Numerical simulation of flow in a fuel-injector of an aircraft engine

○ Younghwa Cho, Hokkaido Univ., Hokkaido, Japan 060-8626, younghwa@eis.hokudai.ac.jp
Rahul Bale, RIKEN-CCS, Kobe, Japan 650-0047, rahul.bale@riken.jp
Chung-Gang Li, Kobe Univ., Japan 657-8501, cgli@aquamarine.kobe-u.ac.jp
Tomonari Satoh, KHI, Kobe, Japan 673-8666, satou_tomonari@khi.co.jp
Ryusuke Matsuyama, KHI, Kobe, Japan 673-8666, matsuyama_ryosuke@khi.co.jp
Takeo Oda, KHI, Kobe, Japan 673-8666, oda_takeo@khi.co.jp
Makoto Tsubokura, RIKEN-CCS, Kobe, Japan 650-0047, mtsubo@riken.jp
Nobuyuki Oshima, Hokkaido Univ., Hokkaido, Japan 060-8626, oshima@eng.hokudai.ac.jp

An investigation of grid sensitivity on local mesh refinement of numerical simulation of cold-flow in a fuel-injector of an aircraft engine is presented in this work. The numerical simulation of fully compressible Navier-Stokes equations is carried out using a hierarchical Cartesian mesh-based solver known as CUBE. With PIV measurement data as the basis, the gird sensitivity analysis is carried out to optimize the region of local mesh refinement in order to minimize the total mesh size. Results of the comparison of simulation results with PIV data and the optimization study is presented.

1. Introduction

Today, high efficiency and low emissions have become essential prerequisites for combustion engines due to global warming and the consequent environmental regulations. Therefore, accurate prediction of the combustion behavior and optimal design of the engine are necessary to improve efficiency and reduce emissions in order to comply with environmental regulations, even more than ever before. In the case of an aircraft engine, a critical variable that governs emission, efficiency, and reliability is the equivalence ratio inside the combustion chamber. This value is transported by swirling turbulent flow from the fuel-injector to the combustion chamber. Therefore, prior to precise prediction and appropriate control of the combustion characteristics of the engine, as a first step, the transport characteristics of flow through the fuel-injector can be investigated through cold flow simulations. A complete evaluation of flow characteristics of the engine requires numerical simulation with the entire annular combustor geometry, which demands massive computational resources. Thus, an optimization strategy to minimize the computational cost through mesh optimization is warranted.

As a first step, this study focuses on the cold flow simulation of a fuel-injector of an aircraft engine. The cold flow simulations were carried out using CUBE **(1)**, abbreviated as Complex Unified Building cubE method. CUBE is a multiphysics simulation software designed for large scale industrial simulations. A hierarchical meshing technique known as building-cube method (BCM) **(2)** is the foundation of CUBE over which numerical solvers are built. Even with the use of BCM for meshing the fuel-injector geometry, using the same near-wall spacing mesh over the entire geometry results in an excessively large numerical mesh. Consequently, relevant geometric components of the injector that strongly influence the flow have to be identified for mesh refinement while coarser mesh can be used for the rest of the geometry. The aim of this study is to investigate the dependence of flow on the numerical mesh and to minimize the total mesh size while ensuring accuracy.

2. Governing equations

The governing equations used in this study are as follows.

$$\frac{\partial \boldsymbol{Q}}{\partial t}+\frac{\partial \boldsymbol{F_1}}{\partial x_1}+\frac{\partial \boldsymbol{F_2}}{\partial x_2}+\frac{\partial \boldsymbol{F_3}}{\partial x_3}=0 \tag{1}$$

where $\boldsymbol{Q}$ is the vector of conservative variables and $\boldsymbol{F_i}$ is the flux vector. Details of these variables are shown below.

$$\boldsymbol{Q}=\begin{pmatrix}\rho\\ \rho u_1\\ \rho u_2\\ \rho u_3\\ \rho e\end{pmatrix} \tag{2}$$

$$\boldsymbol{F_i}=\begin{pmatrix}\rho u_i\\ \rho u_i u_1+p\delta_{i1}-\mu A_{i1}\\ \rho u_i u_2+p\delta_{i2}-\mu A_{i2}\\ \rho u_i u_3+p\delta_{i3}-\mu A_{i3}\\ (\rho e+p)u_i-\mu A_{ij}u_j-\kappa\dfrac{\partial T}{\partial x_i}\end{pmatrix},\quad {}^{\forall}i=1,2,3 \tag{3}$$

where $A_{ij}=\partial u_j/\partial x_i+\partial u_i/\partial x_j-2/3(\partial u_k/\partial x_k)\,\delta_{ij}$ and δ_{ij} is the Kronecker delta.

