SETTING UP OPENFOAM TO MODEL GAS FLOW THROUGH A NOZZLE INTO A VACCUUM USING SNAPPYHEXMESH

NUR KHALEDA BINTI ROSLAN

SCHOOL OF AEROSPACE ENGINEERING
UNIVERSITI SAINS MALAYSIA
2019

SETTING UP OPENFOAM TO MODEL GAS FLOW THROUGH A NOZZLE INTO VACUUM USING SNAPPYHEXMESH

by

NUR KHALEDA BINTI ROSLAN

Thesis submitted in fulfilment of the requirements for the Bachelor degree of Engineering (Honours) (Aerospace Engineering)

June 2019

ENDORSEMENT

I, Nur Khaleda Binti Roslan hereby declare that all corrections and comments made by the supervisor and examiner have been taken consideration and rectified accordingly.

____
(Signature of Student)

Date:

____
(Signature of Supervisor)

Name:
Date:

____
(Signature of Examiner)

Name:
Date:

DECLARATION

This thesis is the result of my own investigation, except where otherwise stated and has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any other degree.

(Signature of Student)

Date:

ACKNOWLEDGMENT

First and above all, I thank the Almighty Allah for his endless blessings. This thesis would not have been possible without the support of many people over the years of my study. First of all, I would like to thank my supervisor, Dr. Mohammad Nishat Akhtar for his guidance throughout the research. I would like to thank him a lot for his advice and ideas that he gave me during simulation process. In addition to that, I would also like to express my gratitude to him for his encouragement and help to understand the whole research which enables me to carry out the analysis more smoothly.

Besides, I would also like to express my gratitude to my friends for their help and support. I would like to thank all the technical staffs in the School of Aerospace Engineering Universiti Sains Malaysia for helping me with the software and their knowledge with me. This has certainly helped me in completing the project efficiently.

Last but not the least, I would like to extend my gratitude to the persons that always give moral support to me throughout the whole research. Although sometimes I felt lost in my research, they always come and give me the spirit to finish this research nicely. The persons that I mentioned are my father, Roslan Bin Abu, my mother, Noryati Binti Hashim, my sisters, Nurul Rehana Binti Roslan and Nurul Rahayu Binti Roslan, and my brother, Mohd. Khairul Anuar Bin Roslan. Thank you, a lot, for your moral support and without all of you, maybe I cannot finish this project by the given time.

SETTING UP OPENFOAM TO MODEL GAS FLOW THROUGH A NOZZLE INTO A VACCUUM USING SNAPPYHEXMESH

ABSTRACT

Nowadays, in the field of computational fluid dynamics simulation, the accuracy of the obtained results is a questionable factor with respect to its validation from the experimental data. In terms of CFD simulation, numerous parameters are to be adjusted for the geometry which should be in line with the experimental parameters. However, for CFD, one of the important aspects is the analysis of results at different segment of the geometry and in order to get the precise visualization for such segments, high degree of meshing is required. There is commercially available software which is used to perform meshing but in all of them the meshing is performed manually and is not automated. Moreover, it becomes quite tedious to attain high degree of mesh quality using manual meshing.

For proposed study, in order to determine the accuracy of the CFD results, a three-dimensional chamber model is to be analysed for a compressible gas flow from converging-diverging nozzle using OpenFOAM blockMesh and snappyHexMesh utility. blockmesh and snappyHexMesh are the meshing tools that are used in OpenFOAM software and the mesh in OpenFOAM is generated from the specified dictionary file of these meshing utility tools. In order to enhance the accuracy of the results, different stages of blockMesh and stereolithography (STL) file are created with different size of grid to study and compare the flow behaviour inside the nozzle. It was also observed that the obtained results were in agreement with the experimental results.

In summary, OpenFOAM may be suitable for running CFD simulations in place of the commercial code like Fluent due to its advanced meshing utility. Nonetheless, the OpenFOAM also does not require any license if compared to other software like an ANSYS. Additionally, the simulations in OpenFOAM are typically configured by using text files which allows to readily link the simulation set up to third party applications.

