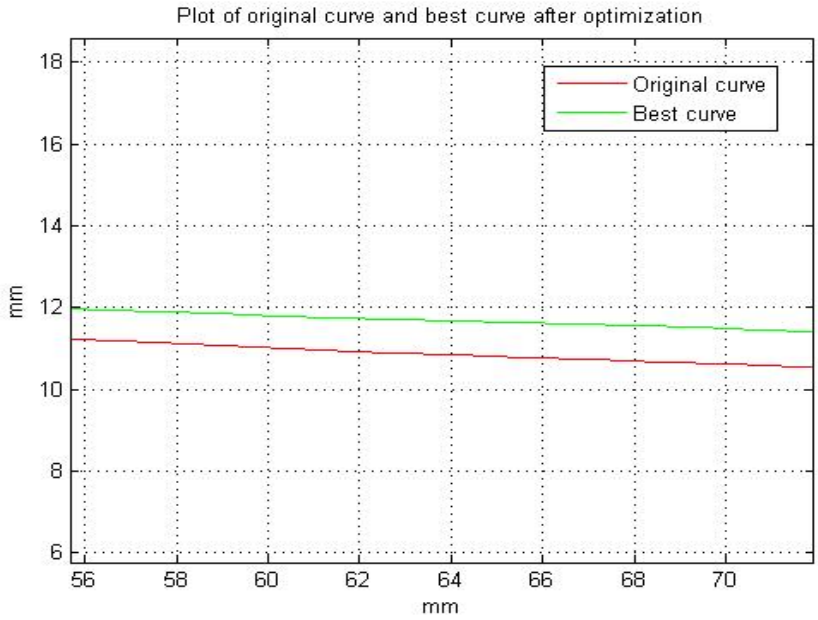

Figure 3.15: Zoomed-in view of Fig. 3.13 part (B)

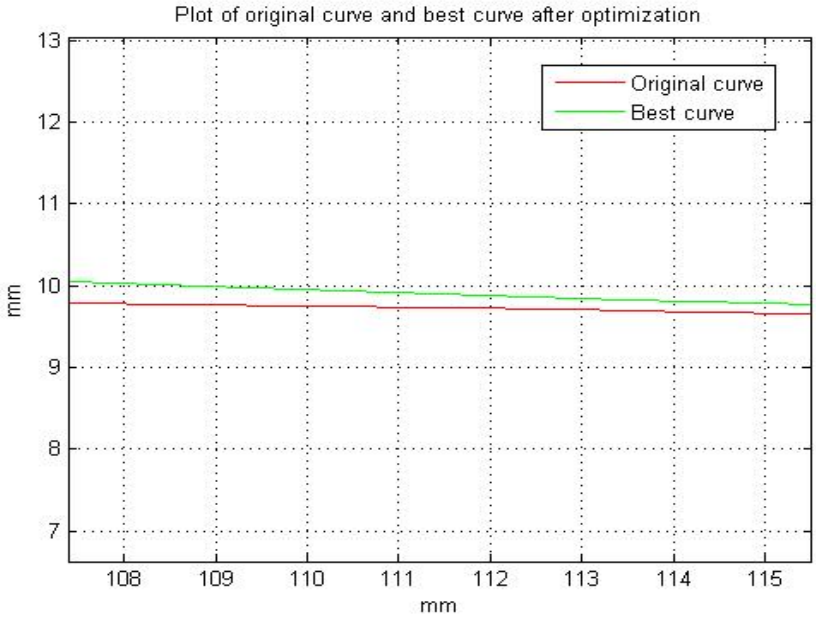

Figure 3.16: Zoomed-in view of Fig. 3.13 part (C)

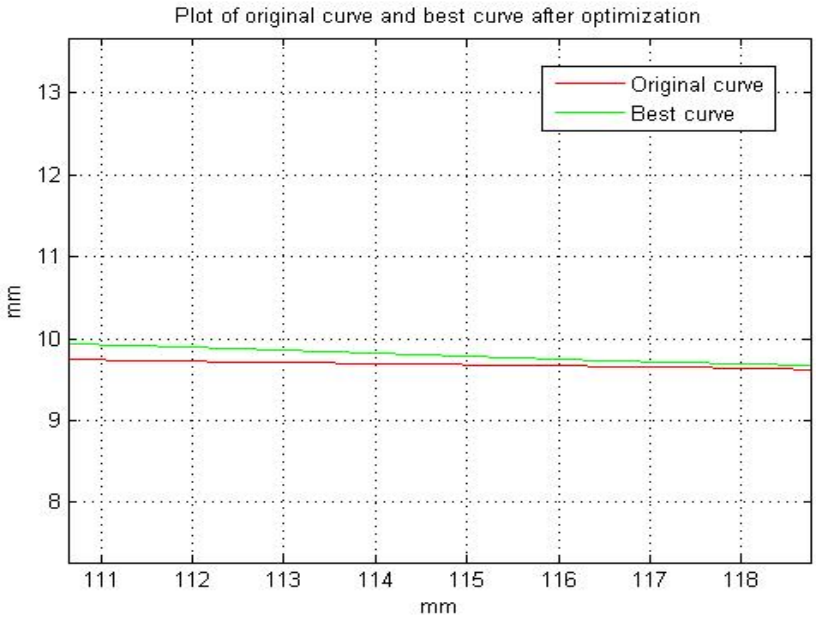

Figure 3.17: Zoomed-in view of Fig. 3.13 part (D)

Figure 3.18 shows the optimized mixing chamber wall curve for the third run after 7 generations with a final ER value = 0.40789. The plot of best and average ER for each generation is shown in Fig. 3.19 for convergence check. The best (optimal) curve is compared to the original curve in Fig. 3.20, and four zoomed-in views of Fig. 3.20 at different sections of the mixing chamber are shown in Figs. 3.21 ~ 3.24. Similar to the results for the second run, the optimized curve changes more gradually and has a larger cross-sectional area all along the entire mixing chamber.

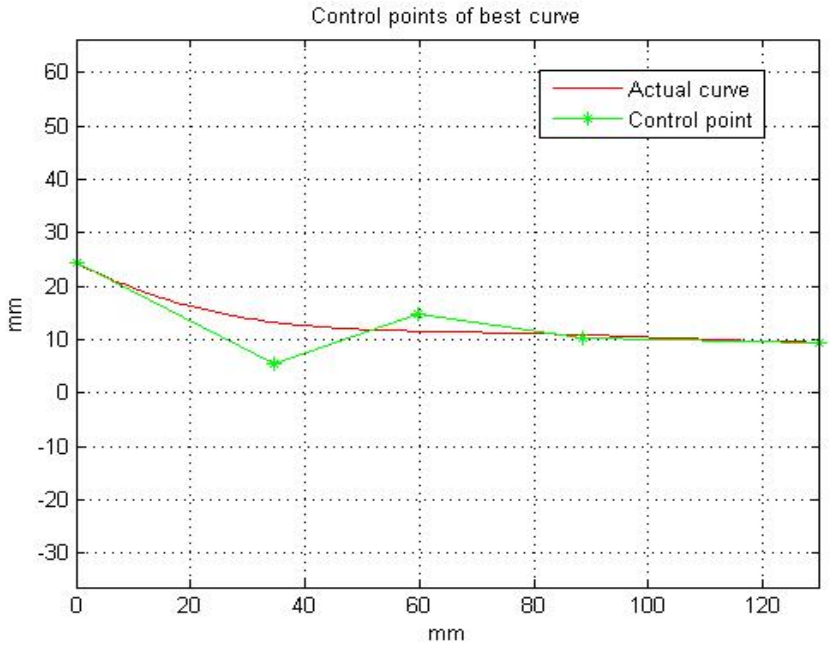

Figure 3.18: Plot of the best curve after optimization run #3 and its Bezier curve control points

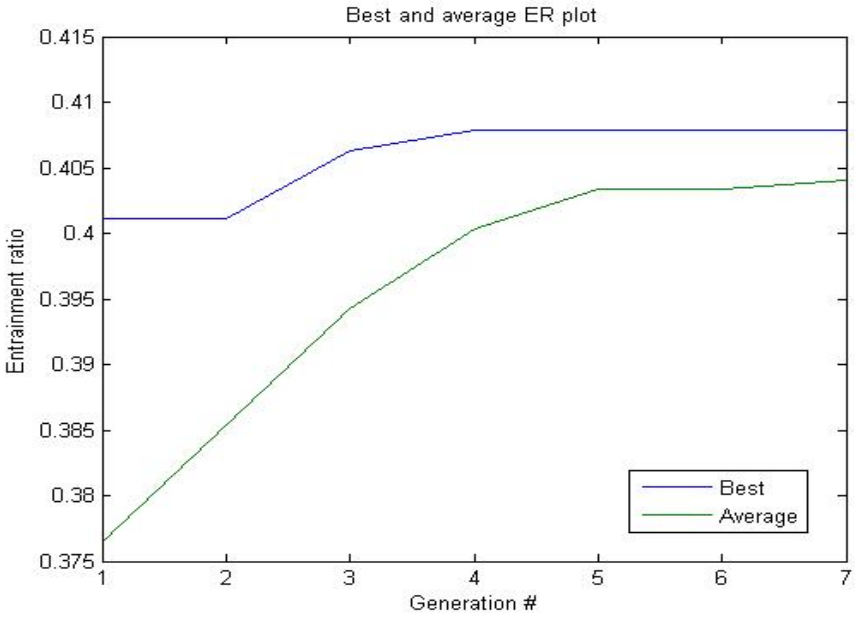

Figure 3.19: Plot of the best and average entrainment ratio for each generation for run #3

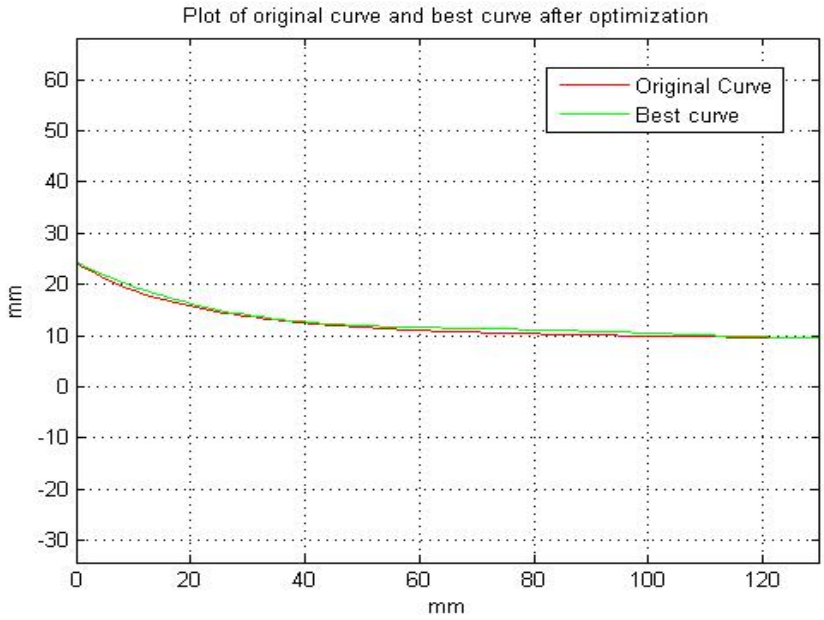

Figure 3.20: Plot of the original curve and the best (optimal) curve after optimization run #3