3. Numerical method

3.1. Complex Unified Building cubE (CUBE)

According to K. Nakahashi **(3)**, demands for next generation CFD for engineering application are as follows:

① *Easy and quick grid generation around complex geometries.*
② *Easy adaptation of local resolution to local flow characteristic length.*
③ *Easy implementation of spatially higher-order schemes.*
④ *Easy massively-parallel computations.*
⑤ *Easy post processing for huge data output.*

The CUBE is a scalable simulation framework developed to answer the above demands. This is based on building-cube method **(2)** to ease computational parallelization and to handle computational load imbalance. To enable simulation of complex geometries such as a fuel-injector, an immersed boundary method developed particularly for compressible flows **(4)** is used in CUBE. The details of CUBE framework can be found in **(1)**. The following sections (**3.1.1** – **3.1.2**) detail the essentials of the methods applied in CUBE and are based on **(4)**.

3.1.1. Building-cube method (BCM)

The BCM is adopted in this study to deal with the imbalance of computational load and to get higher computing performance in a massively parallel environment. In the BCM, the Eulerian domain is discretized using cubic units, named 'cube'. Here, the refinement ratio between fine and coarse cube is fixed at 2. Each cube is further subdivided into cubic cells, and all cubes have the same number of cells. This allows every cube to be equal and independent units of computation. In addition, the cell generation can be individually processed in each cube and be parallelized by OpenMP. In the parallelization process, the cubes are equally allocated to message passing interface (MPI) processors. Therefore, well-balanced computational load between each MPI processor is guaranteed by using the BCM. A schematic representation of the cube and cell generation process in BCM is shown in **Fig. 1**.

The merits of using BCM are as follows: It can contribute to keeping the simplicity of the algorithm minimizing the memory requirement per node and easing implementation of higher-order schemes. In addition, considerable reduction of time for mesh generation can be achieved. As stated above, we can deal with the problem of computational load imbalance, as well.

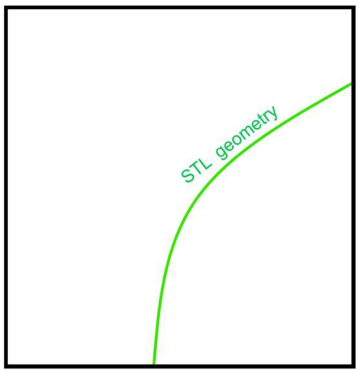
3.1.2. Immersed boundary method (IBM) for compressible flow

The geometry of a fuel-injector of an aircraft engine is considerably sophisticated and the industrial-scale computer-aided design (CAD) data is usually not watertight. If the CAD data is non-watertight and the IBM being used cannot treat non-watertight CAD data, additional time is required for preprocessing the data. The IBM, developed by C.G. Li, M. Tsubokura and R. Bale **(4)**, overcomes this limitation allowing the CAD geometry to be non-watertight eliminating the effort needed for preprocessing.

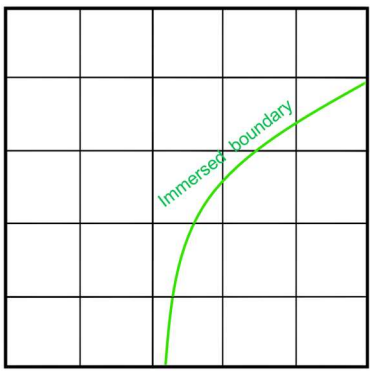
First, the geometry, which is typically in stereolithography (STL) format, is immersed into the BCM mesh. Cells whose centers are with this a distance of one mesh spacing, along principle directions, from the geometry surface are identified as 'interface cell (IC)' and the boundary condition is set up in the ICs. An imaginary 'image point (IP)' is allocated to a point such that IC becomes the midpoint between the IP and the closest point of the geometry surface in normal direction (see **Fig. 2**). This point on wall is defined as the 'wall point'. The flow variables at IP, furthermore, are estimated using bilinear interpolation (trilinear interpolation in 3D) with the corresponding values at the surrounding cells:

$$\varphi_{IP} = w_1\varphi_1 + w_2\varphi_2 + w_3\varphi_3 + w_{IC}\varphi_{IC} \tag{4}$$

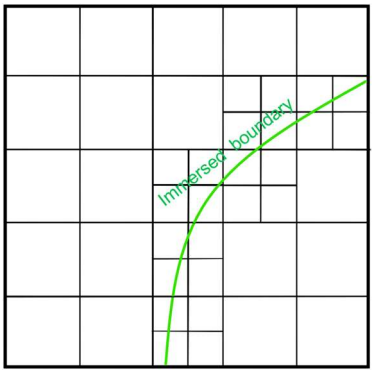
where w_1, w_2, w_3, and w_{IC} are calculated using the VanderMonde matrix which corresponds to the bilinear interpolation



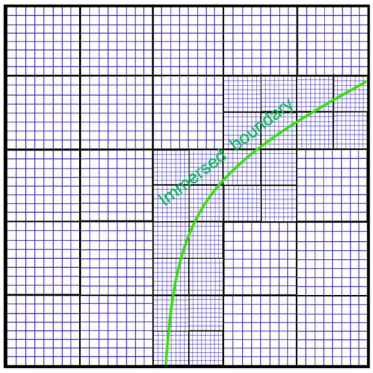
(a) STL geometry on Eulerian domain.