MELARASKAN OPENFOAM UNTUK MODEL ALIRAN GAS MELALUI NOZEL KE VAKUM MENGGUNAKAN SNAPPYHEXMESH

ABSTRAK

Pada masa kini, di dalam bidang simulasi dinamik bendalir berkomputeran, faktor yang dipersoalkan adalah berkenaan dengan ketepatan keputusan yang diperolehi dan pengesahannya dari data eksperimen. Dari segi simulasi CFD, pelbagai parameter perlu diselaraskan supaya geometri sepadan dengan parameter eksperimen. Walau bagaimanapun, bagi CFD, salah satu daripada aspek penting ialah menganalisis keputusan di segmen geometri yang berlainan dan untuk mendapatkan visualisasi yang tepat untuk segmen tersebut, tahap jaringan yang tinggi amat diperlukan. Terdapat perisian yang disediakan secara komersial yang digunakan untuk melakukan jaringan, namun, jaringan tersebut dilakukan secara manual dan tidak automatik. Selain itu, ia menjadi agak meremehkan untuk mendapat tahap jaringan yang tinggi dengan menggunakan jaringan manual.

Untuk kajian yang dicadangkan, untuk menentukan ketepatan keputusan CFD, model ruang tiga dimensi akan dianalisis untuk aliran gas yang boleh dikompres dari muncung tumpu-capah dengan menggunakan utiliti OpenFOAM blockMesh dan snappyHexMesh. blockmesh dan snappyHexMesh adalah alat jaringan yang digunakan dalam perisian OpenFOAM dan jaringan di dalam OpenFOAM dijana daripada kamus fail alat utiliti jaringan ini. Untuk meningkatkan ketepatan keputusan, tahap-tahap blockMesh dan stereolitografi (STL) fail yang berbeza telah dicipta dengan saiz grid yang berbeza untuk mengkaji dan membandingkan keadaan aliran di dalam muncung. Pemerhatian telah dilakukan bahawa hasil yang diperoleh bersesuaian dengan keputusan eksperimen.

Secara ringkasnya, OpenFOAM amat sesuai untuk menjalankan simulasi CFD di tempat kod komersial seperti Fluent kerana utiliti jejaringnya yang canggih. Walau bagaimanapun, OpenFOAM juga tidak memerlukan sebarang lesen jika dibandingkan dengan perisian lain seperti ANSYS. Di samping itu, simulasi dalam OpenFOAM biasanya dikonfigurasikan dengan menggunakan fail teks yang membolehkan untuk menghubungkan simulasi aplikasi pihak ketiga dengan mudah.

TABLE OF CONTENTS

ENDORSEMENT II

DECLARATION XIII

ACKNOWLEDGMENT XIV

ABSTRACT XV

ABSTRAK XVII

LIST OF FIGURES XXI

LIST OF TABLES II

LIST OF ABBREVIATIONS III

NOMENCLATURE IV

CHAPTER 1 14

1.1 Background 14

1.2 Problem Statement 14

1.3 Objectives 15

1.4 Dissertation Outline 15

CHAPTER 2 17

2.1 Mesh generation using *blockMesh* utility 17

2.2 Mesh generation with the *snappyHexMesh* utility 18

2.3 Atmospheric flow over terrain test cases of Askervein 19

CHAPTER 3 23

3.1 Overview 23

3.2 CFD Simulation 23

3.3 The stages flowchart of blockMesh 23

3.4 Flowchart of STL file 25

3.5 Block diagram for the setup of case file. 26

3.6 Installing and running OpenFOAM 27

3.7 The vertices 28

3.8 The blocks 28

3.9 Initial conditions and boundary conditions 30

3.10 Constants and Schemes 30
3.11 Create mesh using *snappyHexMesh* 30
CHAPTER 4 **32**
4.1 Introduction 32
4.2 CFD analysis and result 32
4.3 Comparing between different number of grid generation using *blockMesh*. 33
4.4 Flow visualisation 35
4.5 Grid generation with respect to STL file 38
4.6 Sensitivity Analysis 41
CHAPTER 5 **42**
5.1 Conclusion 42
5.2 Recommendation and Future Work 43
REFERENCES **44**
APPENDIX A **1**
APPENDIX B **2**