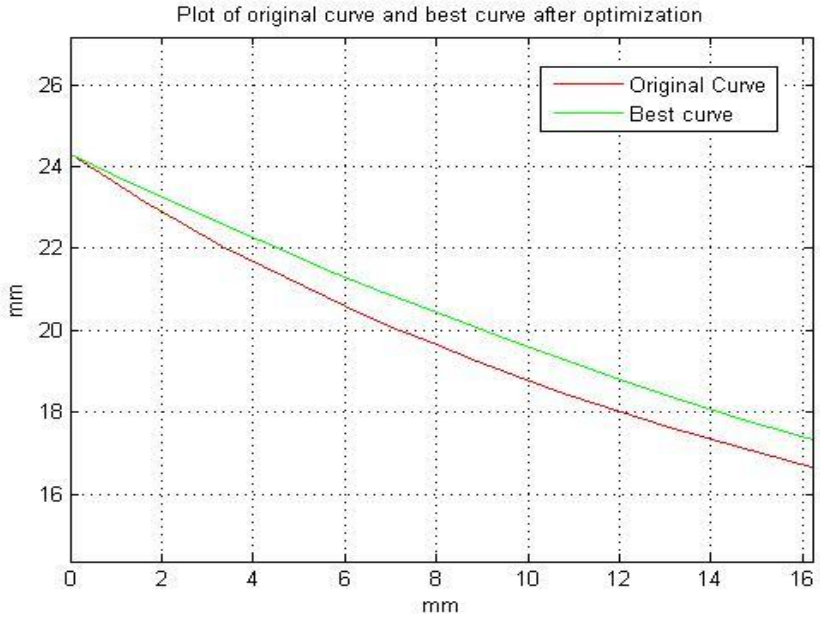

Figure 3.21: Zoomed-in view of Fig. 3.20 part (A)

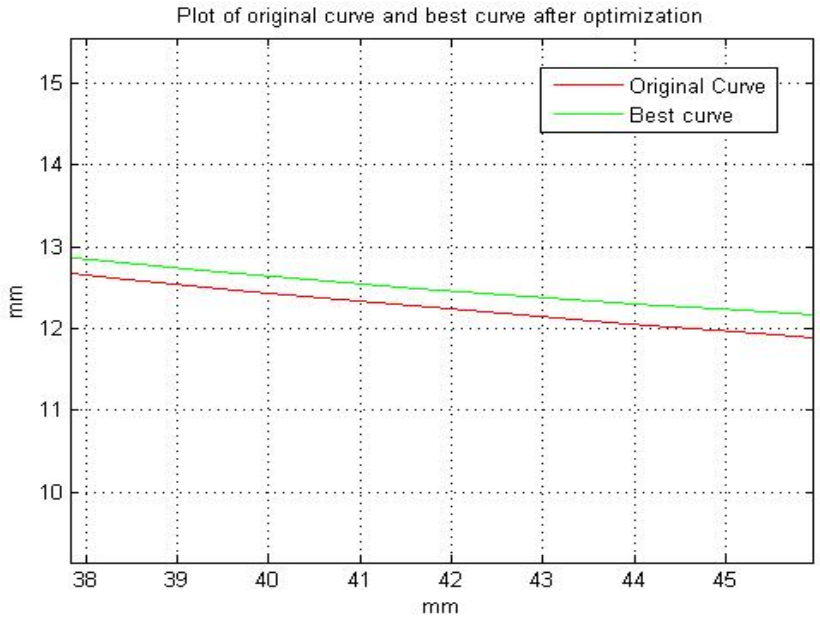

Figure 3.22: Zoomed-in view of Fig. 3.20 part (B)

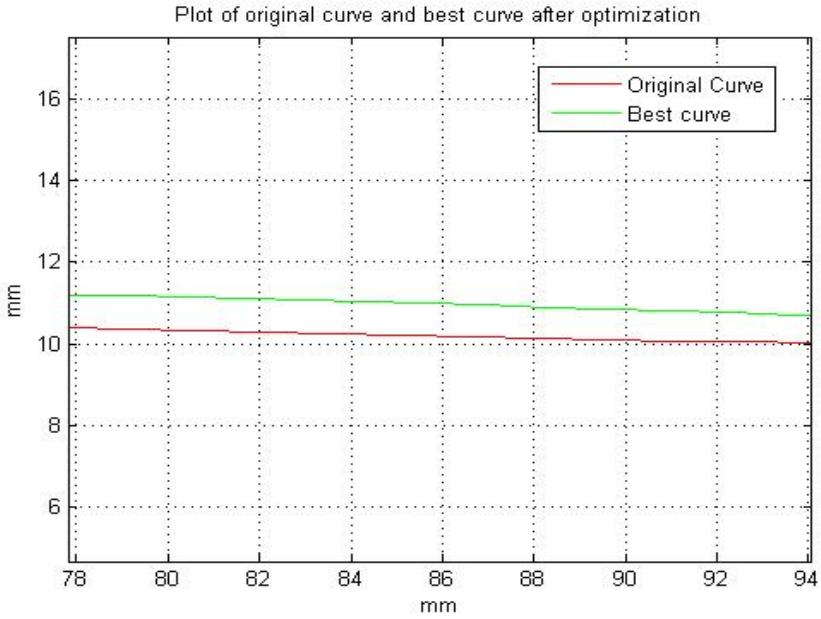

Figure 3.23: Zoomed-in view of Fig. 3.20 part (C)

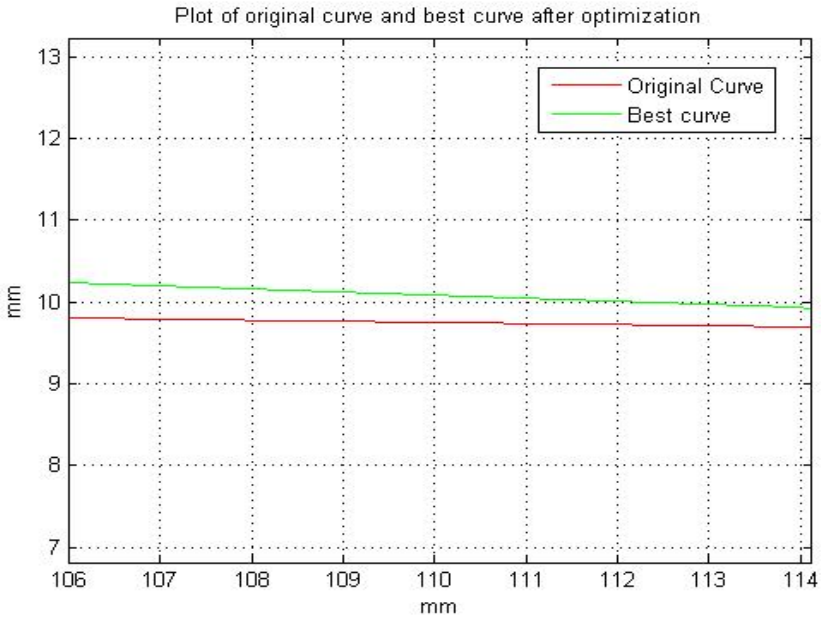

Figure 3.24: Zoomed-in view of Fig. 3.20 part (D)

The best curves of three runs are compared in Figs. 3.25 ~ 3.29. Overall, the vertical difference between any two curves is relatively small (0.5 mm) compared to the scale of the mixing chamber wall. For the first run, the initial part of the optimized curve drops down below the original curve (shown in Fig. 3.6), which gives us a vertical difference of nearly 1 mm. Referring to the control points of these three curves, one can clearly see that the main difference between the first curve and the other two is the position of the second control point. For the first curve, this control point is way left compared to that in the other two curves, which may happen due to occurrence of local optimum. The curve shapes for run #2 and run #3 are fairly close to each other; however they still have some different in ER values.

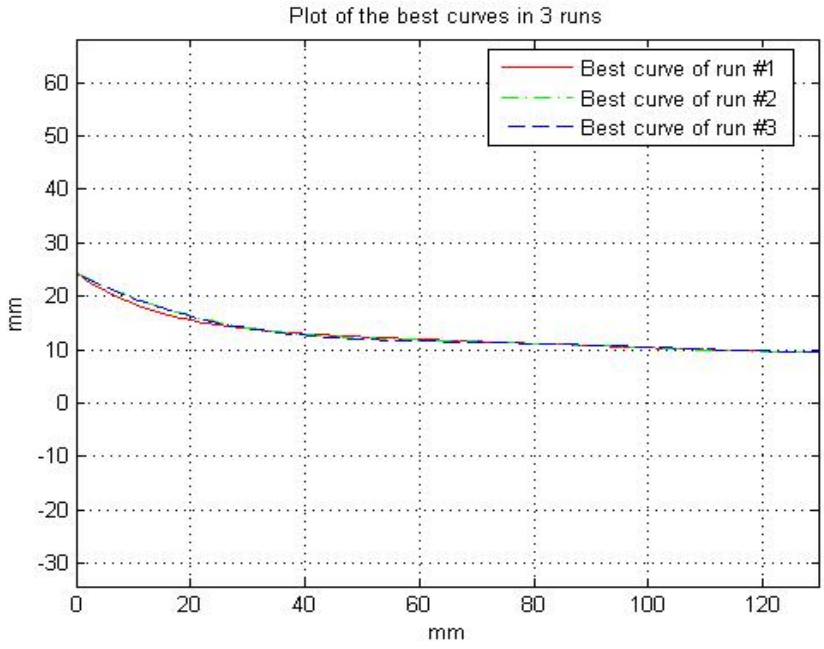

Figure 3.25: Plots of the best curves in three runs

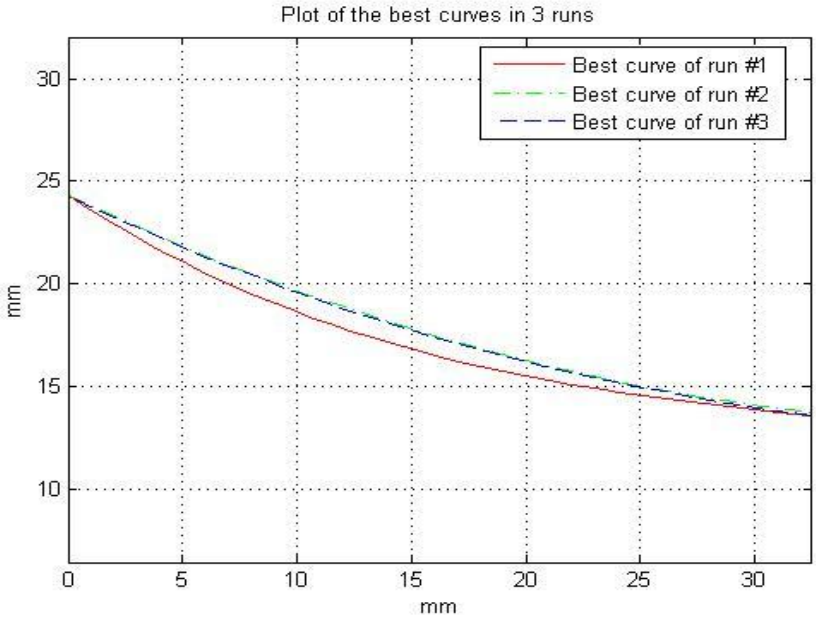

Figure 3.26: Zoomed-in view of Fig. 3.25 part (A)

