International Journal of Aeronautical and Space Sciences (2021) 22:489–501
https://doi.org/10.1007/s42405-020-00337-9

IJASS
International Journal of Aeronautical and Space Sciences

ORIGINAL PAPER

Enhanced Pressure Based Coupled Algorithm to Combine with Pressure–Velocity-Enthalpy for all Mach Number Flow

J. R. Shin[1] · T. W. Kim[2]

Received: 11 June 2020 / Revised: 23 September 2020 / Accepted: 12 November 2020 / Published online: 4 January 2021

Abstract

In this paper, a pressure-based coupled computational fluid dynamics algorithm for numerical analysis of all Mach number region flow is developed. For this purpose, an enhanced pressure based coupled algorithm was developed through the pressure–velocity coupled algorithm and the pressure-enthalpy coupled algorithm were combined. In additional, the Kurganov–Tadmor flux splitting scheme, which is mainly used in density-based solvers, was applied to a developed pressure-based coupled solver. To confirm the analytical ability of developed solver, the variety of Mach number flow problems were performed using the developed solver. It was confirmed that the developed solver had the similar analytical ability with that of the other numerical codes through the analysis of the shock tube problems. In order to verify the analytical ability for the variety Mach number flow region of the developed solver, 2D bump and nozzle problems and 3D missile and wing problems were analyzed and compared with results of experiments and other numerical analysis codes. It is confirmed that the analytical ability of developed solver in the all speed flow region is somewhat improved than the commercial analysis package and is similar to the density based in-house CFD code.

Keywords Finite volume method · Pressure-based solver · Coupled algorithm · Numerical method · OpenFOAM

1 Introduction

Pressure-based computational fluid dynamics analysis is applied as a major analysis method in most commercial analysis packages, and is widely used in research and industries because it is appropriate for analyzing incompressible flow or applying various physical models. In pressure-based computational fluid dynamics, the pressure–velocity coupled algorithm is used to convert a continuity equation to a pressure equation. The Rhie–Chow interpolation scheme is one of the most widely used pressure–velocity coupling methods [1].

The Rhie–Chow interpolation scheme converts a continuity equation to a pressure-based equation using the velocity and pressure values in the intermediate step, which are calculated in the momentum equation using the pressure value of the previous step in a pressure-based segregated analysis algorithm, and determines the velocity and pressure values of the final step through an analysis of this pressure-based equation. This segregated algorithm is widely used because it can save the computation resources and is highly efficient for analysis of an incompressible region with little change in density.

In spite of these advantages, there are limitations in applying this analysis algorithm to compressible flow with large variations in density, flow inside a combustor with sudden changes in pressure, and flow of expanding high-pressure gas. Various studies are being conducted to improve these limitations.

Kim and Gill [2] and Kraposhin et al. [3] analyzed a flow phenomenon with shock waves by applying flux splitting, which is used in a density-based solver, to the pressure-based segregated algorithm. Darwish and Moukalled [4] proposed the pressure-based pressure–velocity coupled algorithm and confirmed that the convergence of the analysis improves at various Mach number. Mangani et al. [5] applied the pressure–velocity coupled algorithm in a steady state to the OpenFOAM, which is an open-source-based computational fluid dynamics platform. Mangani et al. [6] had improved their previous research to analyze incompressible transient flows.

✉ T. W. Kim
truepray@sdu.ac.kr

[1] R&D Center, NEXTfoam Co., LTD., Seoul, South Korea

[2] Department of Mechanical Engineering, Seoul Digital University, Seoul, South Korea

An analysis that includes the energy conservation equation in this pressure–velocity coupled algorithm has the disadvantage that the variation of the continuity equation by enthalpy must be corrected by performing internal iterative calculations. To efficiently calculate the variations of the continuity equation by enthalpy, Emans et al. [7, 8] proposed an algorithm that analyzes flows with large variations in pressure and enthalpy by coupling pressure and enthalpy. Xiao, Denner and co-researchers [9, 10] had developed a fully-coupled algorithm which combines pressure–velocity–enthalpy in a single matrix.