LIST OF FIGURES

Figure 2.1 A single block	18
Figure 2.2 Askervein meshed with HypGrid (Taylor and Teunissen 1987)	20
Figure 2.3 Askervein meshed with snappyHexMesh (Taylor and Teunissen 1987)	21
Figure 3.1 Flowchart of blockMesh	24
Figure 3.2 Flowchart of STL file	25
Figure 3.3 Flowchart setup case file	26
Figure 3.4 Description of case file	27
Figure 3.5 Mesh grading along the block edge	30
Figure 4.1 Multiblock-structured mesh with matching cells of Case 1	33
Figure 4.2 Multiblock-structured mesh with matching cells of Case 2	34
Figure 4.3 Illustrate the pressure contour of Case 1	35
Figure 4.4 Illustrate the pressure contour of Case 2	35
Figure 4.5 Velocity contour in diverging section of the nozzle	36
Figure 4.6 STL creation of case 2	38
Figure 4.7 Illustrate the pressure contour of case 2 with respect to STL file	39
Figure 4.8 Illustrate the temperature contour of case 2 with respect to STL file	39
Figure 4.9 Illustrate the velocity contour of case 2 with respect to STL file	40
Figure 4.10 Illustrate the meshing of case 2 with respect to STL file	40

LIST OF TABLES

Table 4.1 Sensivity Analysis 41

Table 4.2 Pressure Range 41

LIST OF ABBREVIATIONS

CFD	Computational Fluid Dynamics
STL	Stereolithography
MD	Molecular Dynamic
Pb	Back Pressure
M	Mach Number
CFL	Courant-Friedrichs-Lewy

NOMENCLATURE

P	Pressure
ρ	Density
R	Gas Constant
T	Temperature

CHAPTER 1

INTRODUCTION

1.1 Background

OpenFOAM is a free, open source computer fluid dynamics (CFD) software package and was utilized to simulate and analyze nitrogen gas passing through various nozzles. A wide-ranging set of solvers are available for use varying from basic CFD codes to electromagnetics. Here, the compressible flow solver, sonicFoam was used which is a transient solver from transonic or supersonic, laminar flow of a compressible liquid (Berboucha, Mangles et al. 2014). There are many advantages of using OpenFOAM because it is free to use. This OpenFOAM is also capable to solve multi-phase flows (Langragian particles and Eulerian) and Newtonian or non-Newtonian fluids. In general, each built-in solver is tailored for specific type of problem.

1.2 Problem Statement

This project details the process of setup, simulation and post-processing for some OpenFOAM test cases such as to model gas flow through a nozzle into a vacuum by using *snappyHexmesh.* However, this project also aims of introducing users to basic procedures of running OpenFOAM. An OpenFOAM cases needs definitions for meshing, initial fields, physical model and control parameters. OpenFOAM data is stored in a set of files within a case directory and this case directory is given a suitable descriptive name for example *cavity*. Editing files is possible in this OpenFOAM because the I/O uses a plain text dictionary fomat with keywords that delivers clear meaning and ease the users to understand even with the least experienced. Equally important, for Window Users, when using a shared directory as example between Windows and

Docker, users may prefer to use a Windows-based text editor. However, care should be taken to make sure that the changes of the text files remain readable by OpenFOAM.

1.3 Objectives

The objectives in this study are:

1. To simulate and analyse the accuracy of compressible gas flow in pressure driven converging diverging nozzle with *blockMesh* utility.

2. To simulate and analyse the accuracy of compressible gas flow in pressure driven converging diverging nozzle with *snappyHexMesh* utility.

1.4 Dissertation Outline

The thesis has been categorized into specific chapters for better viewing and understanding of the study. This dissertation consists of five chapters.

Chapter 1: Introduction – This chapter gives an overview of the thesis, followed by the problem statement to identify, and understand why this research was carried out and its relevance to current times followed by the objectives of this research in order to set the desired target of work and finally the justification of this research.

Chapter 2: Literature review –The focus was on the meshing part of OpenFOAM which used for many engineering applications. The study is also about the effect of shock and gas flow inside the converging-diverging nozzle.

Chapter 3: Methodology - The method is being discussed about the design models. The simulation parameters and boundary condition are also discussed regarding this research.

Chapter 4: Results and simulation validation – The results are discussed in this chapter. The results were compared with the present research.

Chapter 5: Conclusion of this research and suggestion to improve the efficiency of model.