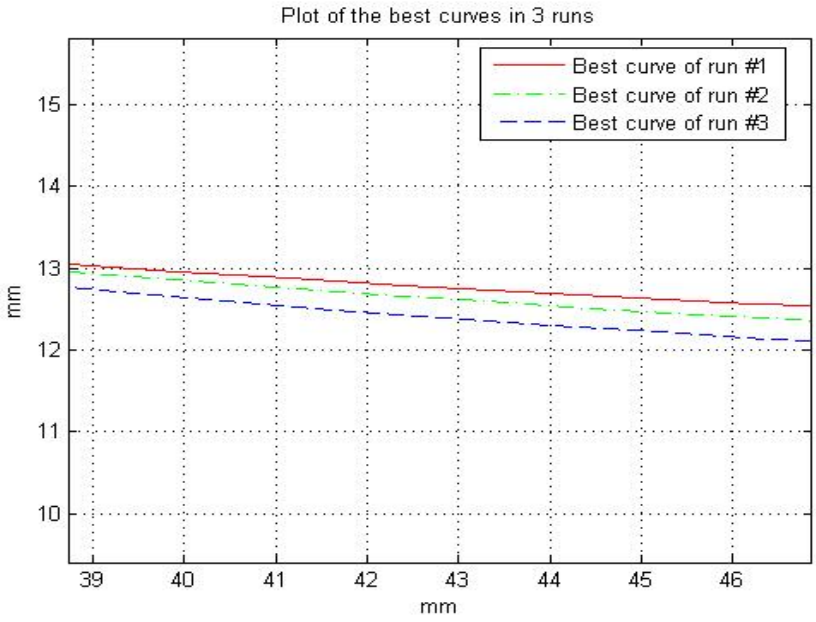

Figure 3.27: Zoomed-in view of Fig. 3.25 part (B)

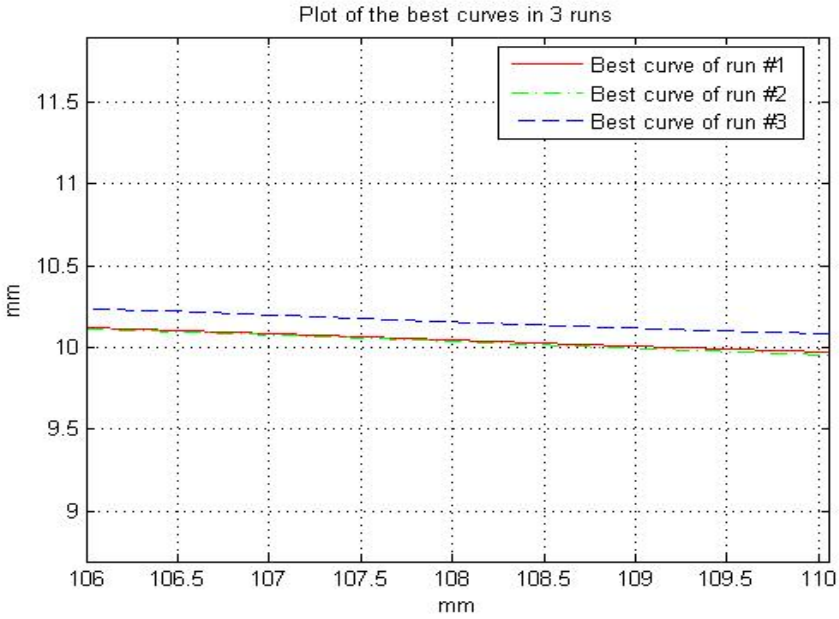

Figure 3.28: Zoomed-in view of Fig. 3.25 part (C)

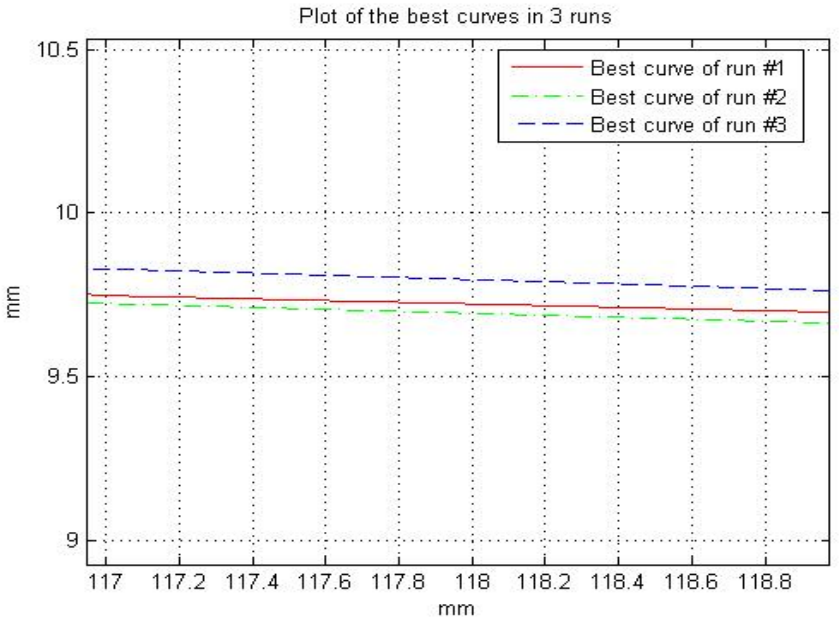

Figure 3.29: Zoomed-in view of Fig. 3.25 part (D)

Based on the ER values of the three runs, the best curve turns out to be that from run #3. The optimization results are summarized in Table 3.2. and Table 3.3.

Table 3.2: Optimization results for ER

	Run #1	**Run #2**	**Run #3**
Initial ER	0.3980	0.3980	0.3980
Final ER	0.40721	0.40723	0.40789
Improvement	2.314%	2.319%	2.485%

Table 3.3: (x, y) Coordinates of control points of optimized curves and original curve

Point #	**Original**	**Run #1**	**Run #2**	**Run #3**
1	(0, 24.31)	(0, 24.31)	(0, 24.31)	(0, 24.31)
2	(19.5, 9.49)	(21.215, 8.5941)	(34.268, 6.4805)	(34.5, 5.5894)
3	(56.16, 11.05)	(43.069, 14.806)	(48.245, 15.329)	(59.621, 14.704)
4	(99.58, 9.75)	(90.424, 9.833)	(93.872, 9.5431)	(88.41, 10.173)

5	(130, 9.49)	(130, 9.49)	(130, 9.49)	(130, 9.49)

Further optimization procedure may be conducted with generations containing more individuals or by increasing the mutation range. Applying a more strict convergence criteria may also reduce the possibility of reaching a local optimum.

Chapter 4 Conclusions

In the first part of this thesis, the commercial computational fluid dynamics (CFD) flow solver ANSYS-Fluent and the mesh generation software ANSYS-ICEM are employed to simulate the flow field of a supersonic steam ejector used in a refrigeration system using two different turbulence models: k-ε realizable and k-ω SST. By comparing the numerical results to the experimental data, k-ε realizable turbulence model is determined to be more accurate for predicting the free shear layer/mixing flows such as the flow field of a round jet. This model is then applied to three different supersonic ejectors (one used in refrigeration application with air as a working fluid and the other two used in desalination application with steam and air as working fluids) for further validation of the k-ε realizable model's accuracy. In the second part of the thesis, a MATLAB based single objective genetic algorithm (SOGA) is applied to optimize the mixing chamber wall shape of the first supersonic steam ejector used in refrigeration application to achieve a maximum entrainment ratio (ER). The optimal result of 2.485% improvement in ER is obtained after three optimization runs of single objective genetic algorithm (SOGA).

Appendix A

OptimizationMainProgram.m

```
% by Chris Seager

% Begin Code
% clear
% clc
warning('off','all')

% 1: Clearing previous data

 global RATIO_MATRIX
RATIO_MATRIX = zeros(20,2);
RATIO_MATRIX(1:20,1) = 1:20;


% 2: Input for times run/turbulence models to be used later
numAges=1;
age = 1;
while age <= numAges                                %number of total trials run
% % 3: Generate 20 Random Curves
%     generation = 1;
%     fid = fopen('Results.txt','w');
%        fprintf(fid,'Generation  %i\n',generation);   %printing  results  for
reference
%     fclose(fid);
%     fprintf('Generation %i\n',generation)
%   numTrial = 1;
%     while numTrial <= 20;                          %running random simulations
%         fprintf('Trial: %i\n',numTrial)
%         RandomBezier(numTrial)
%         pause(2);
%         !refrigeration_ICEM_replay.rpl
%         pause(12);
%         !ICEMkill.bat
%
%           if fluentSim(numTrial) == 1;   %import the data only if there is
data.
```

```
%             importfile('ratio');
%             ratio = data(1)/data(2);
%             RATIO_MATRIX(numTrial,2) = ratio;
%
%             fid = fopen('Results.txt','a');
%             fprintf(fid,'Ratio = %.5f\n\n',ratio);
%             fclose(fid);
%         end
%
%         pause(10);
%         deleteFunction
%         delete ratio
%
%         numTrial = numTrial + 1;                %reiterate
%
%     end

    RATIO_MATRIX = [1,0.064729
2,0.064729
3,0.37123
4,-Inf
5,0.39287
6,0.37596
7,0.2344
8,0.39346
9,0.012452
10,0.37276
11,-Inf
12,-Inf
13,0.40107
14,0.31452
15,0.16932
16,0.38137
17,0.23264
18,0.38423
19,0.37751
20,-0.51699
];
%
generation = 1;
numTrial = 21;

%      dlmwrite('ratio.txt',RATIO_MATRIX);
%4: Genetic Algorithm
    SOGA(numTrial,generation)

%5 Clean data for re-run
    deleteFunction
    generation = 2;
```

```
    numTrial = 11;                                  %parent trials are 1-10.
    fid = fopen('Results.txt','a');
    fprintf(fid,'Generation %i\n',generation);
    fclose(fid);

RATIO_MATRIX

%6 Simulating the daughter curves
    while convergenceCheck == 0
        fprintf('Generation %i\n',generation)
        while numTrial <= 20
            fprintf('Trial %i\n',numTrial)
            daughterCurves = dlmread('daughterCurves.txt');
            curve               =               [daughterCurves(2*(numTrial-10)-
1,:);daughterCurves(2*(numTrial-10),:)];
            GenerateBezier(curve,numTrial)
            pause(2);
            !refrigeration_ICEM_replay.rpl
            pause(12);
            !ICEMkill.bat

        if fluentSim(numTrial) == 1;  %import the data only if there is data.
            importfile('ratio');
            ratio = data(1)/data(2);
            RATIO_MATRIX(numTrial,2) = ratio;

            fid = fopen('Results.txt','a');
            fprintf(fid,'Ratio = %.5f\n\n',ratio);
            fclose(fid);
        end

        dlmwrite('ratio.txt',RATIO_MATRIX);

        pause(10);
        deleteFunction
        delete ratio


        numTrial = numTrial + 1;                %reiterate

        end

        SOGA(numTrial,generation)

        deleteFunction
        pause(3)
```

```
        delete ratio

        generation = generation + 1;
        numTrial = 11;

        fid = fopen('Results.txt','a');
        fprintf(fid,'Generation %i\n',generation);
        fclose(fid);
    end

% %     % getting best values
% %     BPResults = dlmread('BP.txt');
% %
% %     dlmwrite('optimizedCurves.txt',BPResults,'-append')
% %
% %     delete BP.txt

age = age + 1;
end



fprintf('\n\n---- End of simulation ----\n\n')
```

Appendix B

SOGA.m

```
function [] = SOGA(numTrial,generation)
%This is a single objective genetic algorithm maximizing the entrainment
%ratio of a supersonic steam ejector

%4a: load data
global RATIO_MATRIX


%4c: find the best half of all trials

trialData = RATIO_MATRIX;

sortedData = sortrows(trialData,-2);

bestRatio = sortedData(1,2);
average10ratio = sum(sortedData(1:10,2))/10;

selected = sortedData(1:10,:);
survivorNums = selected(:,1);

%finding curves of the survivors
curveData = dlmread('BP.txt');
[rows,cols] = size(curveData);

survivors = zeros(20,cols);

for i = 1:length(survivorNums)
    survivors(2*i-1,:) = curveData(2*survivorNums(i)-1,:);
    survivors(2*i,:) = curveData(2*survivorNums(i),:);
end



%convergence data
fid = fopen('convergenceData.txt','a');
fprintf(fid,'%i\t%5.6e\t%5.6e\n',generation,bestRatio,average10ratio);
```

```
fclose(fid);


%4d: generate daughter curves based on survivors
daughterCurveGen2(survivors,numTrial,generation)

% %4e: save survivors data to be compared with daughter curves
dlmwrite('BP.txt',survivors) %saves the Bezier Points for old gen.
parentData = sortedData(1:10,:);
newTrialNums = RATIO_MATRIX(1:10,1);
parentRatio = [newTrialNums, parentData(:,2)];
dlmwrite('ratio.txt',parentRatio,'delimiter','\t');

RATIO_MATRIX(:,2) = 0;
RATIO_MATRIX(1:10,:) = parentRatio;

end

%End of function.
```