This study developed an algorithm that can robustly analyze flows with large variations in pressure and enthalpy such as compressible flow by combining the pressure–enthalpy coupled algorithm with the pressure–velocity coupled algorithm proposed in previous studies. Furthermore, the flux splitting method for analyzing the discontinuous phenomenon of flows such as shock waves was applied. Then, a solver was developed by applying the above analysis algorithm to the extended version of OpenFOAM, which is an open source-based computational fluid dynamics analysis platform, and applied to various discontinuous flow phenomena cases. This study verified the possibility of analysis by applying the developed algorithm to various shock wave phenomena, rather than verifying the convergence of the developed algorithm.

2 Analysis Method

2.1 Pressure–Velocity Coupled Algorithm

The pressure-based coupled algorithm that combines velocity and pressure was researched by Darwish and Moukalled [4]. This method acquires the velocity and pressure values by simultaneously updating the pressure gradient of the momentum equation and the flux value of the continuity equation. The following shows a pressure–velocity coupled matrix equation:

$$\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}, \qquad (1)$$

where A is a matrix related to convection and diffusion, and B and B^T are arranges about gradient and divergence. Figure 1 shows a flowchart of the pressure–velocity coupled algorithm.

2.1.1 Discretization of Momentum Equation

The momentum equation of the finite volume method (FVM) that is discretized using the Rhie–Chow interpolation scheme can be expressed as follows:

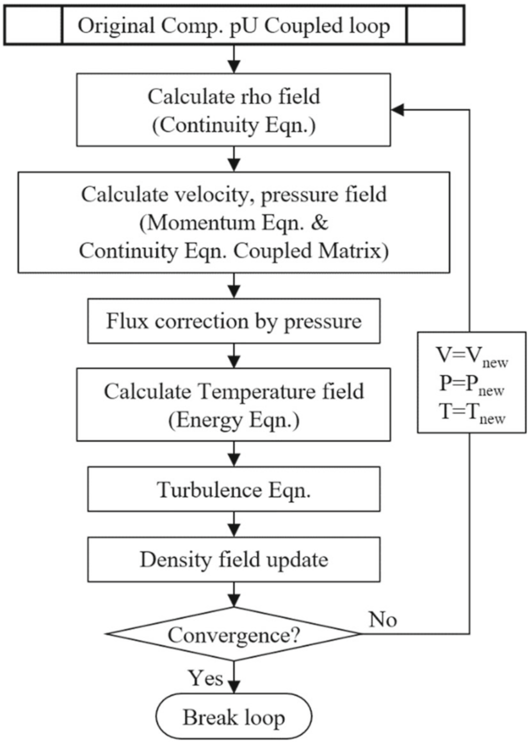
Original Comp. pU Coupled loop
Calculate rho field
(Continuity Eqn.)
Calculate velocity, pressure field
(Momentum Eqn. &
Continuity Eqn. Coupled Matrix)
Flux correction by pressure
Calculate Temperature field
(Energy Eqn.)
Turbulence Eqn.
Density field update
Convergence?
No
Yes
Break loop
V=V_new
P=P_new
T=T_new

Fig. 1 Original pressure based p–U coupled algorithm

$$a_C \boldsymbol{u}_C + \sum_{f=nb(C)} a_f \boldsymbol{u}_f + V_C \nabla p_c = \boldsymbol{b}_C^{\boldsymbol{u}}, \qquad (2)$$

where f is face value and c is cell center value. The above equation can be simplified as follows:

$$\boldsymbol{u}_C + \boldsymbol{H}_C(u) + D_C \nabla p_c = \widetilde{\boldsymbol{b}_C^{\boldsymbol{u}}}. \qquad (3)$$

2.1.2 Discretization of Continuity Equation

The FVM-type discretization equation of the continuity equation can be expressed as follows:

$$\frac{\rho_C - \rho_C^0}{\Delta t} + \sum_{f=nb(C)} \rho_f \boldsymbol{u}_f \cdot S_f = 0. \qquad (4)$$

