Numerical Analysis of Air Film Cooling Effectiveness at High Temperature and Pressure

VEASNA MOM

Undergraduate Mechanical Engineering

Gujarat Power Engineering and Research Institute

Managed By Gujarat Technological University

Synopsis

High temperature gas turbines operate under extreme conditions, necessitating effective cooling techniques to maintain durability and performance. This research aims to numerically simulate air film cooling using OpenFOAM-v2312, focusing on a 2D flat plate geometry at low temperature and low pressure. The **buoyantBoussinesqSimpleFoam** solver for incompressible fluid is employed, and the results are compared against the findings of Li & Wang [1]. The simulation investigates the effects of various parameters, including blowing ratio, mainstream temperature, and injection angle, on the cooling performance. The mainstream flow has a temperature of 400 K and a velocity of 10 m/s, while the coolant air is injected with a blowing ratio of 1.32, a velocity of 10 m/s, and a temperature of 300 K. All walls are considered adiabatic, with the top wall set to a slip condition and the remaining walls set to no-slip conditions. The turbulence dissipation rate and kinetic energy at the inlets are set to $\epsilon = 1\ m^2/s^3$ and $k = 1\ m^2/s^2$, respectively.

This research migration project aims to validate OpenFOAM's capabilities against commercial software, addressing financial barriers associated with software licenses. By analyzing two key parameters influencing film cooling which are blowing ratio and mainstream temperature.

1. Introduction

Gas turbines are critical components in both aircraft propulsion systems and land-based power generation plants, operating under extreme conditions of high temperature and pressure. The operational temperatures often exceed the melting points of the turbine blade materials, making effective cooling strategies essential for maintaining the structural integrity and performance of these components. Managing the thermal loads in gas turbines is, therefore, a significant engineering challenge.

To address this, various cooling techniques have been developed. One prominent method is internal cooling, specifically convective cooling, which involves the circulation of coolant within the turbine blades. Another widely used technique is film cooling, where a coolant is injected onto the surface of the turbine blades to form a protective film that insulates the blades from the hot gases. A notable study by Li and Wang [1] investigated film cooling with mist injection to enhance cooling effectiveness, comparing it against scenarios where only air is used as the coolant. Their research was conducted numerically using Ansys Fluent, a commercial computational fluid dynamics (CFD) software. The objective of this study is to utilize OpenFOAM, a powerful open-source CFD toolbox, to simulate air film cooling and validate its results against those obtained by Li and Wang using Ansys Fluent. This research aims to demonstrate OpenFOAM's capability as a viable alternative to commercial CFD software, which often requires expensive licenses, thereby posing financial barriers to researchers, students, and educational institutions.

By comparing the results from OpenFOAM with those from Ansys Fluent, this study seeks to establish the reliability and accuracy of OpenFOAM in simulating complex cooling processes in gas turbines. Additionally, this research aims to explore the potential benefits and challenges associated with using open-source software for high-fidelity CFD simulations in academic and industrial applications.

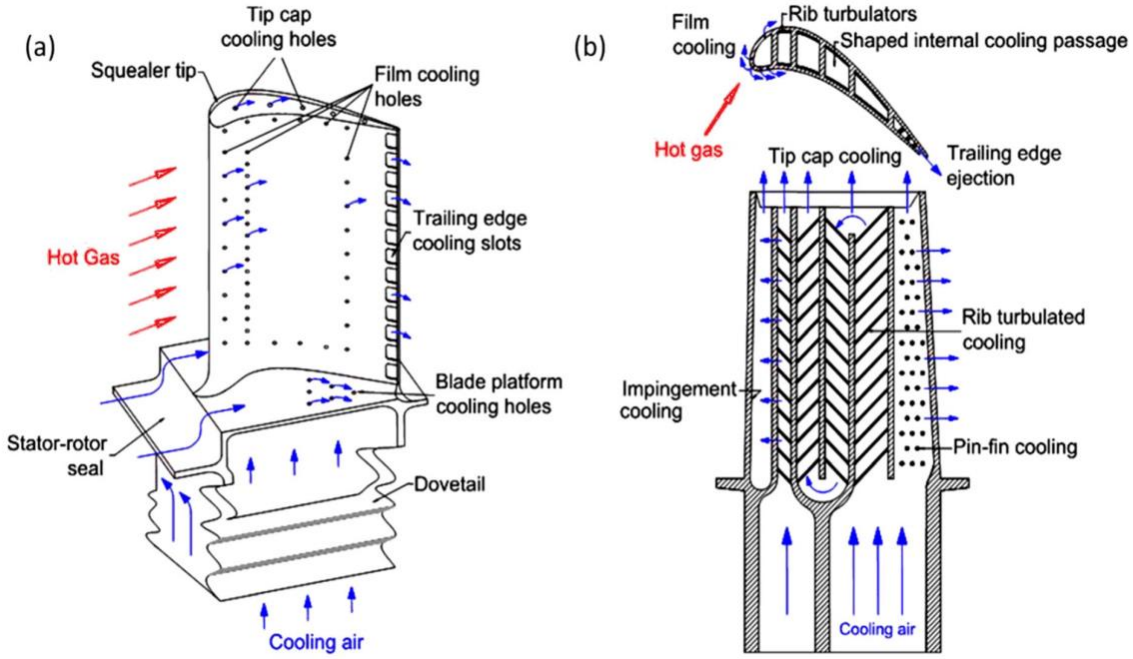


Figure 1: Gas turbine blade cooling schematic: (a) film cooling, (b) internal cooling, [2]

2. Problem Statement

The author considered the 2D plate geometry shown in **Figure (2)** with the main stream temperature is 400K and velocity 10 m/s. Likewise, the injection coolant (air) with respect to blow ratio M=1.32, velocity 10 m/s and Temperature 300K. All the walls are considered to be adiabatic. Topwall is set to slip condition and the remaining wall are no slip conditions. The inlet gas and coolant turbulences dissipation rate $\epsilon = 1\ m^2/s^3$ and turbulences kinetic energy is $k = 1\ m^2/s^2$. To comprehensively understand and evaluate the performance of film cooling, this study will systematically vary key parameters such as the blowing ratio, mainstream temperature, and injection angle. The objective is to gain insights into the influence of these parameters on the cooling effectiveness and overall thermal performance of the system