Appendix C

Crossover.m

```
function [] = crossover(curve1,curve2)

[rows,cols] = size(curve1);

type = randi(3,1);

if type == 1 %exchange full bezier points
    numPoints = randi(2,1); %gives number of points to exchange (either 1 or
2)

    pointNums = uint8(randperm(3,numPoints))+1; %will select from points 2 to
4.
    pointNums = sort(pointNums); % ascending order

    %getting the point values
    c1s = zeros(rows,cols);
    c2s = c1s;
    for i = 1:numPoints
        c1s(:,pointNums(i)) = curve1(:,pointNums(i));
        c2s(:,pointNums(i)) = curve2(:,pointNums(i));

        curve1(:,pointNums(i)) = zeros(rows,1);
        curve2(:,pointNums(i)) = zeros(rows,1);
    end

    new1 = curve1 + c2s;
    new2 = curve2 + c1s;

elseif type == 2 %exchange only y values
    numPoints = randi(3,1); %gives number of points to exchange (1 to 3)

    pointNums = uint8(randperm(3,numPoints))+1;
    pointNums = sort(pointNums);

    y1s = zeros(rows,cols);
    y2s = y1s;
```

```
    for i = 1:numPoints
        y1s(2,pointNums(i)) = curve1(2,pointNums(i));
        y2s(2,pointNums(i)) = curve2(2,pointNums(i));

        curve1(2,pointNums(i)) = 0;
        curve2(2,pointNums(i)) = 0;
    end

    new1 = curve1 + y2s;
    new2 = curve2 + y1s;

else %average the entire curves like before
    numPoints = randi(3,1); %gives number of points to exchange (1 to 3)

    pointNums = uint8(randperm(3,numPoints))+1;
    pointNums = sort(pointNums);

    c1s = zeros(rows,cols);
    c2s = c1s;
    for i = 1:numPoints
        c1s(:,pointNums(i)) = curve1(:,pointNums(i));
        c2s(:,pointNums(i)) = curve2(:,pointNums(i));

        curve1(:,pointNums(i)) = zeros(rows,1);
        curve2(:,pointNums(i)) = zeros(rows,1);
    end

    cAverage = (c1s+c2s)/2;
    new1 = cAverage + curve1;
    new2 = cAverage + curve2;

end

%%%%%fprintf('done with initial crossover\n')

if rand(1) < .5
    dlmwrite('daughterCurves.txt',new1,'-append')
else
    dlmwrite('daughterCurves.txt',new2,'-append')
end


end
```

Appendix D

Mutation.m

```
function [daughterCurves] = mutator(daughterCurves,generation)

    global boxLimit;

    [rowsDC, cols] = size(daughterCurves);
    numCurves = rowsDC/2;

    numMutated = uint8(randi(ceil(numCurves))); %random selector

    curveNum = uint8(randperm(numCurves,numMutated));

    randomizer =15/generation^.4;

    i = 1;
    while i <= numMutated %iterates over all of the curves to be mutated
        curveMutated             =             [daughterCurves(curveNum(i)*2-
1,:);daughterCurves(curveNum(i)*2,:)];
            for m = 1:2 %iterate over all rows
                numPointstoMutate = randi(3);
                pointsBeingMutatedNums                                          =
uint8(1+randperm(3,numPointstoMutate));
                for  p  =  1:length(numPointstoMutate)  %iterate  over  all  the
selected points
                    n = pointsBeingMutatedNums(p);
                    if rand(1) < .5 %changes + or - value
                        curveMutated(m,n)     =     curveMutated(m,n)     +
rand(1)*randomizer;
                    else
                        curveMutated(m,n)     =     curveMutated(m,n)     -
rand(1)*randomizer;
                    end

                    %specific geometry constraints
                    %2nd control point
                    %x
                    if curveMutated(1,2) < 19.5-boxLimit
                        curveMutated(1,2) = 19.5-boxLimit;
```

```
                    elseif curveMutated(1,2) > 19.5+boxLimit
                        curveMutated(1,2) = 19.5+boxLimit;
                    end

                    %y
                    if curveMutated(2,2) < 9.49-boxLimit
                        curveMutated(2,2) = 9.49-boxLimit;
                    elseif curveMutated(2,2) > 9.49+boxLimit
                        curveMutated(2,2) = 9.49+boxLimit;
                    end

                    %3rd control point
                    %x
                    if curveMutated(1,3) < 56.16-boxLimit
                        curveMutated(1,3) = 56.16-boxLimit;
                    elseif curveMutated(1,3) > 56.16+boxLimit
                        curveMutated(1,3) = 56.16+boxLimit;
                    end

                     %y
                    if curveMutated(2,3) < 11.05-boxLimit
                        curveMutated(2,3) = 11.05-boxLimit;
                    elseif curveMutated(2,3) > 11.05+boxLimit
                        curveMutated(2,3) = 11.05+boxLimit;
                    end

                    %4th control point
                    %x
                    if curveMutated(1,4) < 99.58-boxLimit
                        curveMutated(1,4) = 99.58-boxLimit;
                    elseif curveMutated(1,4) > 99.58+boxLimit
                        curveMutated(1,4) = 99.58+boxLimit;
                    end

                     %y
                    if curveMutated(2,4) < 9.75-boxLimit
                        curveMutated(2,4) = 9.75-boxLimit;
                    elseif curveMutated(2,4) > 9.75+boxLimit
                        curveMutated(2,4) = 9.75+boxLimit;
                    end


                end
            end
        %curveMutated             =             [daughterCurves(curveNum(i)*2-
1,:);daughterCurves(curveNum(i)*2,:)]
        daughterCurves((curveNum(i)*2-1):curveNum(i)*2,:) = curveMutated;
```

```
        i = i+1;
    end


end
%end of function
```

Appendix E

RandomBezier.m

```
function [] = RandomBezier(numTrial)

        %counting parameters
        j=0;
        jj=0;

        %number of control points and the Bezier curve function # "n"
        PN=5;
        n=PN-1;

        global boxLimit
        boxLimit = 15;

        %original control points
        BPx = [0 1.5 4.32 7.66 10]*13;
        BPy = [1.87 .73 .85 .75 .73]*13;

        %defining empty vectors size 1x100 to generate original curve
        ux=zeros(1,100);
        uy=zeros(1,100);

        %Bezier function to generate points (see wikipedia)
        for t=0:1/(length(ux)-1):1
            j=j+1;
            for i=0:n
                if i==0
                    a=(1-t)^n;
                elseif i==n
                    a=t^n;
                else
                    a=(factorial(n)/(factorial(i)*factorial(n-i)))*t^(i)*(1-
t)^(n-i);
                end
                ux(j)=ux(j)+BPx(i+1)*a;
                uy(j)=uy(j)+BPy(i+1)*a;
            end
        end
```

```
        %now we have our original curve

        % now we will generate our random curves

        %makes random 1x3 vectors with values less than their limit
        randX = rand(1,3)*boxLimit;
        randY = rand(1,3)*boxLimit;

        %make empty vectors to fill in the loop below
        insertX = zeros(1,3);
        insertY = insertX;

        %randomly makes insert values the original BP values + or - a
        %number less than 15
        for nPoint = 1:3
            if rand(1)<.5
                insertX(nPoint) = BPx(nPoint+1)+randX(nPoint);
            else
                insertX(nPoint) = BPx(nPoint+1)-randX(nPoint);
            end
            if rand(1)<.5
                insertY(nPoint) = BPy(nPoint+1)+randY(nPoint);
            else
                insertY(nPoint) = BPy(nPoint+1)-randY(nPoint);
            end
        end

        %our new control points
        BPxRand = [0 insertX 130];
        BPyRand = [1.87*13 insertY .73*13];

        uxRand = zeros(size(ux));
        uyRand = uxRand;

        %gives us our new points
        for t=0:1/(length(uxRand)-1):1
            jj=jj+1;
            for i=0:n
                if i==0
                    a=(1-t)^n;
                elseif i==n
                    a=t^n;
                else
                    a=(factorial(n)/(factorial(i)*factorial(n-i)))*t^(i)*(1-
t)^(n-i);
                end
                uxRand(jj)=uxRand(jj)+BPxRand(i+1)*a;
                uyRand(jj)=uyRand(jj)+BPyRand(i+1)*a;
```

```
            end
        end



        %plotting

%     figure
%     axis equal
%     plot(ux,uy,'--r','LineWidth',2)
%     axis equal
%     hold on
%     plot(BPx,BPy,'*-g')
%     hold on
%     plot(uxRand,uyRand,'-b','LineWidth',2)
%     hold on
%     plot(BPxRand,BPyRand,'s-.c')

    points = [uxRand;uyRand;zeros(1,length(uxRand))];

    fid = fopen('Points.txt','w');
    fprintf(fid,'%5.0f %1.0f\n', length(ux), 1);
    fprintf(fid,'\t\t%f\t%f\t%f\n', points);
    fclose(fid);

    BP = [BPxRand;BPyRand];

    dlmwrite('BP.txt',BP,'-append')

    fid = fopen('Results.txt','a');
    fprintf(fid,'Curve %i\n',numTrial);
    dlmwrite('Results.txt',BP,'-append');
    fclose(fid);
end
```

Appendix F

ConvergenceCheck.m

```
function [converged] = convergenceCheck

convergenceData = dlmread('convergenceData.txt');

generation = convergenceData(:,1);
bestRatio = convergenceData(:,2);
avg10ratio = convergenceData(:,3);


plot(generation,bestRatio,generation,avg10ratio)
xlabel('Generation #'); ylabel('Entrainment ratio'); title('Best and average
ER plot');

if length(generation) <= 4;
    converged = 0;
elseif length(generation) > 4 && length(generation) < 41
    bestAvgDif = abs(bestRatio(end) - avg10ratio(end))/bestRatio(end);
    bestDif = abs(bestRatio(end) - bestRatio(end-3));
    avgDif = abs(avg10ratio(end) - avg10ratio(end-3))/avg10ratio(end);

    if (bestAvgDif < .01) && (bestDif < .000000001) && (avgDif < .01)
        converged = 1;
        fprintf('\nConvergence       criteria       reached       at       %i
generations\n',length(generation))
    else
        converged = 0;
    end
else
    converged = 1;
    fprintf('\nMax Generations Reached: Generation %i\n',length(generation))
end

end
%end of function
```