(b) The boundary immersed into BCM cubes (black).



(c) Subdivision of BCM cubes (black) in the domain.



(d) Cell (blue) generation in each BCM cube (black).

Fig. 1. A schematic of mesh generation process in BCM: (a) to (d).

scheme suggested by R. Ghias et al. (2007). Furthermore, according to M.D. de Tullio et al. (2007), φ_{IC} can be evaluated using linear interpolation between the IC and the wall value φ_w:

$$\varphi_w = \varphi_{IC} - \left(\frac{\partial \varphi}{\partial n}\right) d \tag{5}$$

where d is the distance from the IC to the wall.

Due to $\partial\varphi/\partial n = (\varphi_{IP} - \varphi_w)/(2d)$, Eq. (5) becomes,

$$\varphi_w = \varphi_{IC} - \frac{\varphi_{IP} - \varphi_w}{2} \;\leftrightarrow\; \varphi_{IC} = \frac{1}{2}(\varphi_{IP} + \varphi_w) \tag{6}$$

For the Dirichlet condition, φ_w is an assigned value φ_{assign}, i.e. $\varphi_w = \varphi_{assign}$ and for the Neumann condition, $\varphi_w = \varphi_{IP}$. Thus, we can represent Eq. (6) for each condition as follows.

$$\varphi_{IC} = \frac{w_1\varphi_1 + w_2\varphi_2 + w_3\varphi_3 + \varphi_{assign}}{2} \quad \text{Dirichlet} \tag{7}$$

$$\varphi_{IC} = \frac{w_1\varphi_1 + w_2\varphi_2 + w_3\varphi_3}{1 - w_{IC} + \varepsilon} \quad \text{Neumann} \tag{8}$$

When IC is at the wall or very close to the wall, w_{IC} can attain value very close to 1 or equal 1. This can lead to numerical divergence of Eq. (8). Therefore, ε, set to 1×10^{-6}, is added to the denominator of Eq. (8) in order to prevent numerical divergence. In the present study, the no-slip condition and the adiabatic condition are applied to the wall, i.e. surface of the geometry. Here, Eq. (7) is adopted for the no-slip condition. Meanwhile, for the adiabatic condition and pressure, Eq. (8) is employed.

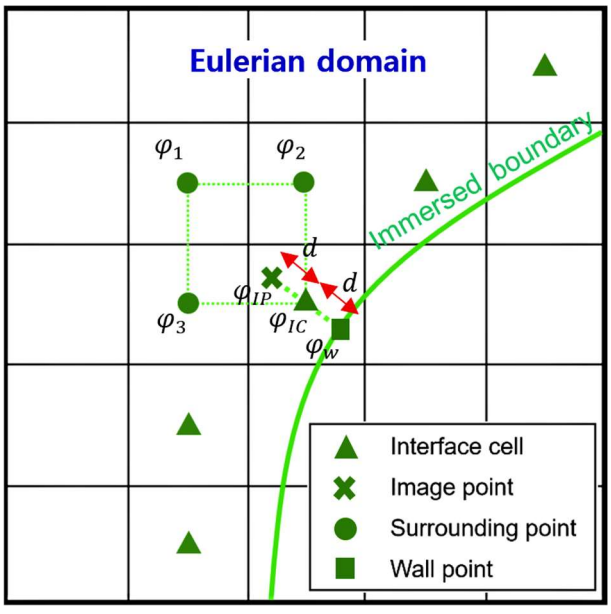


Fig. 2. Details of the immersed boundary method used in this work.

3.2. Numerical schemes

Table 1 shows the numerical schemes used in this work. One can find the details of the solution-limited time stepping method in (5) and those of the low-Mach number fix for the Roe scheme in (6).

Table 1. Numerical schemes used in this work.

Time marching	Lower-upper symmetric Gauss-Seidel with solution-limited time stepping method
Flux	Roe with low-Mach number fix
Diffusion	2nd-order central difference

4. Validation of numerical simulation

4.1. 2D PIV data of Kawasaki Heavy Industries (KHI)

A 2D PIV measurement of air flow in a fuel-injector was conducted. The detail of the measurement is as follows. Table 2 specifies the inflow condition and ambient conditions. The experimental setup and the measurement window are shown in Fig. 3. One can find the result of this measurement in section 4.3.

Table 2. Measurement conditions.

Inflow mass flow rate	0.072 [kg/s]
Ambient pressure	101,300 [Pa]
Ambient temperature	288 [K]

4.2. Numerical validation setup

As a first step, in order to validate the CUBE and the IBM, a very high-resolution mesh was used to carry out the cold flow simulation. The geometry (STL format) used in this work is shown in Fig. 4. The boundary condition is shown in Fig. 5, and the mesh is shown in Fig. 6. Details of the simulation conditions are specified in Table 3. In Table 3, wall boundary condition means the boundary condition on the surface of the geometry. The results of this simulation are discussed in section 4.3.

Table 3. Simulation conditions.