By applying the Newtonian linearization to the $\rho_f \boldsymbol{u}_f$ term in this equation, it can be changed as follows:

$$\rho_f \boldsymbol{u}_f = \rho_f^n \boldsymbol{u}_f + \rho_f \boldsymbol{u}_f^n - \rho_f^n \boldsymbol{u}_f^n. \qquad (5)$$

When the above equation is substituted in Eq. (4) and the newly calculated density value is replaced with a pressure value, the following equation is obtained:

$$\frac{C_p p_C - \rho_C^0}{\Delta t} V_C + \sum_{f=nb(C)} \left(\rho_f^n \boldsymbol{u}_f \cdot S_f + \rho_f \boldsymbol{u}_f^n \cdot S_f - \rho_f^n \boldsymbol{u}_f^n \cdot S_f \right) = 0, \tag{6}$$

where C_p is compressibility. Compressibility rewrite as $1/RT$.

The newly calculated $\boldsymbol{u}_f$ can be expressed as follows using the Rhie–Chow interpolation scheme and Eq. (3): Where overbar is interpolation value.

$$\boldsymbol{u}_f = \overline{\boldsymbol{u}_f} - \overline{\boldsymbol{D}}_f \left(\nabla p_f - \overline{\nabla p_f^n} \right). \tag{7}$$

By substituting Eq. (7) in Eq. (6), the continuity equation in which velocity is implicitly applied can be expressed as follows:

$$\begin{aligned} &\frac{C_p V_C}{\Delta t} p_C + \sum_{f=nb(C)} \left(C_p p_C \boldsymbol{u}_f^n \cdot S_f \right) \\ &+ \sum_{f=nb(C)} \left(\rho_f^n \boldsymbol{u}_f - \rho_f^n \overline{\boldsymbol{D}_f} \nabla p_f \right) \cdot S_f \\ &= \frac{V_C}{\Delta t} \rho_C^0 \\ &+ \sum_{f=nb(C)} \left(\rho_f^n \boldsymbol{u}_f^n \cdot S_f \right) - \sum_{f=nb(C)} \left(\rho_f^n \overline{\boldsymbol{D}_f} \overline{\nabla p_f^n} \cdot S_f \right). \end{aligned} \tag{8}$$

When the pressure term in Eq. (3) and the velocity term in Eq. (8) are implicitly processed and expressed as one matrix equation as shown below. The elements of the matrix are generally composed of coefficients derived by discretizing the convection, diffusion, and divergence terms.

$$\begin{bmatrix} \boldsymbol{a}_C^{\boldsymbol{uu}} & a_C^{\boldsymbol{u}p} \\ \boldsymbol{a}_C^{\boldsymbol{pu}} & a_C^{pp} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_C \\ p_C \end{bmatrix} + \sum_{f=nb(C)} \begin{bmatrix} \boldsymbol{a}_f^{\boldsymbol{uu}} & a_f^{\boldsymbol{u}p} \\ \boldsymbol{a}_f^{\boldsymbol{pu}} & a_f^{pp} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_f \\ p_f \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_C^{\boldsymbol{u}} \\ b_C^p \end{bmatrix}. \tag{9}$$

2.2 Pressure–Enthalpy Coupled Algorithm

Unlike the above-mentioned pressure–velocity coupled algorithm, the pressure–enthalpy coupled algorithm is appropriate for flows with large variations in pressure and enthalpy. In case of typical internal combustion or high-temperature and high-velocity flows, large variations in internal energy appear together with variations in velocity. Therefore, a change of the continuity equation by internal energy needs to be predicted. Emans et al. [6, 7] introduced a method of implicitly coupling pressure and enthalpy, which represent internal energy. Figure 2 shows an illustration of this coupled algorithm. This uses a general SIMPLE algorithm for obtaining a intermediate velocity from the momentum equation, after which the continuity equation and the energy equation are coupled.