Appendix G

ICEM_replay.rpl

```
ic_undo_group_begin
ic_geo_cre_geom_input {C:/Users/su/Desktop/points for the rest of the
geometry.txt} 0.001 input PNTS pnt CRVS crv SURFS srf
ic_boco_solver
ic_boco_clear_icons
ic_csystem_display all 0
ic_csystem_set_current global
ic_boco_nastran_csystem reset
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_geo_new_family GEOM
ic_boco_set_part_color GEOM
ic_delete_geometry curve names crv.00 0
ic_curve point PNTS crv.00 {pnt0 pnt1}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.01 0
ic_curve point PNTS crv.01 {pnt1 pnt2}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.02 0
ic_curve point PNTS crv.02 {pnt2 pnt3}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.03 0
ic_curve point PNTS crv.03 {pnt3 pnt4}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
```

```
ic_undo_group_begin
ic_delete_geometry curve names crv.04 0
ic_curve point PNTS crv.04 {pnt5 pnt4}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.05 0
ic_curve point PNTS crv.05 {pnt5 pnt6}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.06 0
ic_curve point PNTS crv.06 {pnt6 pnt7}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_curve arc PNTS crv.07 {pnt7 pnt8 pnt9}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.08 0
ic_curve point PNTS crv.08 {pnt9 pnt10}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.09 0
ic_curve point PNTS crv.09 {pnt10 pnt11}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.10 0
ic_curve point PNTS crv.10 {pnt11 pnt12}
ic_undo_group_end
ic_undo_group_begin
```

```
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.11 0
ic_curve point PNTS crv.11 {pnt12 pnt13}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_geo_cre_geom_input C:/Users/su/Desktop/OptimizationProgram/Points.txt
0.001 input PNTS {} CRVS crv SURFS {}
ic_boco_solver
ic_boco_clear_icons
ic_csystem_display all 0
ic_csystem_set_current global
ic_boco_nastran_csystem reset
ic_undo_group_end
ic_undo_group_begin
ic_geo_new_family FLUID
ic_boco_set_part_color FLUID
ic_hex_initialize_mesh 2d new_numbering new_blocking FLUID
ic_hex_switch_blocking root
ic_hex_unblank_blocks
ic_hex_multi_grid_level 0
ic_hex_projection_limit 0
ic_hex_default_bunching_law default 2.0
ic_hex_floating_grid off
ic_hex_transfinite_degree 1
ic_hex_unstruct_face_type one_tri
ic_hex_set_unstruct_face_method uniform_quad
ic_hex_set_n_tetra_smoothing_steps 20
ic_hex_set_mesh_params PNTS CRVS SURFS GEOM FLUID -version 110
ic_hex_error_messages off_minor
ic_hex_switch_blocking root
ic_undo_group_end
ic_undo_group_begin
ic_hex_split_grid 11 19 0.0539487 m PNTS CRVS SURFS GEOM FLUID VORFN
ic_undo_group_end
ic_undo_group_begin
ic_hex_split_grid 33 19 0.0126005 m PNTS CRVS SURFS GEOM FLUID VORFN
ic_undo_group_end
ic_undo_group_begin
ic_hex_split_grid 37 19 0.112016 m PNTS CRVS SURFS GEOM FLUID VORFN
ic_undo_group_end
ic_undo_group_begin
ic_hex_split_grid 41 19 0.510972 m PNTS CRVS SURFS GEOM FLUID VORFN
ic_undo_group_end
ic_undo_group_begin
ic_hex_split_grid 41 42 0.741064 m PNTS CRVS SURFS GEOM FLUID VORFN
ic_undo_group_end
ic_undo_group_begin
```

```
ic_undo_group_end
ic_undo
ic_undo
ic_undo_group_begin
ic_hex_split_grid 41 42 0.220275 m PNTS CRVS SURFS GEOM FLUID VORFN
ic_undo_group_end
ic_undo_group_begin
ic_hex_split_grid 41 52 0.657679 m PNTS CRVS SURFS GEOM FLUID VORFN
ic_undo_group_end
ic_undo_group_begin
ic_hex_mark_blocks unmark
ic_hex_mark_blocks superblock 22
ic_hex_mark_blocks superblock 23
ic_hex_mark_blocks superblock 29
ic_hex_mark_blocks superblock 30
ic_hex_mark_blocks superblock 31
ic_hex_change_element_id VORFN
ic_geo_delete_family SURFS
ic_undo_group_end
ic_undo_group_begin
ic_hex_create_composite {crv0 crv.00 crv.01}
ic_hex_set_edge_projection 38 42 0 1 crv0
ic_hex_set_edge_projection 42 46 0 1 crv0
ic_hex_set_edge_projection 46 21 0 1 crv0
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 54 21 0 1 crv.02
ic_hex_set_edge_projection 62 54 0 1 crv.02
ic_hex_set_edge_projection 19 62 0 1 crv.02
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 45 19 0 1 crv.03
ic_hex_set_edge_projection 41 45 0 1 crv.03
ic_hex_set_edge_projection 37 41 0 1 crv.03
ic_hex_set_edge_projection 33 37 0 1 crv.03
ic_hex_set_edge_projection 11 33 0 1 crv.03
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 11 57 0 1 crv.04
ic_undo_group_end
ic_undo_group_begin
ic_hex_create_composite {crv.05 crv.06 crv.07 crv.08}
ic_hex_set_edge_projection 57 58 0 1 crv.05
ic_hex_set_edge_projection 58 59 0 1 crv.05
ic_hex_set_edge_projection 59 60 0 1 crv.05
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 60 52 0 1 crv.09
ic_undo_group_end
ic_undo_group_begin
```

```
ic_hex_set_edge_projection 51 52 0 1 crv.10
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 51 38 0 1 crv.11
ic_undo_group_end
ic_hex_place_node 38 curve:crv0 3.42568835e-009
ic_hex_place_node 51 curve:crv.10 1
ic_hex_place_node 51 curve:crv.10 1
ic_hex_place_node 57 curve:crv.04 0
ic_hex_place_node 58 curve:crv.05 0.999999993
ic_hex_place_node 33 curve:crv.03 0.942095757
ic_hex_place_node 59 curve:crv.07 0.99612926
ic_hex_place_node 37 curve:crv.03 0.936617319
ic_hex_place_node 52 curve:crv.09 1
ic_hex_place_node 60 curve:crv.08 1
ic_hex_place_node 41 curve:crv.03 0.833407651
ic_hex_place_node 42 curve:crv0 0.282210125
ic_hex_place_node 41 curve:crv.03 0.83334452
ic_hex_place_node 46 curve:crv.01 0.00576672033
ic_hex_place_node 46 curve:crv.00 0.999287013
ic_hex_place_node 21 curve:crv.01 0.998334758
ic_hex_place_node 21 curve:crv.01 1
ic_undo_group_begin
ic_set_meshing_params curve {crv.00 crv.01 crv.02 crv.03 crv.04 crv.05 crv.06
crv.07 crv.08 crv.09 crv.10 crv.11 crv0} emax 2 emin 0 ehgt 0 edev 0 hrat 0
ewid 0 nlay 0
ic_undo_group_end
ic_undo_group_begin
ic_hex_renew
ic_hex_set_mesh_params PNTS CRVS GEOM FLUID -version 110
ic_hex_compute_mesh_size PNTS CRVS GEOM FLUID
ic_undo_group_end
ic_hex_list_family_projection
ic_hex_create_mesh PNTS CRVS FLUID proj 2 dim_to_mesh 3
ic_undo_group_begin
ic_hex_set_mesh 52 42 n 30 h1 0.700019 h2 0 r1 2 r2 2 lmax 0 default unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 52 42 n 30 h1rel 0.0749774002828 h2rel 0.00107107664624 r1 2
r2 2 lmax 0 default copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 52 42 n 30 h1rel 0.0749774002828 h2rel 0.00107107664624 r1 2
r2 2 lmax 0 default copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
```

```
ic_hex_set_mesh 60 52 n 15 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 60 52 n 15 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 60 52 n 15 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 41 60 n 30 h1rel 0.0 h2rel 0.1750005 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 41 60 n 30 h1rel 0.0 h2rel 0.17500075 r1 2 r2 2 lmax 0
default copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 41 60 n 30 h1rel 0.0 h2rel 0.175001 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 41 60 n 30 h1rel 0.0 h2rel 0.17500125 r1 2 r2 2 lmax 0
default copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 11 33 n 50 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 11 33 n 50 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 11 33 n 50 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
```

```
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 33 37 n 15 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 33 37 n 15 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 33 37 n 15 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 37 41 n 100 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 37 41 n 100 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 37 41 n 100 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 41 45 n 500 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 41 45 n 500 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 41 45 n 500 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
```

```
ic_hex_set_mesh 45 19 n 150 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 45 19 n 150 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 45 19 n 150 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 2 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_hex_list_family_projection
ic_hex_create_mesh PNTS CRVS FLUID proj 2 dim_to_mesh 3
ic_undo_group_begin
ic_hex_match_edges 60 52 52 42
ic_undo_group_end
ic_undo_group_begin
ic_hex_match_edges 60 52 41 60
ic_undo_group_end
ic_hex_list_family_projection
ic_hex_create_mesh PNTS CRVS FLUID proj 2 dim_to_mesh 3
ic_undo_group_begin
ic_geo_set_part curve crv.04 NOZZLE_INLET 0
ic_undo_group_end
ic_undo_group_begin
ic_geo_set_part curve crv.11 EJECTOR_INLET 0
ic_undo_group_end
ic_undo_group_begin
ic_geo_set_part curve {crv0 crv.00 crv.01} EJECTOR_WALL 0
ic_undo_group_end
ic_undo_group_begin
ic_geo_set_part curve crv.02 EJECTOR_OUTLET 0
ic_undo_group_end
ic_undo_group_begin
ic_geo_set_part curve crv.03 AXIS 0
ic_undo_group_end
ic_undo_group_begin
ic_geo_set_part curve {crv.05 crv.06 crv.07 crv.08 crv.09 crv.10} NOZZLE_WALL
0
ic_undo_group_end
ic_undo_group_begin
ic_boco_solver Fluent_V6
ic_solver_mesh_info Fluent_V6
ic_undo_group_end
ic_geo_new_family VORFN 0
ic_boco_set VORFN {}
```

```
ic_boco_set EJECTOR_WALL {{1 WALL 0}}
ic_boco_set PNTS {}
ic_boco_set AXIS {{1 AXIS 0}}
ic_boco_set CRVS {}
ic_boco_set EJECTOR_INLET {{1 PRESI 0}}
ic_boco_set GEOM { { 1  {color}  16663866  } }
ic_boco_set EJECTOR_OUTLET {{1 PRESO 0}}
ic_boco_set NOZZLE_WALL {{1 WALL 0}}
ic_boco_set FLUID { { 1  {color}  12109107  } }
ic_boco_set ORFN {}
ic_boco_set NOZZLE_INLET {{1 PRESI 0}}
ic_hex_write_file hex.uns PNTS CRVS GEOM FLUID NOZZLE_INLET EJECTOR_INLET
EJECTOR_WALL EJECTOR_OUTLET AXIS NOZZLE_WALL proj 2 dim_to_mesh 3 -
family_boco family_boco.fbc
ic_uns_load hex.uns 3 0 {} 2
ic_uns_update_family_type visible {VORFN EJECTOR_WALL PNTS AXIS CRVS
EJECTOR_INLET GEOM EJECTOR_OUTLET NOZZLE_WALL FLUID ORFN NOZZLE_INLET}
{!LINE_2 QUAD_4} update 0
ic_uns_diag_reset_degen_min_max
ic_boco_solver
ic_uns_update_family_type visible {VORFN EJECTOR_WALL PNTS AXIS CRVS
EJECTOR_INLET GEOM EJECTOR_OUTLET NOZZLE_WALL FLUID ORFN NOZZLE_INLET}
{!LINE_2 QUAD_4} update 0
ic_hex_list_family_projection
ic_hex_create_mesh PNTS FLUID NOZZLE_INLET EJECTOR_INLET EJECTOR_WALL
EJECTOR_OUTLET AXIS NOZZLE_WALL VORFN VORFN VORFN proj 2 dim_to_mesh 3
ic_boco_clear_icons
ic_csystem_display all 0
ic_csystem_set_current global
ic_boco_nastran_csystem reset
ic_boco_solver
ic_boco_save C:/Users/su/Desktop/OptimizationProgram/nastran.fbc
ic_boco_save_atr C:/Users/su/Desktop/OptimizationProgram/nastran.atr
ic_chdir C:/Users/su/Desktop/OptimizationProgram
ic_geo_delete_family CRVS
ic_save_tetin project1.tin 0 0 {} {} 0 0 1
ic_rename project1.uns project1.uns.bak
ic_uns_check_duplicate_numbers
ic_save_unstruct project1.uns 1 {} {} {}
ic_uns_set_modified 1
ic_hex_save_blocking project1.blk
ic_boco_solver
ic_boco_save project1.fbc
ic_boco_save_atr project1.atr
ic_cart_is_loaded
ic_save_project_file C:/Users/su/Desktop/OptimizationProgram/project1.prj
{array\ set\ file_name\ \{ {    catia_dir .} {    parts_dir .} {
domain_loaded 1} {    cart_file_loaded 0} {    cart_file {}} {
domain_saved project1.uns} {    archive {}} {    med_replay {}} {
topology_dir .} {    ugparts_dir .} {    icons
```