Simulation type	Cold flow simulation
Wall boundary condition	No-slip and adiabatic condition
Mesh size (num. of cells)	~500,320,320
Mesh resolution (dx)	0.05 [mm]
Time resolution (dt)	0.0001 [s]
Inflow mass flow rate	0.072 [kg/s]
Inflow velocity	5.23 [m/s]
Ambient pressure	101,300 [Pa]
Ambient temperature	288 [K]
Inflow duct area	0.106×0.106 [m^2]
Total simulation time	2.7 [s]
Time averaging duration	1.7 [s]
Wall time	48 [hr]
CPU used (nodes×cores)	2048×8
Computer used	K computer

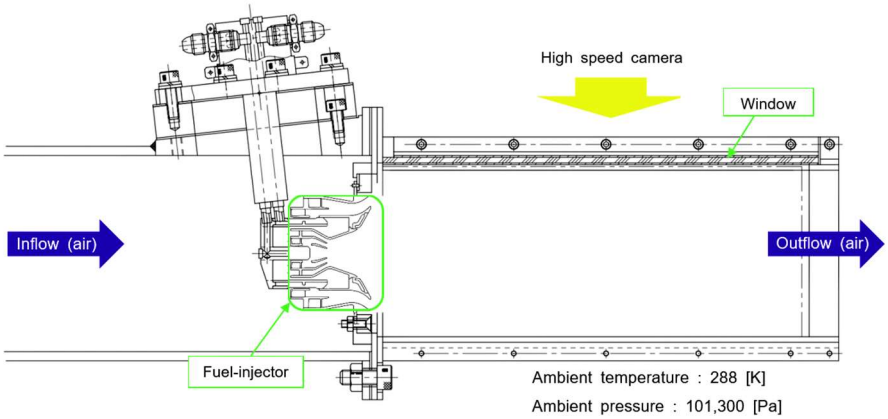
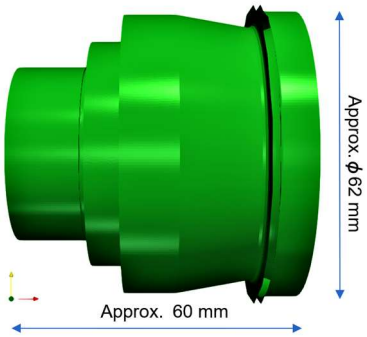
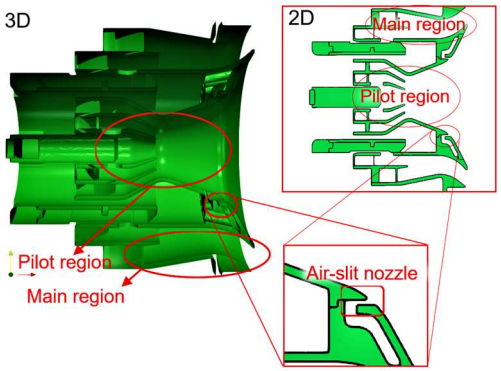


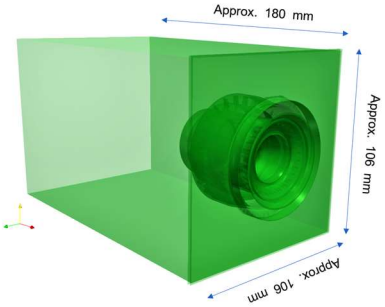
Fig. 3. PIV measurement setup.



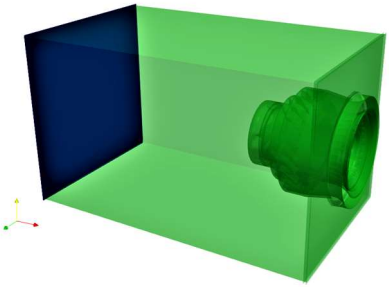
(a) Fuel-injector geometry.



(b) Fuel-injector geometry (cross section).



(c) Fuel-injector with inflow-duct.



(d) Inflow-plane (blue).

Fig. 4. The geometry used in this work

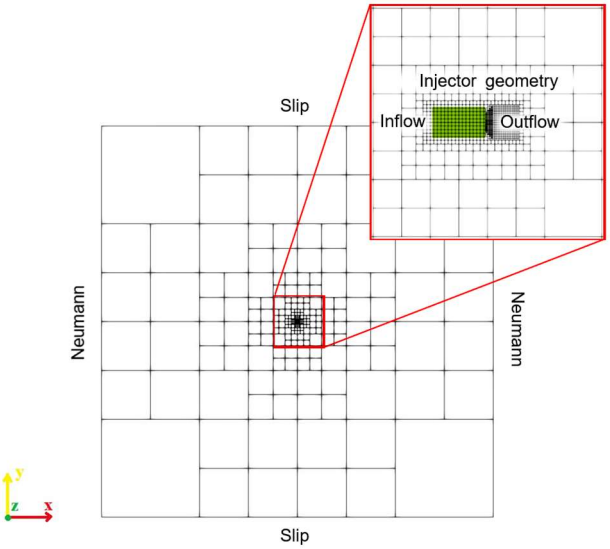


Fig. 5. Boundary condition.