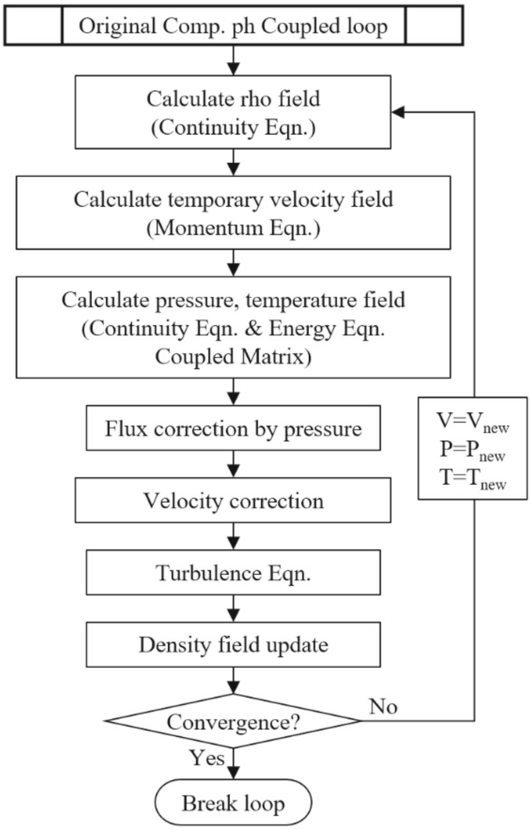

Fig. 2 Original pressure based p–h coupled algorithm

2.2.1 Modification of Continuity Equation

For the continuity equation in the pressure–enthalpy coupled algorithm, the change of density from preface gas is represented by a change of pressure and enthalpy as in Eq. (10) where ρ is $C_p p$ and k is $-p/c_p T$.

$$\mathrm{d}\rho = \left.\frac{\partial \rho}{\partial T}\right|_p \mathrm{d}T + \left.\frac{\partial \rho}{\partial p}\right|_T \mathrm{d}p = C_p \mathrm{d}p + k\mathrm{d}h, \tag{10}$$

$$\frac{\left(C_p p_C + kh_c - kh_C^n \right) - \rho_C^0}{\Delta t} + \sum_{f=nb(C)} \left(C_p p_f + kh_f - kh_f^n \right) \boldsymbol{u}_f^* \cdot S_f = 0. \tag{11}$$

Equation (11) is a modification of the continuity equation to implicitly consider enthalpy, where $\boldsymbol{u}_f^*$ is an intermediate velocity calculated by the momentum equation. The continuity equation is obtained when the value at the calculation cell face through the Rhie–Chow interpolation scheme.

2.2.2 Modification of Energy Equation

The basic equation of the enthalpy-based energy equation is as follows:

$$\frac{\partial(\rho h)}{\partial t}+\nabla\cdot(\rho\boldsymbol{u}h)+\frac{\partial\rho K}{\partial t}+\nabla\cdot(\rho\boldsymbol{u}K)-\frac{\partial p}{\partial t}=-\nabla\cdot q+\nabla\cdot(\tau\cdot\boldsymbol{u})+\rho r+q_h, \tag{12}$$

where K denotes $0.5\boldsymbol{u}^2$, which is calculated explicitly and the source or radiation terms is considered as needed. For the conventional energy equation, the pressure change term over time is calculated explicitly. However, the pressure–enthalpy coupled algorithm processes this term implicitly and uses the pressure calculated in the continuity equation.

When the continuity equation and energy equation are discretized in accordance with each term, they are expressed as a 2×2 matrix as shown in Eq. (13). The pressure and enthalpy values are calculated using this matrix equation. The element terms of the matrix consist of coefficients generally derived by discretizing convection, diffusion, and divergence terms.

$$\begin{bmatrix} a_C^{pp} & a_C^{ph} \\ a_C^{hp} & a_C^{hh} \end{bmatrix}\begin{bmatrix} p_C \\ h_C \end{bmatrix}+\sum_{f=nb(C)}\begin{bmatrix} a_f^{pp} & a_f^{ph} \\ a_f^{hp} & a_f^{hh} \end{bmatrix}\begin{bmatrix} p_f \\ h_f \end{bmatrix}=\begin{bmatrix} b_C^p \\ b_C^h \end{bmatrix}. \tag{13}$$