CHAPTER 2

LITERATURE REVIEW

The aim is to generate meshing by using *snappyHexMesh* and the *blockMesh* utility which are the standard mesh tool of the CFD software package OpenFOAM. *snappyHexMesh* meshing utility are known to generate 3-dimensional meshes containing hexahedra (hex) and split-hexahedra (split-hex) automatically. *snappyHexMesh* utility obey to the surface of the geometry by iteratively refining a starting mesh and morphing the resulting split-hex mesh to the surface and increased accuracy of the result. This is due to the smaller resulting mesh in an optional phase of that geometry and the cell layers were inserted to refine the surface of the geometry for a better result. The surface handling is quite tough with a pre-specified final mesh quality. It also runs in parallel with a load balancing step every iteration.

2.1 Mesh generation using *blockMesh* utility

blockMesh utility creates parametric meshes with curved edges and grading and the mesh is generated from dictionary file named *blockMeshDict* which is located in *constant/polyMesh* directory of a case. *blockMesh* principle is to decompose the domain geometry into a set of one or more three dimensional, hexahedral blocks. The mesh is apparently specified as a number of cells in each direction of the block with sufficient information for *blockMesh* to generate mesh data. Each block is defined by 8 vertices at each corner of a hexahedron. Each vertex can be accessed using its label as the vertices are written in a list in which the first element of the list has label '0' due to the C++ convention used by OpenFOAM. However, each block has a local coordinate system (x_1, x_2, x_3) and must be right-handed. A right-handed set of axes is defined for an observer looking down the 0_z axis with 0 nearest them, the arc from a point 0_x axis to a point on the 0_y axis in a clockwise sense.

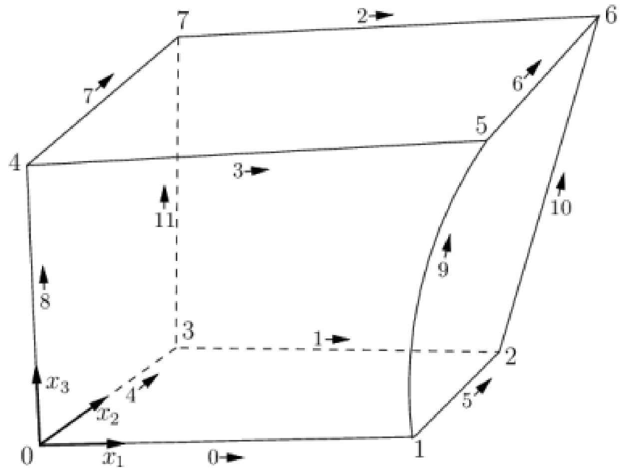


Figure 2.1 A single block

2.2 Mesh generation with the *snappyHexMesh* utility

snappyHexMesh generates 3-dimensional meshes which contain hexahedra (hex) and split-hexahedra (split-hex) automatically from triangulated surface geometries in Stereolithography (STL) format. This meshing approximately matches to the surface by iteratively refining a starting mesh and morphing the resulting split-hex mesh to the surface and the optional phase will shrink back the resulting mesh and insert cell layers. The surface handling is robust with a pre-specified final mesh quality and the mesh refinement level is very flexible. This meshing runs in parallel with a load balancing step for every iteration.

The key steps involved when running snappyHexMesh are:

- Castellation: The cells are beyond a region set by predefines point are deleted
- Snapping: Reconstructs the cells to move the edges from inside the region to the required boundary.
- Layering: Creates additional layers in boundary region.

A direct computational fluid dynamic (CFD) modelling approach was proposed to simulate the flow behaviour in monolithic porous columns (Pawlowski, Nayak et al. 2018). OpenFOAM, an open-source CFD tool, was used to simulate the important parameters for monolith performance characterisation and optimisation for instance velocity and pressure field. The computational mesh of the selected monolith subvolumes was created using Salome v.7.8.0 software and the snappyHexMesh OpenFOAM grid generation utility. This snappyHexMesh utility was used to refine the grid near the boundaries to resolve the boundary conditions more accurate and preliminary mesh independence study was performed to select the appropriate mesh.

Furthermore, CFD tools belonging to the OpenFOAM library were used to mesh the geometry (*snappyHexMesh*) and compute fluid motion (*simpleFoam*) to generate a virtual packed bed that is made of non-spherical polydisperse particles (Pozzobon, Colin et al. 2018). An automatic meshing program called snappyHexMesh was used to mesh the fluid domain in-between the wood chips. Besides, an MPI parallelised molecular dynamics (MD) solver was also implemented entirely within OpenFOAM software framework (Longshaw, Borg et al. 2018). Implementing this mdFoam+ in OpenFOAM enables easier development of hybrid methods that couple MD wit continuum-based solvers.