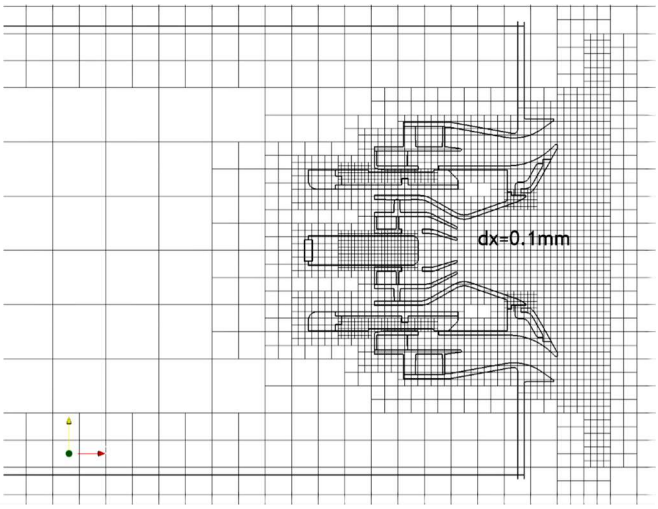


Fig. 6. Validation mesh (cubes).

4.3. Validation results and discussion

Velocity magnitude contours on a mid-sectional plane from the PIV measurement and the numerical simulation are shown in **Fig. 7**. Four distinct identifiable flow signatures are identified in **Fig. 7-(c)**. The flow signatures are termed as **A, B, C** and **D** in the PIV result, and **A', B', C'** and **D'** in the simulation result.

First, **A** (and **A'**) is the primary jet emanating out of the injector. **A'** shows greater velocity magnitude than **A**. In both cases, the flow gradually decelerates as it heads downstream.

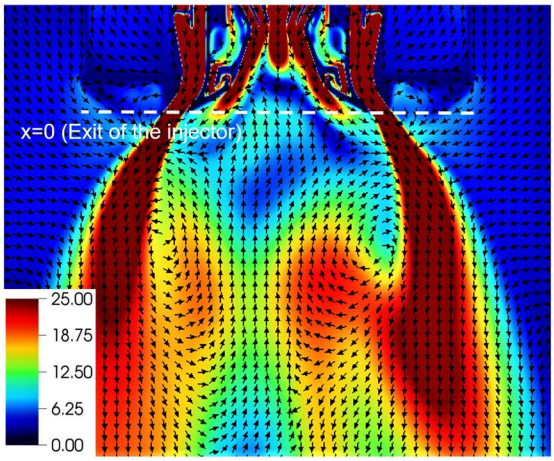
The flow signature **B** (and **B'**) is the one formed by the flows from the pilot region and air-silt nozzle. It is observed that **B** is separated from **A** and forms vortex. On the other hand, **B'** does not form any vortex but just joins into **A'**, supplying momentum to **A'**. It is hypothesized that intrinsically, **B'** is just sucked into **A'** because of the large velocity magnitude in the exit-vicinity region of **A'**.

Meanwhile, **C'** is more widely distributed although **C** is localized in narrow region. It is regarded that the absence of vortex in **B** and the difference in intensity of primary jet (**A-A'**) may affect the distribution. Furthermore, the velocity magnitude of the flow in **C'** is greater than the flow in **C**, and it is likely that this is due to the difference in intensity of the primary jet, i.e. **A-A'**, which induces **C-C'**.

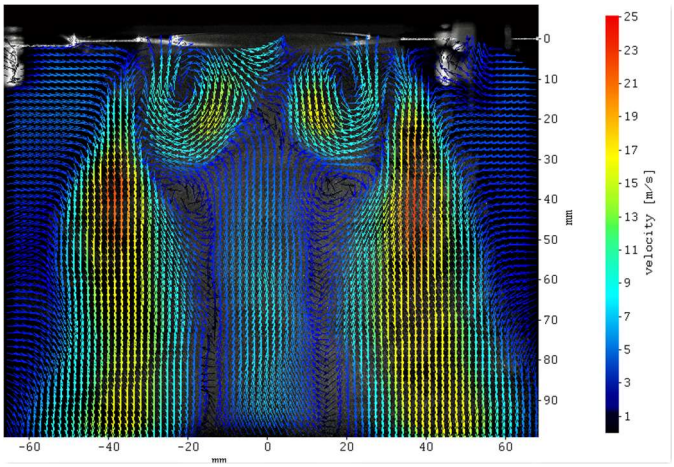
Finally, **D'** stretches longer than **D** up to the exit of the injector. It can be considered that the greater intensity in the region **C'** and the absence of vortex in **B'** assist the flow in **D'** stretching it longer than its PIV counterpart **D**.

To sum up, it is regarded that the local differences are mainly affected by the difference between **B** and **B'** induced by overestimation of **A'**. Therefore, this is the sole qualitative difference between the simulation and the PIV data. It can be inferred that the simulation qualitatively resolves the injector flow.