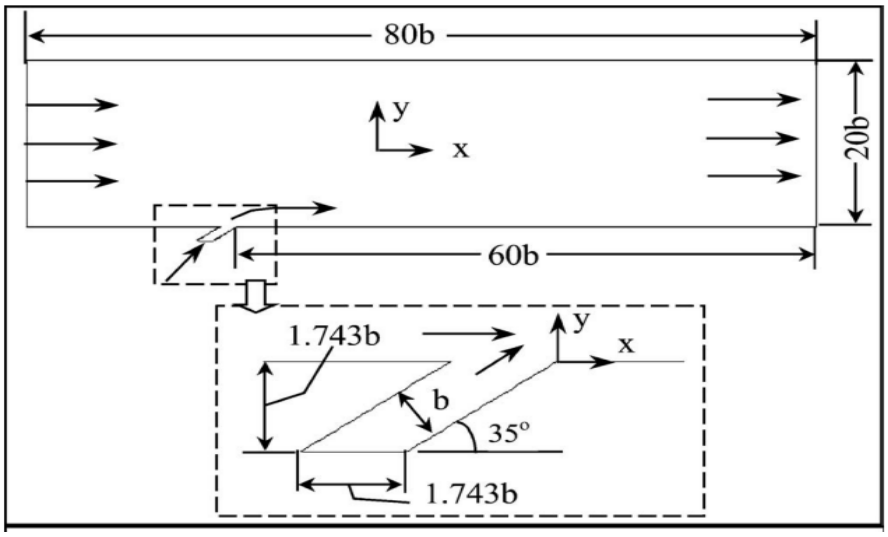


Figure 2: Computational Domain,[1]

3. Governing Equations

In this project the RANS (Reynold Average-Navier Stroke) was implemented for incompressible fluid and using Boussinesq's Approximation for density variation based on temperature and thermal expansion of fluid in the gravity-driven source term.

3.1 Continuity Equation

The continuity equation is expressed as: $$\frac{\partial \bar{u}_j}{\partial x_j} = 0 \quad (1)$$

3.2 Momentum Conservation Equation

The taken solver is designed for incompressible flow, thereby assuming that the density,ρ remains constant throughout the flow field, except when considering buoyancy effects in the source term. The filtered or averaged momentum equation, incorporating buoyancy, is therefore expressed as follows:

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_i}\left(\bar{u}_j \bar{u}_i\right) = -\frac{\partial}{\partial x_i}\left(\frac{\bar{p}}{\rho_o}\right) + \frac{1}{\rho_o}\frac{\partial}{\partial x_j}\left(\tau_{ij} + \tau_{t_{ij}}\right) + \frac{\bar{\rho}}{\rho_o} g_i \qquad (2)$$

Where g_i is the gravity acceleration vector $\tau_{t_{ij}}$ is the turbulence stress tensor and

$\tau_{ij} = \mu\left[\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \frac{2}{3}\left(\frac{\partial \bar{u}_k}{\partial x_k}\right)\delta_{ij}\right]$ is viscous stress tensor

The final equation, using Boussinesq's Approximation for density in the source term due to buoyancy effect, can be expressed as:

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_i}\left(\bar{u}_j \bar{u}_i\right) - \frac{\partial}{\partial x_j}\left\{\nu_{eff}\left[\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \frac{2}{3}\left(\frac{\partial \bar{u}_k}{\partial x_k}\right)\delta_{ij}\right]\right\} = -\frac{\partial \tilde{p}}{\partial x_i} + g_i\left(1 - \beta(\bar{T} - T_{ref}\right) \quad (3)$$

Where:

- $\nu_{eff} = \nu_o + \nu_t$ is the effective kinematic viscosity.
- $\tilde{p}$ is resolved kinematic pressure
- $\rho_k = 1 - \beta(T - T_{ref}\ )$ is the effective kinematic density
- β is the thermal expansion
- T_{ref} is the reference temperature

3.3 Energy Conservation Equation

The instantaneous internal energy equation is given by

$$\frac{\partial}{\partial t}(\rho e) + \frac{\partial}{\partial x_j}\left(\rho e u_j\right) = -p\frac{\partial u_k}{\partial x_k} + \tau_{ij}\frac{\partial u_i}{\partial x_j} - \frac{\partial q_k}{\partial x_k} \quad (4)$$

Where:

- Term $-p\frac{\partial u_k}{\partial x_k}$ = 0 domination of continuity equation
- Term $\tau_{ij}\frac{\partial u_i}{\partial x_j}$ can be neglected for incompressible flow

Simplified,

$$\frac{\partial}{\partial t}(\rho e) + \frac{\partial}{\partial x_j}\left(\rho e u_j\right) = -\frac{\partial q_k}{\partial x_k} \quad (5)$$

For convective term will be decomposed to $\rho(\bar{e} + e')(\bar{u} + u')$ where the prime denotes fluctuation or residual quantity. Equation (4) Becomes

$$\frac{\partial}{\partial t}(\rho e) + \frac{\partial}{\partial x_j}\left(\rho \bar{e}\bar{u}_j\right) = -\frac{\partial q_{t_k}}{\partial x_k} - \frac{\partial q_k}{\partial x_k} \quad (6)$$

Where $q_{t_k} = \overline{\rho \bar{e} u_k'} + \overline{\rho e' \bar{u}_k} + \overline{\rho e' u_k'}$ is turbulence heat flux

Now, by using Fourier's law of heat conduction $\bar{q}_k = -k\frac{\partial \bar{T}}{\partial x_k}$ with constant density $\rho = \rho_o$ and $\mu = \mu_o$ as well as the fact that $e = c_p T$, substitute these relations in equation (5). It becomes

$$\frac{\partial \bar{T}}{\partial t} + \frac{\partial}{\partial x_j}\left(\bar{T}\bar{u}_j\right) = \frac{\partial}{\partial x_k}\left[\left(\frac{\mu_o k}{\mu_o \rho_o c_p}\right)\frac{\partial \bar{T}}{\partial x_k}\right] + \frac{\partial}{\partial x_k}\left[\left(\frac{\mu_t k}{\mu_t \rho_o c_p}\right)\frac{\partial \bar{T}}{\partial x_k}\right]$$

$$\frac{\partial \bar{T}}{\partial t} + \frac{\partial}{\partial x_j}\left(\bar{T}\bar{u}_j\right) = \frac{\partial}{\partial x_k}\left[\left(\frac{\nu_t}{Pr_t} + \frac{\nu_o}{Pr}\right)\frac{\partial \bar{T}}{\partial x_k}\right] \quad (7)$$

Mean Temperature Equation

$$\frac{\partial \bar{T}}{\partial t} + \frac{\partial}{\partial t}\left(\bar{T}\bar{u}_j\right) = \frac{\partial}{\partial x_k}\left(\alpha_{eff}\frac{\partial \bar{T}}{\partial x_k}\right) \quad (8)$$