{{$env(ICEM_ACN)/lib/ai_env/icons} {$env(ICEM_ACN)/lib/va/EZCAD/icons}
{$env(ICEM_ACN)/lib/icons} {$env(ICEM_ACN)/lib/va/CABIN/icons}}} {   tetin
project1.tin} {   family_boco project1.fbc} {   iges_dir .} {
solver_params_loaded 0} {   attributes_loaded 0} {   project_lock {}} {
attributes project1.atr} {   domain project1.uns} {   domains_dir .} {
settings_loaded 0} {   settings project1.prj} {   blocking project1.blk} {
hexa_replay {}} {   transfer_dir .} {   mesh_dir .} {   family_topo {}} {
gemsparts_dir .} {   family_boco_loaded 0} {   tetin_loaded 0} {
project_dir .} {   topo_mulcad_out {}} {   solver_params {}} \} array\ set\
options\ \{ {   expert 1} {   remote_path {}} {   tree_disp_quad 2} {
tree_disp_pyra 0} {   evaluate_diagnostic 0} {   histo_show_default 1} {
select_toggle_corners 0} {   remove_all 0} {   keep_existing_file_names 0}
{   record_journal 0} {   edit_wait 0} {   face_mode all} {   select_mode
all} {   med_save_emergency_tetin 1} {   user_name su} {   diag_which all}
{   uns_warn_if_display 500000} {   bubble_delay 1000} {   external_num 1}
{   tree_disp_tri 2} {   apply_all 0} {   default_solver Fluent_V6} {
temporary_directory {}} {   flood_select_angle 0} {   home_after_load 1} {
project_active 0} {   histo_color_by_quality_default 1} {   undo_logging 1}
{   tree_disp_hexa 0} {   histo_solid_default 1} {   host_name CFDlabPC8}
{   xhidden_full 1} {   editor {}} {   mouse_color orange} {   clear_undo
1} {   remote_acn {}} {   remote_sh csh} {   tree_disp_penta 0} {
n_processors 1} {   remote_host {}} {   save_to_new 0} {   quality_info
Quality} {   tree_disp_node 0} {   med_save_emergency_mesh 1} {
redtext_color red} {   tree_disp_line 0} {   select_edge_mode 0} {
use_dlremote 0} {   max_mesh_map_size 1024} {   show_tris 1} {
remote_user {}} {   auto_save_views 1} {   max_cad_map_size 512} {
display_origin 0} {   uns_warn_user_if_display 1000000} {   detail_info 0}
{   win_java_help 0} {   show_factor 1} {   boundary_mode all} {
clean_up_tmp_files 1} {   med_save_emergency_blocking 1} {
max_binary_tetin 0} {   tree_disp_tetra 0} \} array\ set\ disp_options\ \{ {
uns_dualmesh 0} {   uns_warn_if_display 500000} {   uns_normals_colored 0}
{   uns_icons 0} {   uns_locked_elements 0} {   uns_shrink_npos 0} {
uns_node_type None} {   uns_icons_normals_vol 0} {   uns_bcfield 0} {
backup Wire} {   uns_nodes 0} {   uns_only_edges 0} {   uns_surf_bounds 0}
{   uns_wide_lines 0} {   uns_vol_bounds 0} {   uns_displ_orient Triad} {
uns_orientation 0} {   uns_directions 0} {   uns_thickness 0} {
uns_shell_diagnostic 0} {   uns_normals 0} {   uns_couplings 0} {
uns_periodicity 0} {   uns_single_surfaces 0} {   uns_midside_nodes 1} {
uns_shrink 100} {   uns_multiple_surfaces 0} {   uns_no_inner 0} {
uns_enums 0} {   uns_disp Wire} {   uns_bcfield_name {}} {
uns_color_by_quality 0} {   uns_changes 0} {   uns_cut_delay_count 1000} \}
{set icon_size1 24} {set icon_size2 35} {set thickness_defined 0} {set
solver_type 1} {set solver_setup 1} array\ set\ prism_values\ \{ {
n_triangle_smoothing_steps 5} {   min_smoothing_steps 6} {
first_layer_smoothing_steps 1} {   new_volume {}} {   height {}} {
prism_height_limit {}} {   interpolate_heights 0} {
n_tetra_smoothing_steps 10} {   do_checks {}} {   delete_standalone 1} {
ortho_weight 0.50} {   max_aspect_ratio {}} {   ratio_max {}} {
total_height {}} {   use_prism_v10 0} {   intermediate_write 1} {
delete_base_triangles {}} {   ratio_multiplier {}} {

```
refine_prism_boundary 1} {    max_size_ratio {}} {    triangle_quality {}} {
max_prism_angle 180} {    tetra_smooth_limit 0.3} {    max_jump_factor 5} {
use_existing_quad_layers 0} {    layers 3} {    fillet 0.10} {    into_orphan
0} {    init_dir_from_prev {}} {    blayer_2d 0} {    do_not_allow_sticking
{}} {    top_family {}} {    law exponential} {    min_smoothing_val 0.1} {
auto_reduction 0} {    stop_columns 1} {    stair_step 1} {
smoothing_steps 12} {    side_family {}} {    min_prism_quality 0.01} {
ratio 1.2} \} {set aie_current_flavor {}} array\ set\ vid_options\ \{ {
wb_NS_to_subset 0} {    auxiliary 0} {    show_name 0} {    inherit 1} {
default_part GEOM} {    new_srf_topo 1} {    DelPerFlag 0} {
show_item_name 0} {    composite_tolerance 1.0} {    wb_import_scale_geo 0} {
replace 0} {    same_pnt_tol 1e-4} {    tdv_axes 1} {    vid_mode 0} {
DelBlkPerFlag 0} \} {set savedTreeVisibility {geomNode 1 geom_subsetNode 2
geomPointNode 0 geomCurveNode 2 meshNode 1 mesh_subsetNode 2 meshLineNode 0
meshShellNode 2 meshQuadNode 2 blockingNode 1 block_subsetNode 2
block_vertNode 0 block_edgeNode 0 block_faceNode 0 block_blockNode 0
block_meshNode 0 topoNode 2 topo-root 2 partNode 2 part-AXIS 2 part-
EJECTOR_INLET 2 part-EJECTOR_OUTLET 2 part-EJECTOR_WALL 2 part-FLUID 2 part-
NOZZLE_INLET 2 part-NOZZLE_WALL 2 part-PNTS 2 part-VORFN 0}} {set last_view
{rot {0 0 0 1} scale {10.7560759087 10.7560759087 10.7560759087} center
{183.0 12.155 0.0} pos {1943.43758722 59.5658226741 0}}} array\ set\
cut_info\ \{ {    active 0} \} array\ set\ hex_option\ \{ {
default_bunching_ratio 2.0} {    floating_grid 0} {
n_tetra_smoothing_steps 20} {    trfDeg 1} {    wr_hexa7 0} {    smooth_ogrid
0} {    find_worst 1-3} {    hexa_verbose_mode 0} {    old_eparams 0} {
uns_face_mesh_method uniform_quad} {    multigrid_level 0} {    uns_face_mesh
one_tri} {    check_blck 0} {    proj_limit 0} {    check_inv 0} {
project_bspline 0} {    hexa_update_mode 1} {    default_bunching_law
BiGeometric} \} array\ set\ saved_views\ \{ {    views {}} \}} {ICEM CFD}
ic_exec {C:/Program Files/ANSYS Inc/v145/icemcfd/win64_amd/icemcfd/output-
interfaces/fluent6} -dom
{C:/Users/su/Desktop/OptimizationProgram/project1.uns} -b project1.fbc -dim2d
./fluent
ic_uns_num_couplings
ic_undo_group_begin
ic_uns_create_diagnostic_edgelist 1
ic_uns_diagnostic subset all diag_type uncovered fix_fam FIX_UNCOVERED
diag_verb {Uncovered faces} fams {} busy_off 1 quiet 1
ic_uns_create_diagnostic_edgelist 0
ic_undo_group_end
```