2.3 Atmospheric flow over terrain test cases of Askervein

In 1987, Taylor and Teunissen had conducted a research of Askervein by using OpenFOAM meshing utility and was compared with EllipSys meshing tool for the result accuracy. For this test cases, flow solvers of OpenFOAM and EllipSys are compared between it meshing strategies. Askervein study was to further the understanding of boundary-layer over relatively low hills, especially as it relates to WECS siting (Taylor and Teunissen 1987). The purpose of this study is to compare OpenFOAM meshing utility with EllipSys on a complex terrain case and as well as on the solvers accuracy. The differences between these two solvers are the wall function. OpenFOAM uses a Nikuradse sand roughness length model, while EllipSys uses Richard and Hoxey's surface roughness (Cavar, Réthoré et al. 2016). In this research, two meshing strategies are investigated by using their utility tools, HypGrid and snappyHexMesh. From the research, it is stated that HypGrid gives consistent smaller meshes than snappyHexMesh. However, these meshes can perform accurately on the Askervein test cases and snappyHexMesh seems to be difficult to be use on very complex terrains.

Methods explained the use of same turbulence model (RANS steady state k-epsilon), a QUICK scheme and a SIMPLE pressure solver and multigrid solver (Taylor and Teunissen 1987). Both EllipSys and OpenFOAM uses different formulation which are Richard and Hoxey and Nikuradse. EllipSys uses structured meshes while OpenFOAM can use both structured and unstructured meshes. Besides, snappyHexMesh has restriction which is the cell aspect ratio must be close to one. Larger aspect ratio will have a slower OpenFOAM converges and larger numerical error. While, EllipSys has no restrictions and it uses a hyperbolic mesh generator that creates smaller meshes.

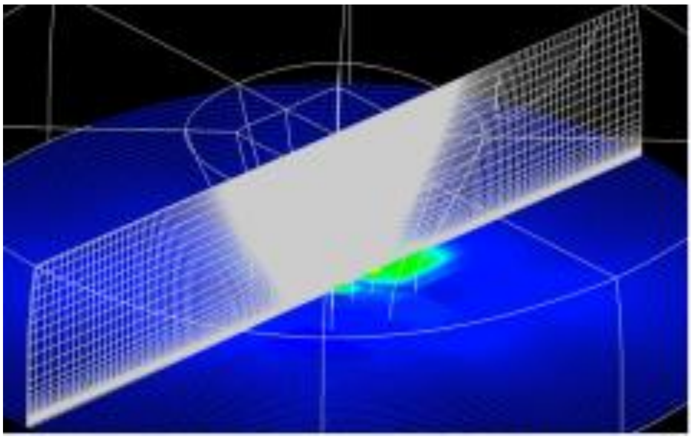


Figure 2.2 Askervein meshed with HypGrid (Taylor and Teunissen 1987)

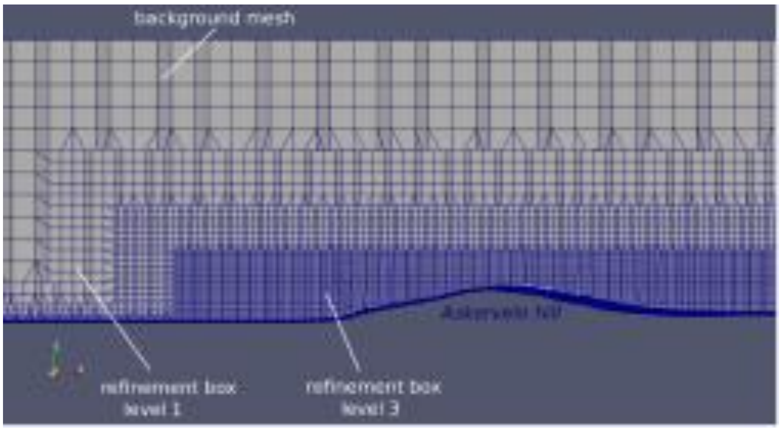


Figure 2.3 Askervein meshed with snappyHexMesh (Taylor and Teunissen 1987)

However, OpenFOAM has few limitations compared to EllipSys, thus, it will affect the quality of the simulation. From this research, it is proven that OpenFOAM and EllipSys predicted very similar speed-up's over the hilltop and only small differences for the first few metres over terrain. This may be due to the differences between the wall function inconsistency and the grid refinement. Besides, snappyHexMesh predicted higher speed-up but the reasons remain uncertain due to the restriction of the mesher and yielding more inaccuracy (Taylor and Teunissen 1987). In conclusion, Taylor and Teunissen have mentioned that OpenFOAM software has a restriction which is the cell aspect ratio must be always close to one and it will affect the accuracy of the result which will produced inconsistent result.