Where $\alpha_{eff} = \frac{\nu_t}{Pr_t} + \frac{\nu_o}{Pr}$ is heat transfer coefficient or effective thermal diffusivity

- In this specific solver of OpenFOAM-v2312 within **TEqn.H** file, the equation of temperature to be solved expressed as:

$$\nabla \cdot (\phi \mathrm{T}) - \nabla(\alpha_{\mathrm{eff}}\nabla T) = S_T \quad (9)$$

  - S_T are source terms including radiation and others source if available

3.4 Turbulence Model

The k-epsilon model was incorporated into the study with enhanced wall treatment. In OpenFOAM (More details in User Guide), the k-epsilon is written in general equation as follows

Turbulence Kinetic Energy Equation:

$$\frac{\partial}{\partial t}(\alpha\rho k) + \nabla.(\alpha\rho \mathrm{uk}) - \nabla^2(\alpha\rho D_k) = \alpha\rho G - \left(\frac{2}{3}\,\alpha\rho\nabla.uk\right) - \left(\alpha\rho\frac{\epsilon}{k}k\right) + S_k + S_{fvOption} \quad (9)$$

Turbulence Kinetic Energy dissipation rate:

$$\frac{\partial}{\partial t}(\alpha\rho\epsilon) + \nabla.(\alpha\rho \mathrm{u}\epsilon) - \nabla^2(\alpha\rho D_\epsilon\epsilon) = C_1\alpha\rho G\frac{\epsilon}{k} - \left(\left(\frac{2}{3}C_1 - C_{3,RDT}\right)\alpha\rho\nabla.u\epsilon\right) - \left(C_2\alpha\rho\frac{\epsilon}{k}k\right) + S_\epsilon + S_{fvOption} \quad (10)$$

For anisotropic turbulence,

$$k = \frac{3}{2}\left(I\left|u_{ref}\right|\right)^2$$

- I = Turbulence Intensity [%]
- u_{ref} = A reference flow speed [ms^{-1}]

$$\epsilon = \frac{C_\mu^{0.75} k^{1.5}}{L}$$

- C_μ = A model constant 0.09 by default
- L = A reference length scale [m]

4. Simulation Procedure

4.1 Geometry and Mesh

The mesh was created utilizing the blockMesh utility available in OpenFOAM. Comprehensive coordinate details were calculated and specified within the **system/blockMeshDict** OpenFOAM directory in the case folder. Four blocks were selected, numbered from 0 to 3, as depicted in the **figure (3)**. It is imperative to acknowledge that OpenFOAM's blockMesh utility necessitates the correct ordering of vertices to successfully generate the block mesh. The arrow displayed in block 0 denotes the direction in which the vertices are arranged.

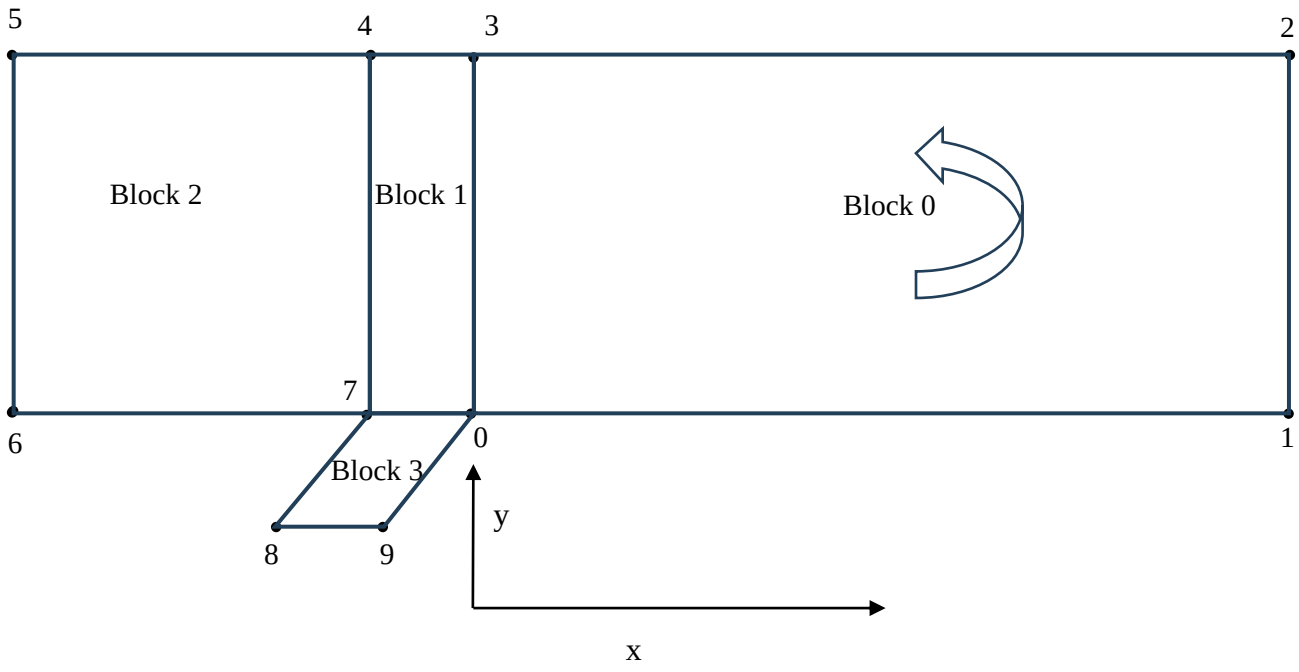


Figure 3: Number Blocks

The offset vertices are specified in a similar manner, starting from 10 to 19. The provided **figure (4)** exemplifies block 0, with coordinates orderly arranged to correspond with the offset in the z-axis.