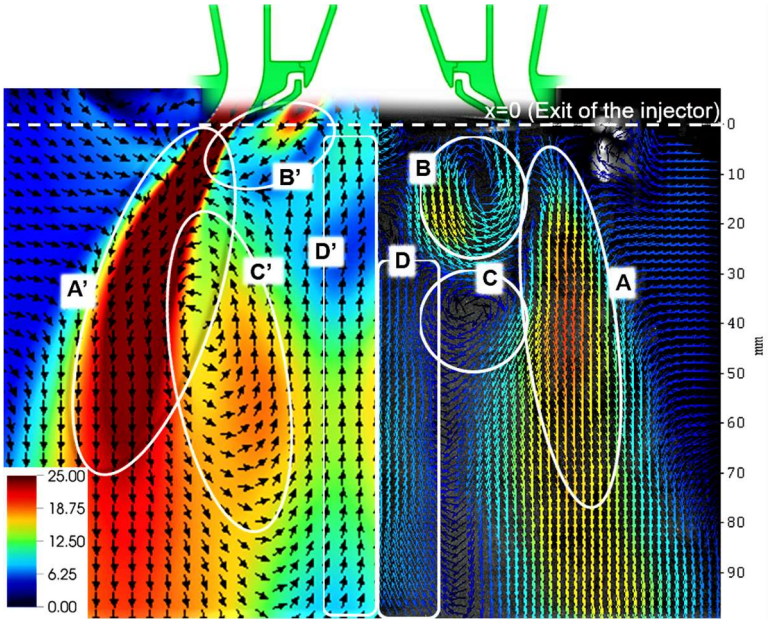
Consequently, the high-resolution simulation data presented in this section will be used as the base (**(III)** Validation mesh in **Table 4**) for the mesh optimization study in section **5**.



(a) Time averaged velocity magnitude
High-resolution simulation.



(b) Time averaged velocity magnitude
2D PIV data.



(c) A comparision of the simulation [left: (a)] and PIV [right: (b)] results: 4 prominent features can be identified in eqch case.

Fig. 7. A qualitative comparison of the simulation and the 2D PIV data (KHI).

Table 4. Simulation conditions of the three meshes considered in this work.

	Optimization meshes		(III) Validation mesh
	(I) Reduced mesh 1	(II) Reduced mesh 2	
Simulation type	Cold flow simulation		
Wall boundary condition	No-slip and adiabatic condition		
Mesh size (num. of cells)	**~25,000,000**	**~120,000,000**	**~500,320,320**
Mesh resolution (dx)	**0.095 [mm]**	**0.076 [mm]**	**0.05 [mm]**
Mesh-generation Time	**~30 [min]**		
Time resolution (dt)	0.0001 [s]		
Inflow mass flow rate	0.072 [kg/s]		
Inflow velocity	5.23 [m/s]		
Ambient pressure	101,300 [Pa]		
Ambient temperature	288 [K]		
Inflow duct area	0.106×0.106 [m^2]		
Total simulation time	0.2 [s]	0.06 [s]	2.7 [s]
Time averaging duration	0.05 [s]	0.02 [s]	1.7 [s]
Wall time	120 [hr]	153 [hr]	48 [hr]
CPU used (nodes × cores)	288×1		2048×8
Computer used	ITO computer		K computer

5. Mesh optimization

5.1. Optimization setup

For the optimization study, we consider two meshes – **(I)** Reduced mesh 1: smallest mesh spacing of 0.095 [mm] and mesh size of 25M, and **(II)** Reduced mesh 2: smallest mesh spacing of 0.076 [mm] and mesh size of 120M. The details of these two meshes are tabulated in **Table 4**. One remarkable thing is that there is little difference in mesh-generation time between **(I)** – **(III)**, even though there is a considerable deviation of mesh size. The boundary condition of the computational domain remains the same as the one shown in **Fig. 5**, and the geometry is as shown in **Fig. 4**. As shown in **Table 4**, 288 CPUs were used in ITO computer. It was figured out that the steady-state is generally reached at the 200th time step, i.e., 0.02 [s] in simulation time.

5.2. Results and discussion

As shown in **Table 4**, the computation speed is nearly inversely proportional to the mesh size. Using the definition of computation speed in the equation below, we compare the computation speed of **(I)** and **(II)**,

$$\text{Computation speed} = \frac{\text{Simulation Time}}{\text{Wall Time}}$$

Table 5. Comparison of computation speed.

(I) Reduced mesh 1	$\sim 1.7 \times 10^{-3}$
(II) Reduced mesh 2	$\sim 3.9 \times 10^{-4}$
(I) / (II)	**4.3**

It is clear that increasing the spatial resolution 4.8 times requires 4.3 times greater computation cost, assuming that the same number of CPUs and the same computer is used (see **Table 5**). This reminds the necessity of this investigation for future works, as well.