Appendix H

FLUENT_journal.Jou

```
(cx-gui-do cx-activate-item "MenuBar*ReadSubMenu*Mesh...")
(cx-gui-do cx-set-text-entry "Select File*FilterText"
"c:\users\su\desktop\optimizationprogram\*")
(cx-gui-do cx-activate-item "Select File*Apply")
(cx-gui-do cx-set-text-entry "Select File*Text" "fluent.msh")
(cx-gui-do cx-activate-item "Select File*OK")
(cx-gui-do cx-set-list-selections "GraphicsArea*GraphicsView1*DropDownList1"
'( 0))
(cx-gui-do cx-activate-item "GraphicsArea*GraphicsView1*DropDownList1")
(cx-gui-do cx-activate-item "MenuBar*DefineMenu*Operating Conditions...")
(cx-gui-do cx-set-real-entry-list "Operating
Conditions*Frame1*Frame1(Pressure)*Table1(Pressure)*RealEntry2(Operating
Pressure)" '( 0))
(cx-gui-do cx-activate-item "Operating
Conditions*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item
"General*Frame1*Table1*Frame1(Mesh)*ButtonBox1(Mesh)*PushButton1(Scale)")
(cx-gui-do cx-set-list-selections "Scale
Mesh*Frame2(Scaling)*Table2(Scaling)*DropDownList2(Mesh Was Created In)" '(
3))
(cx-gui-do cx-activate-item "Scale
Mesh*Frame2(Scaling)*Table2(Scaling)*DropDownList2(Mesh Was Created In)")
(cx-gui-do cx-activate-item "Scale
Mesh*Frame2(Scaling)*Table2(Scaling)*PushButton4(Scale)")
(cx-gui-do cx-set-list-selections "Scale Mesh*DropDownList3(View Length Unit
In)" '( 2))
(cx-gui-do cx-activate-item "Scale Mesh*DropDownList3(View Length Unit In)")
(cx-gui-do cx-activate-item "Scale Mesh*PanelButtons*PushButton1(Close)")
(cx-gui-do cx-set-toggle-button
"General*Frame1*Table1*Frame2(Solver)*Table2(Solver)*ButtonBox1(Type)*Density
-Based" #f)
(cx-gui-do cx-activate-item
"General*Frame1*Table1*Frame2(Solver)*Table2(Solver)*ButtonBox1(Type)*Density
-Based")
(cx-gui-do cx-set-toggle-button
"General*Frame1*Table1*Frame2(Solver)*Table2(Solver)*ToggleBox6(2D
Space)*Axisymmetric" #f)
```

```
(cx-gui-do cx-activate-item
"General*Frame1*Table1*Frame2(Solver)*Table2(Solver)*ToggleBox6(2D
Space)*Axisymmetric")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton5(Models)")
(cx-gui-do cx-set-list-selections "Models*Frame1*Table1*Frame1*List1(Models)"
'( 2))
(cx-gui-do cx-activate-item "Models*Frame1*Table1*Frame1*List1(Models)")
(cx-gui-do cx-activate-item "Models*Frame1*Table1*PushButton2(Edit)")
(cx-gui-do cx-set-toggle-button "Viscous
Model*Frame1*Table1*Frame1(Model)*ToggleBox1(Model)*k-epsilon (2 eqn)" #f)
(cx-gui-do cx-activate-item "Viscous
Model*Frame1*Table1*Frame1(Model)*ToggleBox1(Model)*k-epsilon (2 eqn)")
(cx-gui-do cx-set-toggle-button "Viscous Model*Frame1*Table1*Frame6(k-epsilon
Model)*ToggleBox6(k-epsilon Model)*Realizable" #f)
(cx-gui-do cx-activate-item "Viscous Model*Frame1*Table1*Frame6(k-epsilon
Model)*ToggleBox6(k-epsilon Model)*Realizable")
(cx-gui-do cx-activate-item "Viscous Model*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton6(Materials)")
(cx-gui-do cx-set-list-selections
"Materials*Frame1*Table1*Frame1*List1(Materials)" '( 1))
(cx-gui-do cx-activate-item
"Materials*Frame1*Table1*Frame1*List1(Materials)")
(cx-gui-do cx-activate-item
"Materials*Frame1*Table1*ButtonBox2*PushButton1(Create/Edit)")
(cx-gui-do cx-activate-item "Create/Edit
Materials*Frame1*Table1*Frame1*ButtonBox3*PushButton1(Fluent Database)")
(cx-gui-do cx-set-list-selections "Database
Materials*Frame1*Table1*Frame1*Frame1*List1(Materials)" '( 560))
(cx-gui-do cx-activate-item "Database
Materials*Frame1*Table1*Frame1*Frame1*List1(Materials)")
(cx-gui-do cx-activate-item "Database
Materials*PanelButtons*PushButton1(Copy)")
(cx-gui-do cx-activate-item "Database
Materials*PanelButtons*PushButton1(Close)")
(cx-gui-do cx-set-list-selections "Create/Edit
Materials*Frame2(Properties)*Table2(Properties)*Frame4*Frame2*DropDownList1"
'( 1))
(cx-gui-do cx-activate-item "Create/Edit
Materials*Frame2(Properties)*Table2(Properties)*Frame4*Frame2*DropDownList1")
(cx-gui-do cx-activate-item "Create/Edit
Materials*PanelButtons*PushButton1(Change/Create)")
(cx-gui-do cx-activate-item "Create/Edit
Materials*PanelButtons*PushButton1(Change/Create)")
(cx-gui-do cx-activate-item "Create/Edit
Materials*PanelButtons*PushButton1(Close)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton8(Cell Zone
Conditions)")
(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame2*Table2*Frame4*Table4*ButtonBox1*PushButton1(E
dit)")
```

```
(cx-gui-do cx-set-list-selections "fluid-16-
1*Frame3*Table3*Frame1*Table1*DropDownList1(Material Name)" '( 0))
(cx-gui-do cx-activate-item "fluid-16-
1*Frame3*Table3*Frame1*Table1*DropDownList1(Material Name)")
(cx-gui-do cx-activate-item "fluid-16-1*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton9(Boundary
Conditions)")
(cx-gui-do cx-set-list-selections "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 1))
(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)")
(cx-gui-do cx-set-list-selections "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 0))
(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)")
(cx-gui-do cx-set-list-selections "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 1))
(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)")
(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame2*Table2*Frame4*Table4*ButtonBox1*PushButton1(E
dit)")
(cx-gui-do cx-set-real-entry-list "pressure-inlet-19-
1*Frame4*Frame1(Momentum)*Frame1*Table1*Frame5*Table5*RealEntry2(Gauge Total
Pressure)" '( 1230))
(cx-gui-do cx-set-real-entry-list "pressure-inlet-19-
1*Frame4*Frame3(Thermal)*Frame1*Table1*Frame1*Table1*RealEntry2(Total
Temperature)" '( 283.15))
(cx-gui-do cx-activate-item "pressure-inlet-19-
1*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-list-selections "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 2))
(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)")
(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame2*Table2*Frame4*Table4*ButtonBox1*PushButton1(E
dit)")
(cx-gui-do cx-set-real-entry-list "pressure-outlet-21-
1*Frame4*Frame1(Momentum)*Frame1*Table1*Frame4*Table4*RealEntry2(Gauge
Pressure)" '( 3000))
(cx-gui-do cx-set-real-entry-list "pressure-outlet-21-
1*Frame4*Frame3(Thermal)*Frame1*Table1*Frame1*Table1*RealEntry2(Backflow
Total Temperature)" '( 297.23))
(cx-gui-do cx-activate-item "pressure-outlet-21-
1*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-list-selections "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 3))
(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)")
```

```
(cx-gui-do cx-set-list-selections "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 4))
(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)")
(cx-gui-do cx-set-list-selections "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 6))
(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)")
(cx-gui-do cx-set-list-selections "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 5))
(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame1*List1(Zone)")
(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame2*Table2*Frame4*Table4*ButtonBox1*PushButton1(E
dit)")
(cx-gui-do cx-set-real-entry-list "pressure-inlet-18-
1*Frame4*Frame1(Momentum)*Frame1*Table1*Frame5*Table5*RealEntry2(Gauge Total
Pressure)" '( 270280))
(cx-gui-do cx-set-real-entry-list "pressure-inlet-18-
1*Frame4*Frame3(Thermal)*Frame1*Table1*Frame1*Table1*RealEntry2(Total
Temperature)" '( 403.15))
(cx-gui-do cx-activate-item "pressure-inlet-18-
1*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton11(Dynamic
Mesh)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton12(Reference
Values)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton14(Solution
Methods)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton15(Solution
Controls)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton16(Monitors)")
(cx-gui-do cx-set-list-selections
"Monitors*Frame1*Table1*Frame1*List1(Residuals, Statistic and Force
Monitors)" '( 0))
(cx-gui-do cx-activate-item "Monitors*Frame1*Table1*Frame1*List1(Residuals,
Statistic and Force Monitors)")
(cx-gui-do cx-activate-item
"Monitors*Frame1*Table1*Frame2*Table2*PushButton2(Edit)")
(cx-gui-do cx-set-real-entry-list "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame1(Equations)*Table1(Equations)*Real
Entry11" '( 1e-006))
(cx-gui-do cx-set-real-entry-list "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame1(Equations)*Table1(Equations)*Real
Entry17" '( 1e-006))
(cx-gui-do cx-set-real-entry-list "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame1(Equations)*Table1(Equations)*Real
Entry23" '( 1e-006))
```

```
(cx-gui-do cx-set-real-entry-list "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame1(Equations)*Table1(Equations)*Real
Entry29" '( 1e-006))
(cx-gui-do cx-set-real-entry-list "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame1(Equations)*Table1(Equations)*Real
Entry35" '( 1e-006))
(cx-gui-do cx-set-real-entry-list "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame1(Equations)*Table1(Equations)*Real
Entry41" '( 1e-006))
(cx-gui-do cx-activate-item "Residual Monitors*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item
"Monitors*Frame1*Table1*Frame2*Table2*PushButton1(Create)")
(cx-gui-do cx-activate-item "MenuBar*PopupMenuCreateMonitor*Drag...")
(cx-gui-do cx-set-list-selections "Force Moment
Monitor*Frame1*Table1*Frame2*Table2*Frame1*List1(Wall Zones)" '( 0))
(cx-gui-do cx-activate-item "Force Moment
Monitor*Frame1*Table1*Frame2*Table2*Frame1*List1(Wall Zones)")
(cx-gui-do cx-set-toggle-button "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckButt
on1(Print to Console)" #f)
(cx-gui-do cx-activate-item "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckButt
on1(Print to Console)")
(cx-gui-do cx-set-toggle-button "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckButt
on2(Plot)" #f)
(cx-gui-do cx-activate-item "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckButt
on2(Plot)")
(cx-gui-do cx-activate-item "Force Moment
Monitor*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item
"Monitors*Frame1*Table1*Frame4*Table4*PushButton1(Create)")
(cx-gui-do cx-set-position "Surface Monitor" '(x 8 y 145))
(cx-gui-do cx-set-position "Surface Monitor" '(x 18 y 159))
(cx-gui-do cx-set-position "Surface Monitor" '(x 39 y 184))
(cx-gui-do cx-set-position "Surface Monitor" '(x 56 y 215))
(cx-gui-do cx-set-position "Surface Monitor" '(x 87 y 258))
(cx-gui-do cx-set-position "Surface Monitor" '(x 115 y 304))
(cx-gui-do cx-set-position "Surface Monitor" '(x 143 y 350))
(cx-gui-do cx-set-position "Surface Monitor" '(x 168 y 392))
(cx-gui-do cx-set-position "Surface Monitor" '(x 192 y 431))
(cx-gui-do cx-set-position "Surface Monitor" '(x 201 y 456))
(cx-gui-do cx-set-position "Surface Monitor" '(x 213 y 481))
(cx-gui-do cx-set-position "Surface Monitor" '(x 221 y 497))
(cx-gui-do cx-set-position "Surface Monitor" '(x 224 y 505))
(cx-gui-do cx-set-position "Surface Monitor" '(x 228 y 517))
(cx-gui-do cx-set-position "Surface Monitor" '(x 231 y 526))
(cx-gui-do cx-set-position "Surface Monitor" '(x 234 y 536))
(cx-gui-do cx-set-position "Surface Monitor" '(x 234 y 540))
```

```
(cx-gui-do cx-set-position "Surface Monitor" '(x 235 y 547))
(cx-gui-do cx-set-position "Surface Monitor" '(x 235 y 553))
(cx-gui-do cx-set-position "Surface Monitor" '(x 235 y 558))
(cx-gui-do cx-set-position "Surface Monitor" '(x 235 y 561))
(cx-gui-do cx-set-position "Surface Monitor" '(x 235 y 562))
(cx-gui-do cx-set-position "Surface Monitor" '(x 235 y 563))
(cx-gui-do cx-set-position "Surface Monitor" '(x 235 y 566))
(cx-gui-do cx-set-position "Surface Monitor" '(x 235 y 568))
(cx-gui-do cx-set-position "Surface Monitor" '(x 236 y 568))
(cx-gui-do cx-set-position "Surface Monitor" '(x 236 y 567))
(cx-gui-do cx-set-position "Surface Monitor" '(x 236 y 563))
(cx-gui-do cx-set-position "Surface Monitor" '(x 236 y 560))
(cx-gui-do cx-set-position "Surface Monitor" '(x 236 y 557))
(cx-gui-do cx-set-position "Surface Monitor" '(x 235 y 555))
(cx-gui-do cx-set-position "Surface Monitor" '(x 235 y 552))
(cx-gui-do cx-set-position "Surface Monitor" '(x 234 y 548))
(cx-gui-do cx-set-position "Surface Monitor" '(x 233 y 545))
(cx-gui-do cx-set-position "Surface Monitor" '(x 232 y 541))
(cx-gui-do cx-set-position "Surface Monitor" '(x 230 y 538))
(cx-gui-do cx-set-position "Surface Monitor" '(x 229 y 535))
(cx-gui-do cx-set-position "Surface Monitor" '(x 228 y 531))
(cx-gui-do cx-set-position "Surface Monitor" '(x 227 y 529))
(cx-gui-do cx-set-position "Surface Monitor" '(x 226 y 527))
(cx-gui-do cx-set-position "Surface Monitor" '(x 225 y 524))
(cx-gui-do cx-set-position "Surface Monitor" '(x 224 y 522))
(cx-gui-do cx-set-position "Surface Monitor" '(x 224 y 521))
(cx-gui-do cx-set-list-selections "Surface
Monitor*Frame1*Table1*Frame2*Table2*DropDownList1(Report Type)" '( 3))
(cx-gui-do cx-activate-item "Surface
Monitor*Frame1*Table1*Frame2*Table2*DropDownList1(Report Type)")
(cx-gui-do cx-set-list-selections "Surface
Monitor*Frame1*Table1*Frame2*Table2*Frame5*Table5*Frame1*List1(Surfaces)" '(
1))
(cx-gui-do cx-activate-item "Surface
Monitor*Frame1*Table1*Frame2*Table2*Frame5*Table5*Frame1*List1(Surfaces)")
(cx-gui-do cx-set-list-selections "Surface
Monitor*Frame1*Table1*Frame2*Table2*Frame5*Table5*Frame1*List1(Surfaces)" '(
1 2))
(cx-gui-do cx-activate-item "Surface
Monitor*Frame1*Table1*Frame2*Table2*Frame5*Table5*Frame1*List1(Surfaces)")
(cx-gui-do cx-set-list-selections "Surface
Monitor*Frame1*Table1*Frame2*Table2*Frame5*Table5*Frame1*List1(Surfaces)" '(
1 2 5))
(cx-gui-do cx-activate-item "Surface
Monitor*Frame1*Table1*Frame2*Table2*Frame5*Table5*Frame1*List1(Surfaces)")
(cx-gui-do cx-set-toggle-button "Surface
Monitor*Frame1*Table1*Frame1*Table1*Frame2(Options)*Table2(Options)*CheckButt
on2(Plot)" #f)
```

```
(cx-gui-do cx-activate-item "Surface
Monitor*Frame1*Table1*Frame1*Table1*Frame2(Options)*Table2(Options)*CheckButt
on2(Plot)")
(cx-gui-do cx-activate-item "Surface Monitor*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton17(Solution
Initialization)")
(cx-gui-do cx-activate-item "Solution
Initialization*Frame1*Table1*ButtonBox10*PushButton2(Initialize)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton18(Calculation
Activities)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton19(Run
Calculation)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton18(Calculation
Activities)")
(cx-gui-do cx-activate-item "MenuBar*AdaptMenu*Yplus/Ystar...")
(cx-gui-do cx-activate-item "Yplus/Ystar
Adaption*PanelButtons*PushButton1(Compute)")
(cx-gui-do cx-activate-item "Yplus/Ystar
Adaption*PanelButtons*PushButton2(Cancel)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton19(Run
Calculation)")
(cx-gui-do cx-set-toggle-button "Run
Calculation*Frame1*Table1*CheckButton16(Solution Steering)" #f)
(cx-gui-do cx-activate-item "Run
Calculation*Frame1*Table1*CheckButton16(Solution Steering)")
(cx-gui-do cx-set-integer-entry "Run
Calculation*Frame1*Table1*IntegerEntry9(Number of Iterations)" 1600)
(cx-gui-do cx-activate-item "Run
Calculation*Frame1*Table1*IntegerEntry9(Number of Iterations)")

(cx-gui-do cx-activate-item "ToolBar*General Tools*savepicture")
(cx-gui-do cx-activate-item "Save Picture*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-text-entry "Select File*Text" "FluentRunCheck.jpg")
(cx-gui-do cx-activate-item "Select File*OK")
(cx-gui-do cx-activate-item "Save Picture*PanelButtons*PushButton2(Cancel)")

(cx-gui-do cx-activate-item "Run
Calculation*Frame1*Table1*PushButton21(Calculate)")
(cx-gui-do cx-set-list-selections "GraphicsArea*GraphicsView1*DropDownList1"
'( 0))
(cx-gui-do cx-activate-item "GraphicsArea*GraphicsView1*DropDownList1")
(cx-gui-do cx-set-list-selections "GraphicsArea*GraphicsView1*DropDownList1"
'( 2))
(cx-gui-do cx-activate-item "GraphicsArea*GraphicsView1*DropDownList1")
(cx-gui-do cx-set-list-selections "GraphicsArea*GraphicsView1*DropDownList1"
'( 1))
(cx-gui-do cx-activate-item "GraphicsArea*GraphicsView1*DropDownList1")
(cx-gui-do cx-set-list-selections "GraphicsArea*GraphicsView1*DropDownList1"
'( 0))
```

```
(cx-gui-do cx-activate-item "GraphicsArea*GraphicsView1*DropDownList1")
(cx-gui-do cx-set-list-selections "GraphicsArea*GraphicsView1*DropDownList1"
'( 2))
(cx-gui-do cx-activate-item "GraphicsArea*GraphicsView1*DropDownList1")
(cx-gui-do cx-activate-item "Information*OK")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton23(Reports)")
(cx-gui-do cx-set-list-selections
"Reports*Frame1*Table1*Frame1*List1(Reports)" '( 0))
(cx-gui-do cx-activate-item "Reports*Frame1*Table1*Frame1*List1(Reports)")
(cx-gui-do cx-activate-item
"Reports*Frame1*Table1*Frame2*Table2*PushButton1(Set Up)")
(cx-gui-do cx-set-list-selections "Flux Reports*Frame2*List2(Boundaries)" '(
1))
(cx-gui-do cx-activate-item "Flux Reports*Frame2*List2(Boundaries)")
(cx-gui-do cx-set-list-selections "Flux Reports*Frame2*List2(Boundaries)" '(
1 2))
(cx-gui-do cx-activate-item "Flux Reports*Frame2*List2(Boundaries)")
(cx-gui-do cx-set-list-selections "Flux Reports*Frame2*List2(Boundaries)" '(
1 2 5))
(cx-gui-do cx-activate-item "Flux Reports*Frame2*List2(Boundaries)")
(cx-gui-do cx-activate-item "Flux Reports*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-list-selections "Flux Reports*Frame2*List2(Boundaries)" '(
1 5))
(cx-gui-do cx-activate-item "Flux Reports*Frame2*List2(Boundaries)")
(cx-gui-do cx-activate-item "Flux Reports*PanelButtons*PushButton1(Write)")
(cx-gui-do cx-set-text-entry "Select File*Text" "ratio")
(cx-gui-do cx-activate-item "Select File*OK")
(cx-gui-do cx-activate-item "Flux Reports*PanelButtons*PushButton2(Cancel)")

(cx-gui-do cx-activate-item "ToolBar*General Tools*savepicture")
(cx-gui-do cx-activate-item "Save Picture*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-text-entry "Select File*Text" "FluentDone.jpg")
(cx-gui-do cx-activate-item "Select File*OK")
(cx-gui-do cx-activate-item "Save Picture*PanelButtons*PushButton2(Cancel)")
(cx-gui-do cx-activate-item "MenuBar*WriteSubMenu*Stop Journal")

(cx-gui-do cx-activate-item "MenuBar*WriteSubMenu*Stop Journal")
exit
o
```