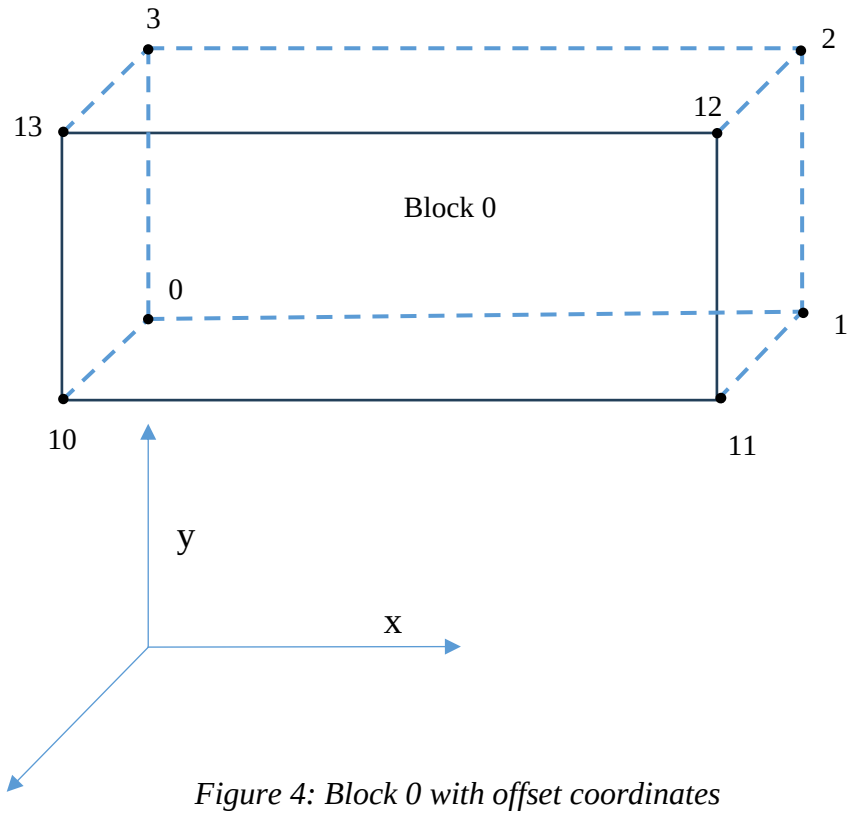


Figure 4: Block 0 with offset coordinates

By considering anti-clockwise direction, the block 0 can be created by using hexahedral keyword as: **hex (0 1 2 3 10 11 12 13)**.

4.1.1 Mesh Generation and Grading

In order to generate the necessary cells and grading for each block, it is important to carefully review the **blockMeshDict** file, specifically focusing on the **blocks** section. This is where the number of blocks, cell divisions, and grading are specified. Here is an illustrative snippet for block 0.

hex (0 1 2 3 10 11 12 13) (220 65 1) simpleGrading (1 -0.067 1)

220 cells in x-direction

65 cells in y-direction

1 cell in z-direction

the cells are equal in size along the x-axis and z-axis, whereas along the y-axis cells gradually increase in size with a 0.067 expansion ratio.

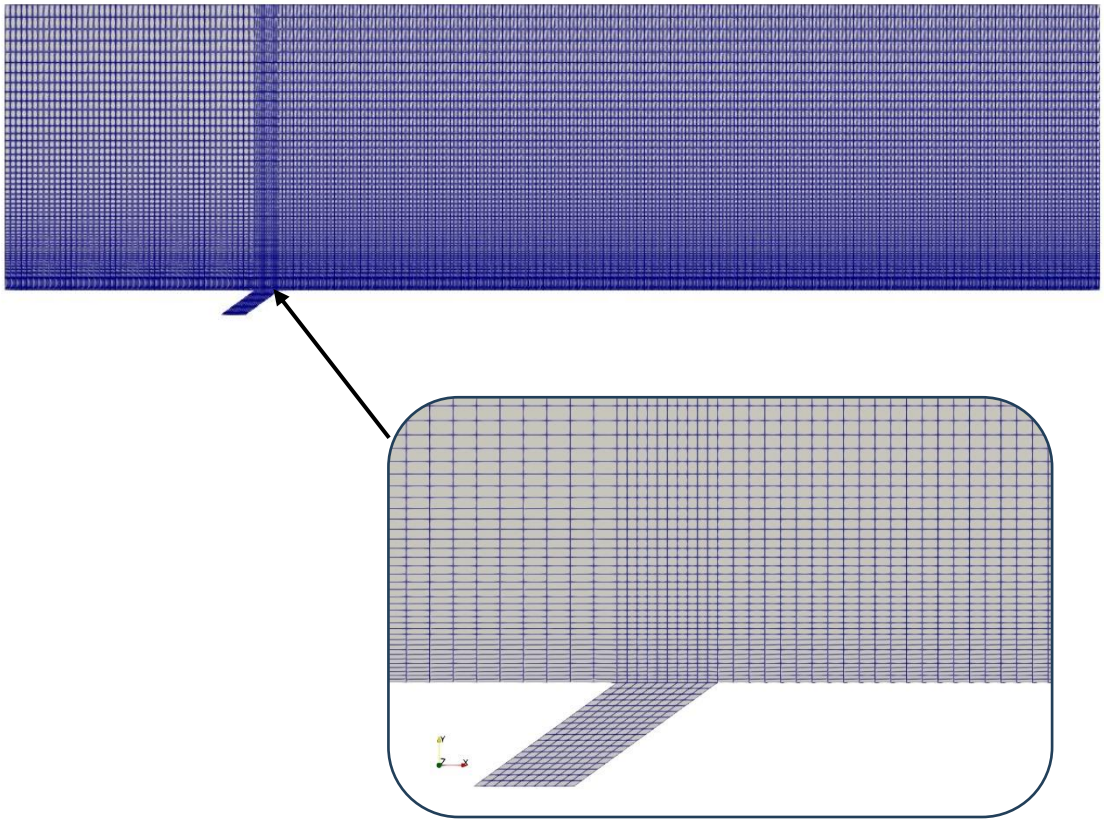


Figure 5: Mesh and Refinement

Mesh	BlockMesh Utility
Mesh Detail	18055elements, structured Max aspect ratio = 6.85068 Max skewness = 0.845943 Min volume = $3.30394e^{-10}$ Max volume = $1.1475e^{-08}$ Non-orthogonality Max: 55 Average: 5.14574 First Cell width bottom_plate wall 0.000236882 boundingBox: (-0.08 -0.006972 0) (0.24 0.08 0.002)

Table1: Mesh Details

4.2 Initial and Boundary Conditions

In the context of realistic physical phenomena and the impact of simulation output, the appropriate boundary condition plays a crucial role. For the case set up in the **0** folder of the OpenFOAM case directory, the following details are pertinent: the files named **p, T, U, T, k**, **epsilon, p_rgh** and **nut**. In this case, the pressure at walls is set to be zeroGradient. Employing the enhanced wall treatment for k and epsilon is essential for accurately capturing near-wall behaviour, utilizing the specified value from the reference paper ($1 < \boldsymbol{y^+} < \mathbf{5}$). The boundary condition at the bottom plate wall is of utmost importance for ensuring the accuracy and stability of simulation results. Initially, the turbulence kinetic energy **(k)** was set up using **kqRWallFunction**, while **epsilonWallFunction** was used for turbulence kinetic energy dissipation rate ($\boldsymbol{\epsilon}$). However, this approach yielded significantly different results compared to the reference paper. This discrepancy arose because the standard ($\boldsymbol{k-\epsilon}$) model was primarily designed for high Reynolds turbulence flow with recommendation of $\boldsymbol{y^+ > 30}$, providing less accurate results in low Reynolds flow regions, especially near the wall where resolution of the viscous sublayer is necessary. To rectify this issue the **lowRecorrection** was implemented to enhance accuracy by employing stepwise blending method that can be found in OpenFOAM user guide in detail.