*5.2.1. The result of **(I)** (Reduced mesh 1) and discussion*

Fig. 8 shows the comparison of the meshes of **(I)** and **(III)**. The finest resolution of dx =0.095 [mm] is assigned in the narrowest parts represented by the air-silt nozzle. This make **(I)** about 2 times coarser than **(III)** Validation mesh (see **Fig. 8**).

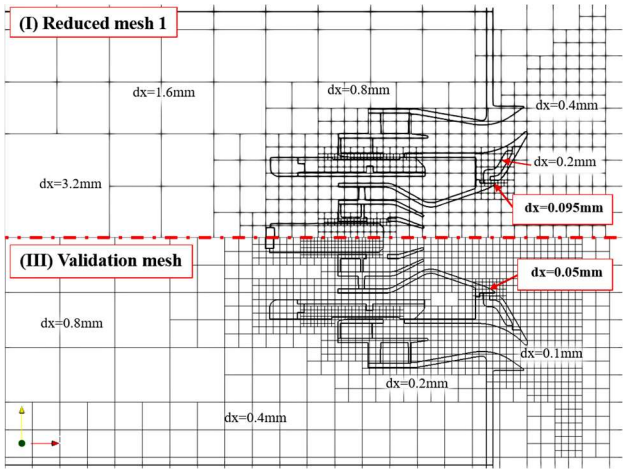


Fig. 8. Comparison of Reduced mesh 1 and Validation mesh.

The result of the simulation **(I)** is shown in **Fig. 9**. It shows that although **(I)** adopted 2 times coarser mesh, it was able to resolve the flow through the air-slit nozzle. However, in the region **E-E'**, a noticeable difference between **(I)** and **(III)** is seen. In the case of **(III)**, the flow from the pilot region, i.e. **E**, is attached to the wall. However, in **(I)**, the flow, i.e. **E'**, is developed as jet flow, separating from the wall. Therefore, it appears that the region **E-E'** is a very sensitive region near the exit of the nozzle that significantly influence the flow.

Another difference in flow between **(I)** and **(III)** is the difference of velocity magnitude in the region **F-F'** and **G-G'**. It is evident that the velocity magnitude in regions **F'** and **G'** is lower than its counterpart in **(III)**. As a consequence of the low magnitude of velocity, it can be said that the swirl in **(I)** is not fully developed.

In summary, **(I)** was successfully able to resolve all flow-paths in the geometry. However, the result is not satisfactory because the flow in region **E'** is does not qualitatively agree with the flow in region **E**. As a result of this, it is derived that the pilot region shows the highest dependency upon resolution change. That is, a resolution, with which the pilot region becomes qualitatively consistent with **E**, is the lower bound of qualitative resolution of this injector flow.

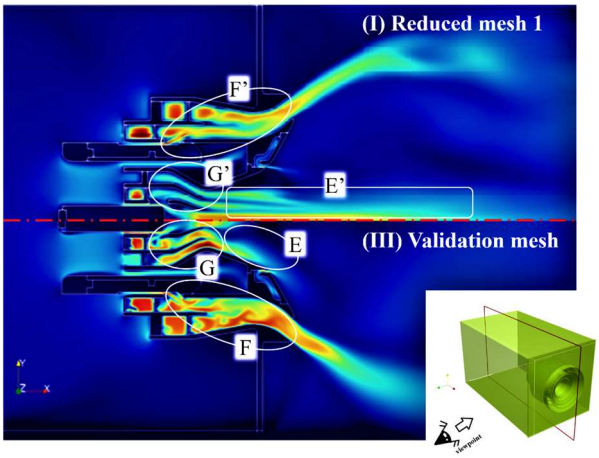


Fig. 9. Velocity magnitude of Reduced mesh 1 compared with Validation mesh.

*5.2.2. The result of **(II)** (Reduced mesh 2) and discussion*

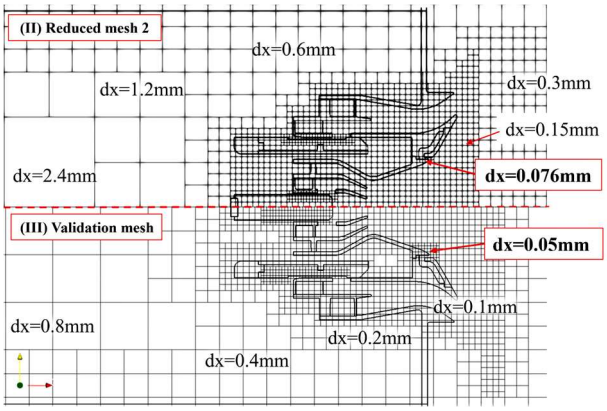


Fig. 10. Reduced mesh 2 compared with Validation mesh.

In **Fig. 10**, a comparison of the meshes of **(II)** and **(III)** is shown. The resolution of the air-slit nozzle region is chosen to be 0.076 [mm] in **(II)**. As it is evident from the analysis of the result of **(I)**, the pilot region is highly sensitive to resolution. Therefore, for the present mesh, a mesh spacing of 0.15 [mm] is chosen for the pilot region which was 0.4 [mm] in the mesh of **(I)**.