References

[1] Abdelouahid, D., Zine, A., and Nicolas, G., "On the Performance of Ejector Refrigeration Systems," Proceedings of the 5th IASME/WSEAS international conference on Energy & Environment, Greece, 2007, pp. 124-128.

[2] Sriveerakul, T., Aphornratana, S., and Chunnanond, K., "Performance Prediction of Steam Ejector using Computational Fluid Dynamics: Part 2. Flow Structure of a Steam Ejector Influenced by Operating Pressures and Geometries," International Journal of Thermal Sciences, vol. 46, 2007, pp. 823-833. doi:10.1016/j.ijthermalsci.2006.10.012

[3] Gagan, J., Smierciew, K., Butrymowicz, D., and Karwacki, J., "Comparative Study of Turbulence Models in Application to Gas Ejectors," International Journal of Thermal Sciences, vol. 78, 2014, pp. 9-15. doi:10.1016/j.ijthermalsci.2013.11.009

[4] Ruangtrakoon, N., Thongtip, T., Aphornratana, S., and Thanarath, S., "CFD Simulation on the Effect of Primary Nozzle Geometries for a Steam Ejector in Refrigeration Cycle," International Journal of Thermal Sciences, vol. 63, 2013, pp. 133-145. doi:10.1016/j.ijthermalsci.2012.07.009

[5] Sriveerakul, T., Aphornratana, S., and Chunnanond, K., "Performance Prediction of Steam Ejector using Computational Fluid Dynamics: Part 1.Validation of the CFD Results," International Journal of Thermal Sciences, vol. 46, 2007, pp. 812-822. doi:10.1016/j.ijthermalsci.2006.10.014

[6] Kolar, J. and Dvorak, V., "Verification of k-ω SST Turbulence Model for Supersonic Internal Flows," World Acad. Sci. Eng. Technol., vol. 81, 2011, pp. 377-389.

[7] ANSYS Inc., 2012. Fluent User's Guide, Canonburg, PA.

[8] Jeong, H., Utomo, T., Ji, M., Lee, Y., Lee, G., and Chung, H., "CFD Analysis of Flow Phenomena inside Thermo Vapor Compressor Influenced by Operating Conditions and Converging Duct Angles," Journal of Mechanical Science and Technology, vol. 23, 2009, pp. 2366-2375. doi:10.1007/s12206-009-0626-7

[9] Oumar, S., Nicolas, G., and Mikhail, S., "Thermodynamic Study of Multi-Effect Thermal Vapour-Compression Desalination Systems," International Journal of Energy, vol. 72, 2014, pp. 69-79.

[10] Yang, Y. and Lior, N., "Performance Analysis of Combined Humidified Gas Turbine Power Generation and Multi-Effect Thermal Vapor Compression Desalination Systems – Part 1: The Desalination Unit and its Combination with a Steam-Injected Gas Turbine Power System," International Journal on the Science and Technology of Desalting and Water Purification, vol. 196, 2006, pp. 84-104.

[11] Aly, K., EI-Sayed, N., and Ramadan, E., "Experimental Investigation of the Effect of Ejector Geometry on its Performance," International Journal of Nuclear Desalination, vol. 3 (2), 2008, pp. 215-229.

[12] "Shock Diamonds and Mach Disks," Accessed January 20, 2015. Aerospace.org. http://www.aerospaceweb.org/question/propulsion/q0224.shtml

[13] Amanda, M., Karolline, R., and Ricardo, M., "Optimization of a Supersonic Ejector Coupled with a CFD Analysis," Proceedings of the 2012 Spring Meeting & 8th Global Congress on Process Safety, USA, 2012.

[14] Vaclav, D., "Shape Optimization of Supersonic Ejectors with Several Primary Nozzles," Proceedings of the 2nd International Conference on Engineering Optimization, Portugal, 2010.

[15] He, Yilei, "Shape Optimization of Airfoils Without and With Ground Effect Using a Multi-Objective Genetic Algorithm," M.S. Thesis, Washington University in St. Louis, June 2014.

[16] Bandyopadhyay, S. and Saha, S., "Some Single- and Multiobjective Optimization Techniques," in Unsupervised Classification - Similarity Measures, Classical and Metaheuristic Approaches, and Applications, 17-21, Springer-Verlag Berlin Heidelberg, 2013. doi: 10.1007/978-3-642-32451-2

[17] Nosratollahi, M., Mortazavi, M., Adami, A., and Hosseini, M., "Multidisciplinary Design Optimization of a Reentry Vehile using Genetic Algorithm," International Journal of Aircraft Engineering and Aerospace Technology, vol. 82 (3), 2010, pp. 194-203. doi:10.1108/00022661011075928

[18] "Bézier curve," Accessed January 29, 2015, Wikipedia.org, http://en.wikipedia.org/wiki/B%C3%A9zier_curve

Vita

Liju Su

Degrees M.S. Mechanical Engineering, Washington University in St. Louis, August 2015
B.E. Thermal Power and Power Engineering, Hohai University, June 2013

Publications Liju Su and Ramesh K. Agarwal, "CFD Simulation of a Supersonic Steam Ejector for Refrigeration Application," ASME-JSME-KSME Joint Fluids Engineering Conference, Seoul, South Korea, July 26-31, 2015.

August 2015