Similarly, the **kLowReWallFunction** was selected for bottom plate and injection to approximate the turbulence kinetic energy $\boldsymbol{k}$ profile near wall regions, accounting for viscosity and low Reynolds effects. Another crucial boundary condition, **nutLowReWallFunction** was introduced for bottom plate wall and hole wall. This boundary wall function allow solver to resolve the turbulence viscosity near the wall. It sets the turbulence viscosity to zero and provide an access to calculate $\boldsymbol{y^+}$.

Boundaries	**Pressure**	**Velocity**	**Temperature**	**k**	**epsilon**
inlet_gas	zeroGradient	uniform (10 0 0)	400K	fixedValue uniform 1	fixedValue uniform 1
inlet_coolant	zeroGradient	uniform (8.19 5.73 0)	300K	fixedValue uniform 1	fixedValue uniform 1
outlet	fixedValue uniform 0	zeroGradient	zeroGradient	zeroGradient	zeroGradient
topwall	zeroGradient	slip	zeroGradient	kqRWallFunction uniform 0	epsilonWallFunction uniform 1

bottom_plate	zeroGradient	noSlip	zeroGradient	kLowReWallFunction uniform 0	epsilonWallFunction lowRecorrection true
hole_wall	zeroGradient	noSlip	zeroGradient	kLowReWallFunction uniform 0	epsilonWallFunction lowRecorrection true

Table2: Initial and Boundary Conditions

Notes: $v_c = 10\ m/s$ is expressed in components as ($v_c cos\theta$, $v_c sin\theta$, 0) for the coolant inlet velocity.

Boundaries	**p_rgh**	**nut**
inlet_gas	fixedValue $internalField	calculated uniform 0
inlet_coolant	fixedValue $internalField	calculated uniform 0
outlet	fixedValue $internalField	calculated uniform 0
topwall	zeroGradient	nutkWallFunction; uniform 0
bottom_plate	zeroGradient	nutLowReWallFunction uniform 0
hole_wall	zeroGradient	nutLowReWallFunction uniform 0

Table3: Initial and Boundary Conditions p_rgh &nut

4.3 Solver

The **buoyantBoussinesqSimpleFoam** was used to simulate this problem. Since in this study we focus only incompressible flow with heat transfer so that **buoyantBoussinesqSimpleFoam** is suitable for present scenario even though it is not corresponding to real operation of gas turbine, it serves as a conceptual study toward finding optimization parameters. However, in actual practice the compressible solvers or multiphase flow solver can be considered as per requirement and the flow physic.

Characteristic:

- Heat Transfer Solver
- Incompressible Fluid

- Steady-State Solver
- Buoyancy-Driven

4.4 Execution and Parallel Computing

To execute the case the follow commands are used in the terminal:

- Step 1: **blockMesh** this command is used to generate mesh
- Step 2: **buoyantBoussinesqSimpleFoam** this command is run the case with this solver
- Step 3: **paraFoam** this command to is used to view the result in ParaView or,
- Step3: **touch** desiredname.foam this command is used to create (.foam) file that can be opened using ParaView in case paraFoam is not working, especially for window users

➢ **Parallel Computing**

OpenFOAM represents a robust computational tool that enables users to conduct parallel simulations, thereby providing time-saving capabilities for complex problems at no cost, a feature not typically available in commercial software. Comprehensive information regarding mesh decomposition is available in the OpenFOAM user guide. To initiate parallel computing, it is essential to decompose the mesh into various parts, each of which is subsequently allocated to specific processor cores for concurrent computation. For this project, the number of domain decompositions can be found in the **system/decomposeParDict** directory. This project was simulated using 6 core processor computer and the following is the workflow for execution:

- Step 1: **blockMesh**
- Step 2: **decomposePar** this command is used to decompose mesh in to different parts
- Step 3: **mpirun -np 6 buoyantBoussinesqSimpleFoam -parallel**
- Step 4: **reconstructPar**
- Step 5: **paraFoam**

Where:

- -np 6: means number of processors will be used are 6 cores, this number depends on the domain decomposition that was specified in **decomposeParDict** and number of cores cannot exceed the availability of the computer.

5. Results and Discussions

The study involved numerically simulating parameters such as the blowing ratio, coolant injection angle, and mainstream temperature, and then comparing the results with data from Li and Wang's study. Relevant survey data was meticulously extracted using the online tool PlotDigitizer to ensure precise digitization of graphical information. Further analysis and

detailed plotting were conducted using MATLAB for accurate data representation and comparison. Additionally, ParaView software played a crucial role in visualizing flow characteristics through advanced graphical representations and contour plotting, providing comprehensive insights into the spatial distribution of key flow variables. ParaView was also used to extract detailed data from the simulations, enabling a thorough analysis of the flow phenomena and ensuring accurate interpretation and validation of the numerical results against the experimental data. This combination of tools and methodologies ensured a robust and comprehensive approach to studying flow dynamics under various conditions.

5.1 Parameters Definition

- Blowing Ratio is defined by $M = \frac{(\rho V)_c}{(\rho V)_\infty}$ where the subscript c stands for the properties at the cooling hole inlet (secondary flow), and ∞ refers to properties of mainstream inlet flow (primary flow). The blowing ratio signifies the amount of coolant to be injected to the hole against high temperature mainstream. From the Li & Wang paper four different blowing ratios were taken into consideration whose value M=0.66, M=1.32, M=1.05, and M=1.58. The blowing ratio of M=1.32 is the default value corresponding to velocity$V_\infty = 10\ m/s$ and $V_c = 10\ m/s$.
- The performance of film cooling is expressed using the adiabatic film cooling effectiveness η , defined as:

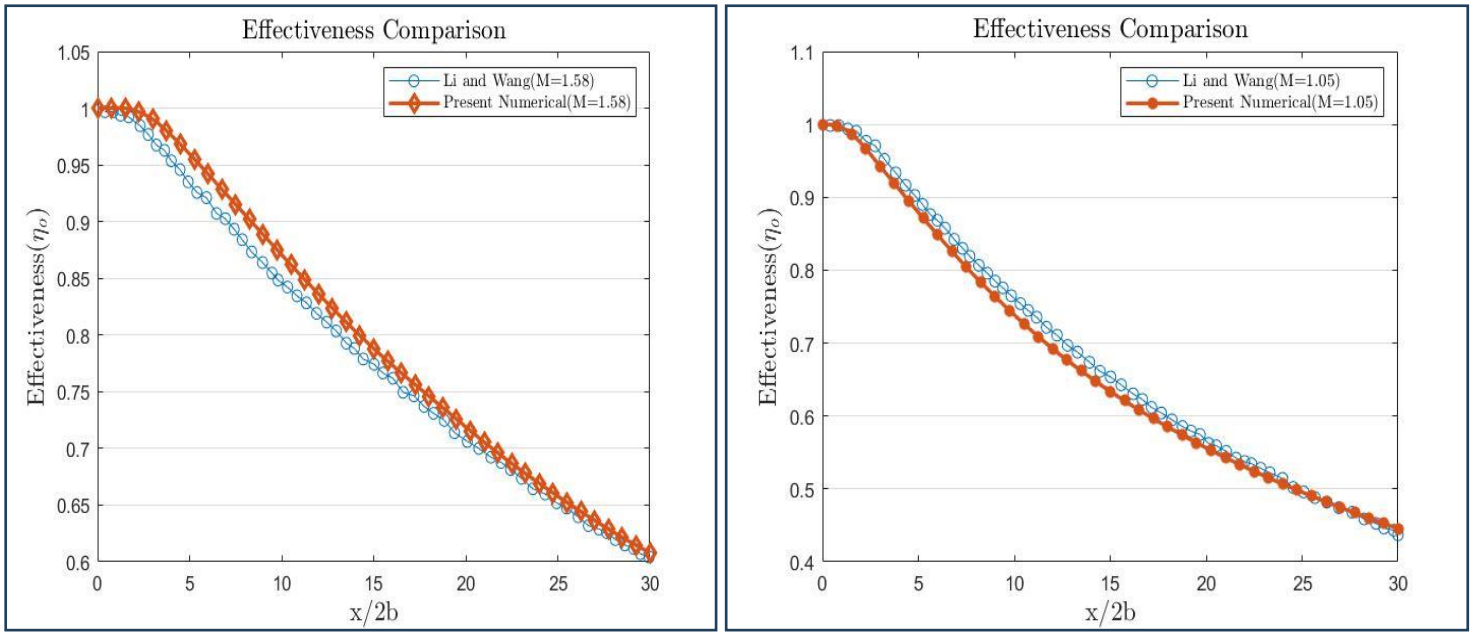
$$\eta = \frac{T_g - T_w}{T_g - T_c}$$

Where T_g is the temperature of mainstream hot gas, T_w is the temperature of the wall and T_c is the temperature of coolant at injection hole. This effectiveness provides a quantitative measure of adiabatic plate.

5.2 Validation by Comparing Effectiveness

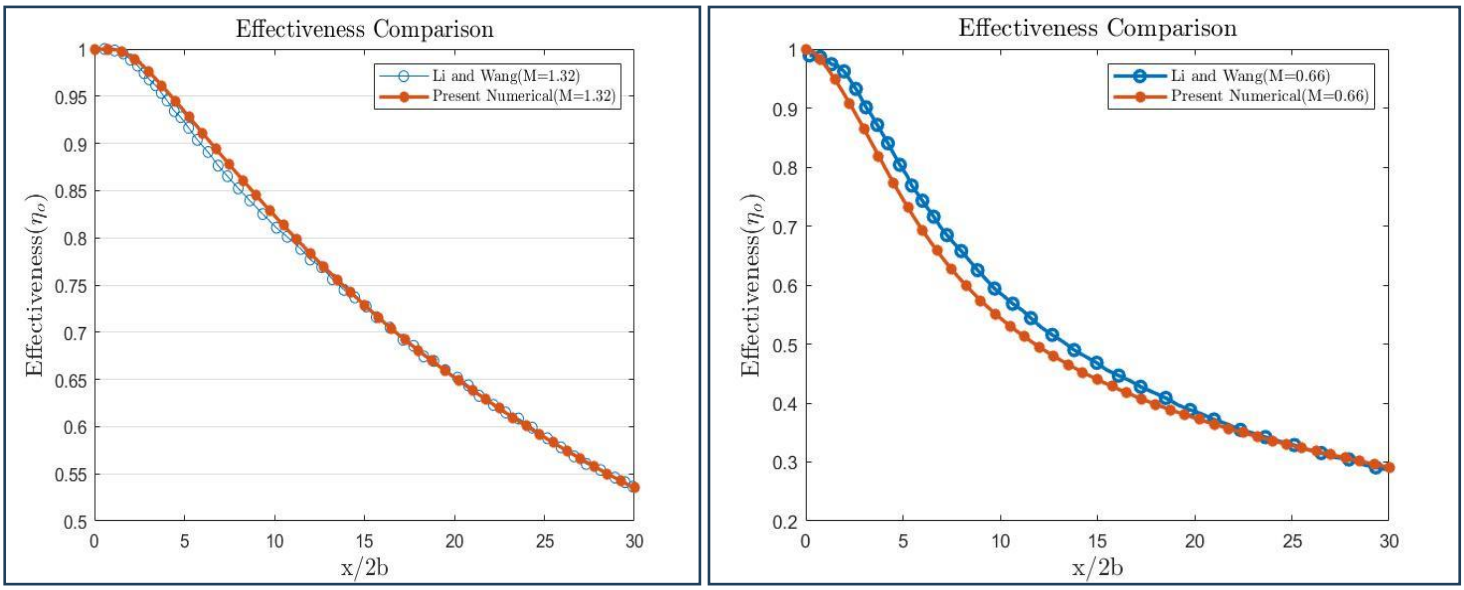
The figures presented below depict the effectiveness of film cooling as obtained from computational simulations incorporating diverse blowing ratios, in comparison with the findings articulated in Li and Wang's influential work [1]. Upon careful examination, it becomes evident that at a blowing ratio of M=1.32, the observed thermal effectiveness closely approximates the values reported in the reference literature. Nevertheless, it is imperative to note that simulations conducted at blowing ratios of M=0.66, M=1.05, and M=1.58 exhibit

minor discrepancies in contrast to the reference, signifying a general tendency towards agreement with the established findings. This elucidates the nuanced nature of the results and highlights the need for comprehensive validation studies to further refine the understanding of film cooling processes.