The result of the simulation of **(II)** is shown in **Fig.11**. It is evident from the result that the mesh of **(II)** was successfully able to capture the flow attached to the wall in the pilot region (**E"**).

Finally, the distribution of velocity magnitude in the region **F"** and **G"** is also considerably consistent with its counterpart in **(III)**.

Based on the results of our initial trial mesh of **(I)**, we were able to identify the key regions that strongly influence the flow and accordingly generate a new trial mesh in **(II)**. As a result, **(II)** derived a result consistent with **(III)**. Therefore, the overall injector flow was consequently resolved with the mesh of **(II)** with reasonable degree of success.

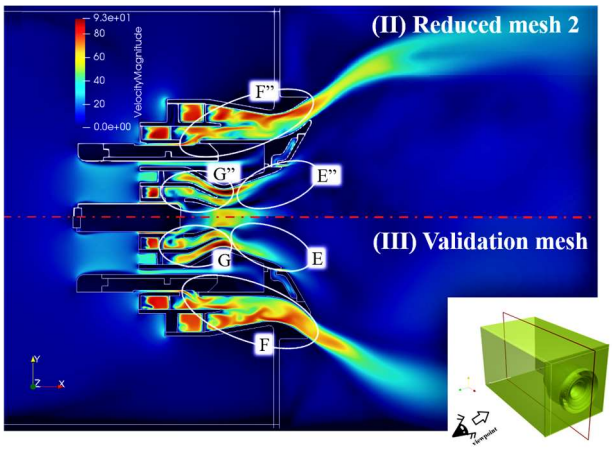


Fig. 11. The prominent flow features of Reduced mesh 2 qualitatively agree with those of Validation mesh.

6. Summary and conclusion

The time-averaged 2D PIV data of KHI was utilized to validate CUBE using high-resolution simulation. As a result of this, the simulation can be regarded that it is qualitatively consistent with the real phenomena, i.e., PIV data. Therefore, the simulation result was adopted as the criterion for the mesh optimization study.

For the optimization process, two simulations with different spatial resolutions were conducted in order to investigate the optimal mesh using CUBE. It was found that the mesh generation time was insensitive to mesh resolution and that the total mesh generation time was about 30 min for all the meshes reported in this work.

In the first trial, i.e. **(I)** (Reduced mesh 1), it was clearly shown that all flow-paths can be resolved with the resolution of $dx = 0.095$ [mm] and that the pilot region is the most resolution-dependent part in the injector flow. Thus, it was expected that the overall flow could be resolved if sufficient mesh resolution is assigned to the pilot region.

In **(II)** (Reduced mesh 2), the pilot region was resolved with the resolution of $dx = 0.076$ [mm]. As a result, the flow in the pilot region improved considerably. Furthermore, the overall distribution of the flow field showed reasonable agreement with the result of **(III)**. In short, it is clarified that **(II)** (Reduced mesh 2) can qualitatively resolve the injector flow. Meanwhile, **(II)** (Reduced mesh 2) has the spatial resolution, which is slightly finer than that of **(I)** (Reduced mesh 1) and slightly coarser than **(III)** (Validation mesh).

Therefore, the resolution of **(II)** (Reduced mesh 2) can be regarded as the optimal mesh for the injector flow of the meshes we considered. There is perhaps scope for further optimization by balancing the best of mesh **(I)** and **(III)**. This can be part of a future study.

Acknowledgement

This research was partially supported by 'High-Performance Computing Infrastructure (HPCI) project (hp190049)'.

Bibliography

(1) N. Jansson, R. Bale, K. Onishi, M. Tsubokura, CUBE: A scalable framework for large-scale industrial simulations, International Journal of High Performance Computing Applications Vol.33 (2018): 678-698.
(2) K. Nakahashi, Building-cube method for flow problems with broadband characteristic length, Computational Fluid Dynamics 2002, Springer, Berlin, Heidelberg (2003): 77-81.
(3) K. Nakahashi, Building-Cube Method; A CFD Approach for Near-Future PetaFlops Computers, 8th World Congress on Computational Mechanics (WCCM8), Venice Italy (2008).
(4) C.G. Li, M. Tsubokura, R. Bale, Framework for simulation of natural convection in practical applications, International Communications in Heat and Mass Transfer Vol.75 (2016): 52-58.
(5) C. Lian, G. Xia, C.L. Merkle, Solution-limited time stepping to enhance reliability in CFD applications, Journal of Computational Physics Vol.228 (2009): 4836-4857.
(6) F. Rieper, A low-Mach number fix for Roe's approximate Riemann solver, Journal of Computational Physics Vol.230 (2011): 5263-5287.
(7) T. Ishida, S. Takahashi, K. Nakahashi, Fast cartesian mesh generation for building-cube method using multi-core pc, 46th AIAA Aerospace Sciences Meeting and Exhibit (2008).
(8) K. Nakahashi, High-density mesh flow computations with pre-/post-data compressions, 17th AIAA Computational Fluid Dynamics Conference (2005).