Apart from that, the simulation time for OpenFOAM was approximately ten times slower than EllipSys. snappyHexMesh created larger meshes than HypGrid even though OpenFOAM have been saved by using unstructured meshes. For this case, it was difficult to generate terrain mesh with OpenFOAM. Nikuradse law of the wall for rough surface is known to be problematic and inconsistent and the inaccuracy in OpenFOAM can be seen in the vertical direction of the first few cells of computational domain. By implementing new wall boundary conditions based on Richard and Hoxey's profiles in OpenFOAM can help reducing the inconsistency issues.

As conclusions, from the test cases, snappyHexMesh utility has its potential when observing fairly good agreement with respect to the simulations performes with EllipSys and OpenFOAM on HypGrid mesh (Taylor and Teunissen 1987). However, it was difficult to maintain the total amount of cells to a reasonable value since the utility have some limits and also due to the limitation of aspect ratio of the background mesh.

CHAPTER 3

METHODOLOGY

3.1 Overview

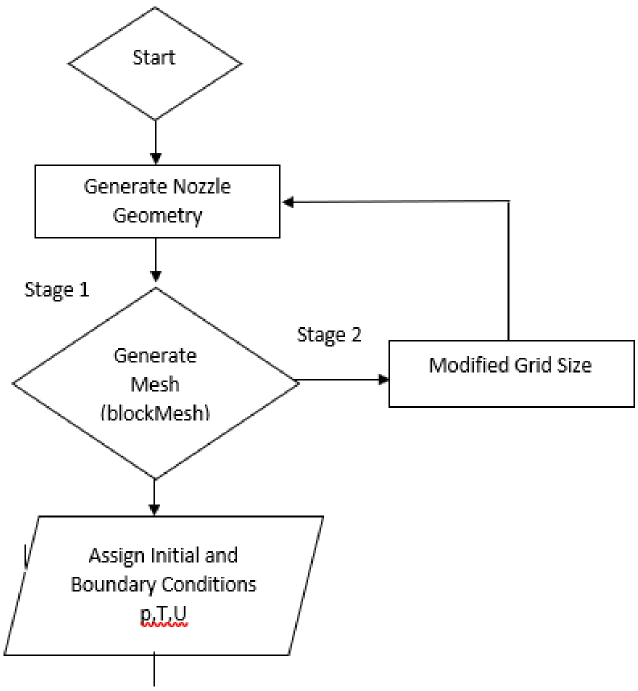
This chapter discuss the methodology for the whole research including simulation validation. This chapter explained about the parameter used in this simulation test setup. Simulation process can also find in detail manner including boundary conditions.

3.2 CFD Simulation

The modelling and simulation of the converging-diverging nozzle using OpenFOAM v6. The simulation has been done for two cases of converging-diverging nozzle. To start this simulation, the geometry of a converging-diverging nozzle model was created similar to the present research. Figure below describes the flow chart of the simulation process.

3.3 The stages flowchart of blockMesh

Figure below describes the flow chart of the simulation process.



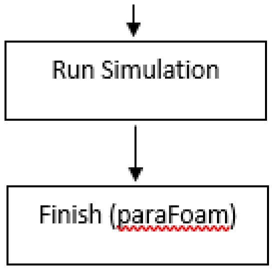


Figure 3.1 Flowchart of blockMesh

To start the simulation processes, converging-diverging nozzle geometry was created after running the application of nozzle cases. This application is designed to be executed from a terminal command line and data files for this case is stored in a directory. For stage 1, simple meshes of blocks of hexahedral cells was generated with 8200 hexahedral cells by using blockMesh utility. This process was proceeded with the physical properties that is required to define in this case such as pressure, temperature and velocity. After solving the test case, the results of contour plots were analysed by using paraFoam that uses ParaView software. However, this process was repeated for stage 2 with 32800 number of hexahedral cells by using blockMesh utility.