(a) (b)

Figure 6: Effectiveness Comparison of Blowing Ratio (a) M=1.85 and (b) M=1.05



(c) (d)

Figure 7: Effectiveness Comparison of Blowing Ratio (c) M=1.32 and (d) M=0.66

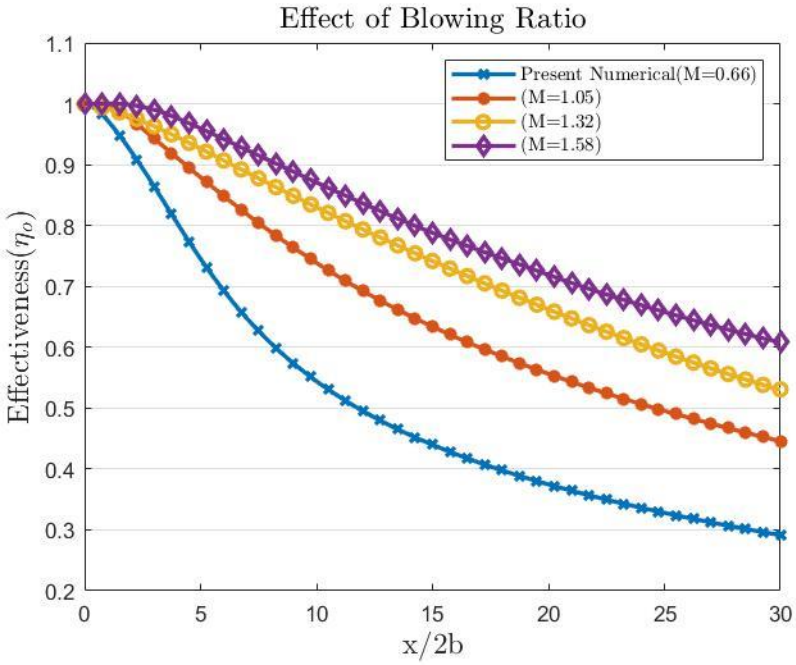


Figure 8: Effect of Blowing Ratio on Adiabatic Effectiveness

- Reason on the effect of blowing ratio
  - As the blowing ratio is increased, the more coolant will be injected into mainstream which results in increasing cooling effectiveness

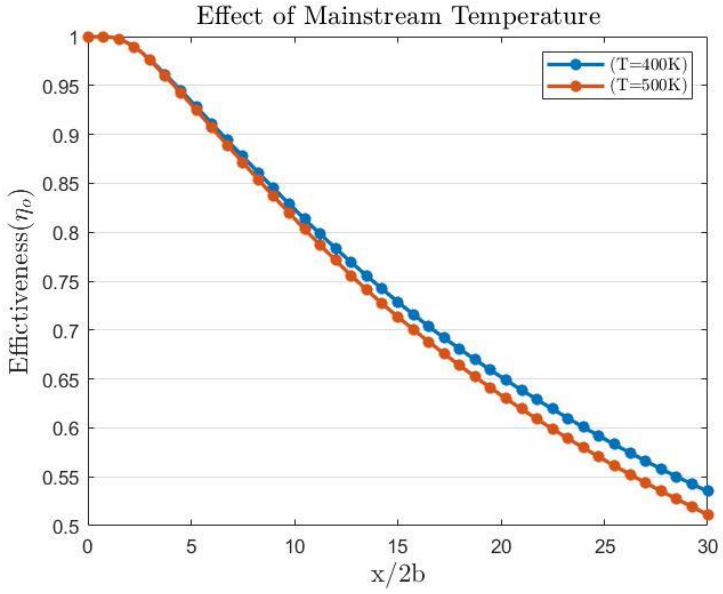


Figure 9: Effect of Mainstream Temperature on Adiabatic Effectiveness

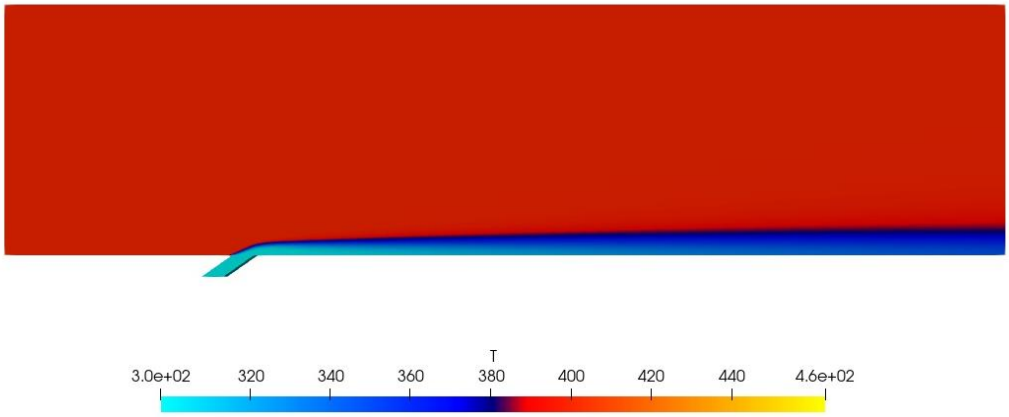


Figure 10: Temperature Contour(M=1.32)

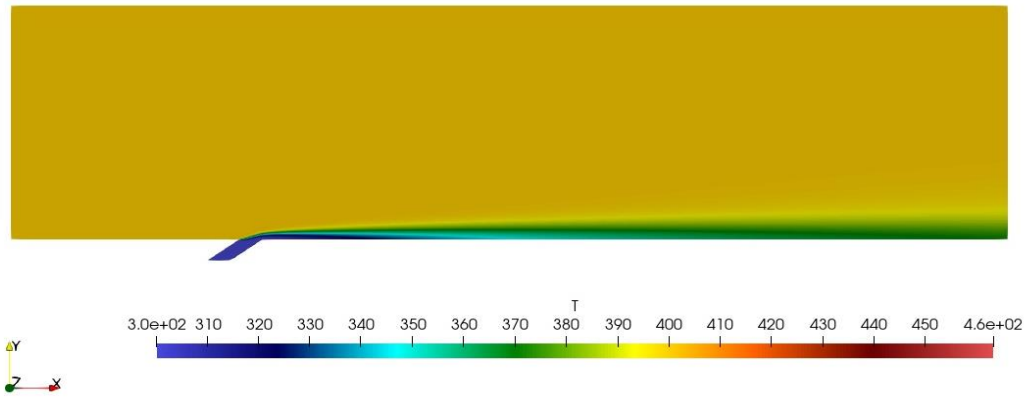


Figure 11 : Temperature Contour(M=0.66)

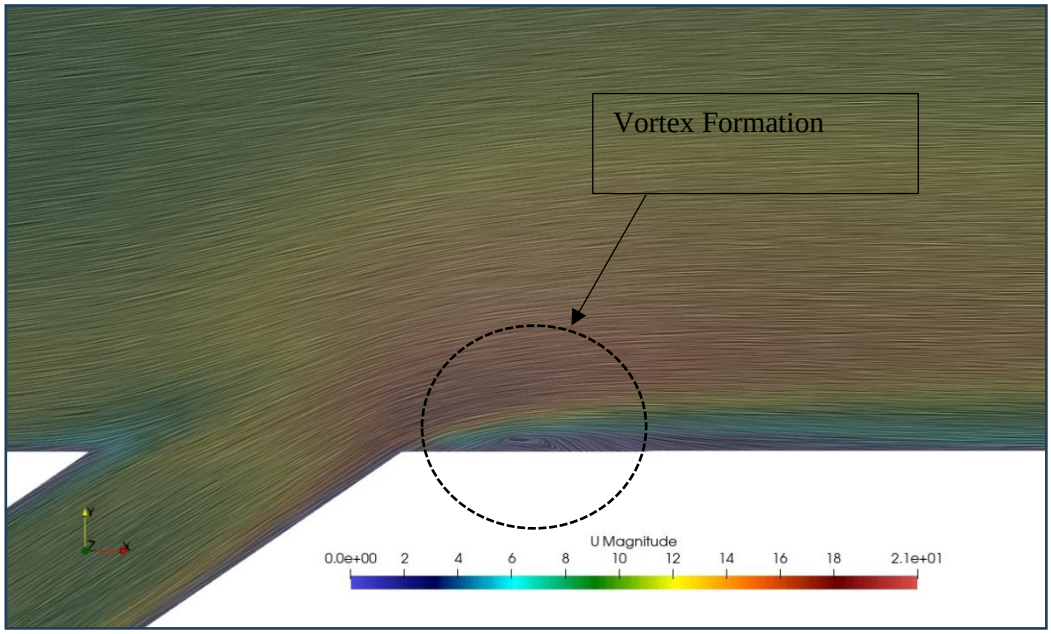


Figure 12: Vortex Formation Near Hole

5.3 Observations

5.3.1. Mechanisms of Vortex Formation Near Cooling Holes

A. **Jet-in-Crossflow Interaction**:
   - When cooling air is injected through holes into a high-speed mainstream flow (the hot gases), it forms a jet-in-crossflow configuration. The interaction between the injected jet and the crossflow creates complex flow structures.
   - The jet, being cooler and often at a different velocity than the mainstream flow, interacts with the boundary layer of the hot gas, leading to the formation of vortices.

B. **Shear Layer Instabilities**:
   - The shear layers at the interface between the injected coolant jet and the mainstream flow can become unstable and roll up into vortices. This is due to the velocity and density differences between the two flows.
   - These shear layer instabilities are a primary cause of vortex formation.

C. **Counter-Rotating Vortex Pair (CRVP)**:
   - A common vortex structure formed near cooling holes is the counter-rotating vortex pair (CRVP). As the coolant jet exits the hole, it tends to roll up into two vortices rotating in opposite directions.
   - The CRVP forms because of the pressure difference around the jet and the interaction with the mainstream flow. These vortices lift the cooling jet away from the surface, which can reduce the cooling effectiveness by mixing the coolant with the hot gases more quickly.

D. **Horseshoe Vortices**:
   - At the leading edge of the cooling hole, the interaction between the boundary layer of the mainstream flow and the protruding jet can create horseshoe vortices. These vortices wrap around the base of the jet and contribute to the complex flow patterns.

E. **Separation and Reattachment**:
   - The jet injection can cause flow separation on the downstream side of the cooling hole, leading to the formation of additional vortices. These

separation vortices can further interact with the CRVP and influence the film cooling performance.

5.3.2. Implications for Film Cooling

A. **Cooling Effectiveness**:
   - The formation of vortices can both enhance and degrade the cooling effectiveness. While vortices can promote mixing and cooling of the hot gases, they can also lift the coolant away from the surface, reducing the protective film's effectiveness.

B. **Thermal Protection**:
   - Understanding the vortex dynamics is crucial for optimizing hole geometry, injection angles, and coolant flow rates to maximize cooling effectiveness and ensure adequate thermal protection.

6. Conclusion

From the above validation between simulation results against the Li &Wang [1] Paper. The OpenFOAM provides a good agreement with Ansys Fluent (commercial software) that was used in the given paper.

- The higher blowing ratio, the more effective of cooling that means more coolant was supplied.
- When increasing main stream temperature, the effectiveness will be reduced.
- OpenFOAM is suitable for research, education as well as industries which provide a reliable result compared to various commercial software
- When increase mainstream temperature from 300K to 500K it shows less changes in effectiveness within (x/2b) <10 length scale, but slightly different in far field. When dealing with high temperature, this solver may not be suitable for accurate result as thermal expansion will affect the density.

Future Concerns Study

In the aforementioned report, the analysis was constrained to the simulation of incompressible and buoyancy-driven flow, a scenario that does not accurately replicate the operational conditions of a gas turbine. To address this limitation, future investigations should employ

multiphase flow solvers, conjugate heat transfer, and models for compressible flow in order to yield more authentic and practical results.

Acknowledgements:

I wish to extend my heartfelt appreciation to **Prof. Sathi Rajesh Reddy** for his unwavering encouragement, support, and invaluable expertise in the realm of Computational Fluid Dynamics (CFD). In addition, I would like to express my gratitude to **Mr. John Pinto**, whose mentorship, substantial assistance, and profound insights into OpenFOAM proved instrumental during the course of my internship. Furthermore, I am profoundly grateful to Prof. **Janani Murallidharan** for her consistent oversight of the students' activities. I also hold deep appreciation for the entire FOSSEE team for affording students the opportunity to exhibit their zeal and make contributions to open source during the internship.

References

[1] Li, X., & Wang, T. (2006). Simulation of film cooling enhancement with mist injection.

[2] Han, J. C. (2013). Fundamental gas turbine heat transfer. *Journal of thermal science and engineering applications*, *5*(2), 021007.

[3] OpenFOAM User Guide v2312

[4] Paraview

[5]. (https://openfoamwiki.net/index.php/BuoyantBoussinesqPisoFoam, n.d.)

DISCLAIMER: This project reproduces the results from an existing work, which has been acknowledged in the report. Any query related to the original work should not be directed to the contributor of this project