2.3 Pressure–Velocity–Enthalpy Coupled Algorithm

The above-mentioned two coupled algorithms modify each equation so that the momentum equation or energy equation is calculated simultaneously with the continuity equation. These coupled algorithms have disadvantages because divergence of computation occurs when analyzing flows with large simultaneous variations of velocity and energy or an internal iterative calculation loop must be performed to derive accurate results for one time-advancing analysis. To overcome these disadvantages, this study developed a simultaneous pressure–velocity–enthalpy coupling method for one time advancing analysis.

Figure 3 illustrates the coupling method proposed in this study. The pressure–velocity–enthalpy coupled algorithm first obtains the velocity value by analyzing the pressure–velocity coupling method and then uses the obtained velocity in the discretization of the continuity equation of the pressure–enthalpy coupled algorithm in Eq. (12). This method can robustly perform analysis even when discontinuous flows are predicted such as shock waves in supersonic external or internal flows with large variations in velocity, pressure, and energy. The inner iteration was employed to perform with large CFL number simulation.

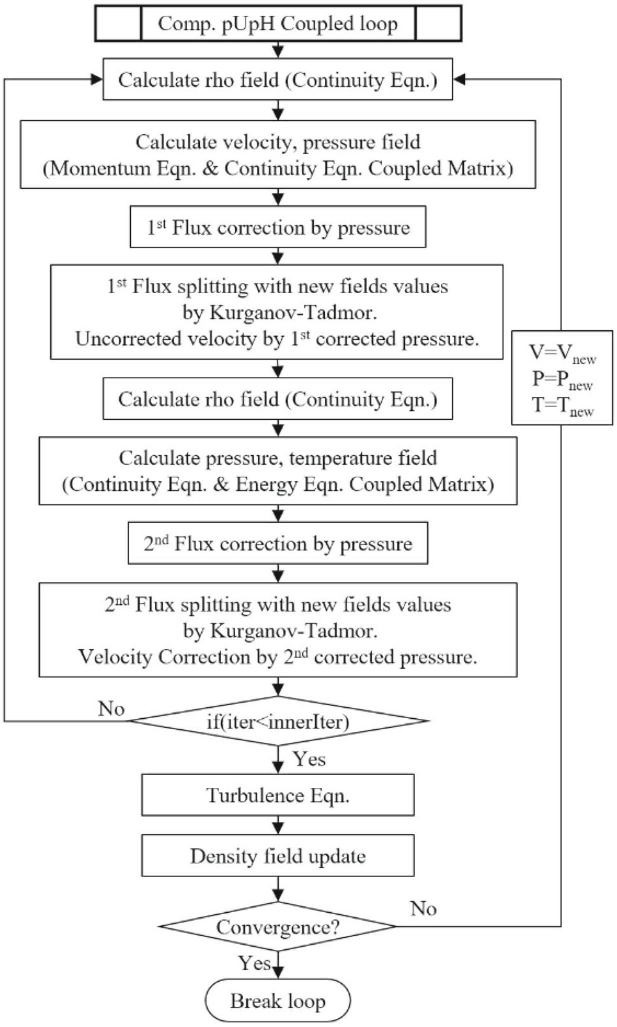
Comp. pUpH Coupled loop
Calculate rho field (Continuity Eqn.)
Calculate velocity, pressure field
(Momentum Eqn. & Continuity Eqn. Coupled Matrix)
1st Flux correction by pressure
1st Flux splitting with new fields values
by Kurganov-Tadmor.
Uncorrected velocity by 1st corrected pressure.
Calculate rho field (Continuity Eqn.)
Calculate pressure, temperature field
(Continuity Eqn. & Energy Eqn. Coupled Matrix)
2nd Flux correction by pressure
2nd Flux splitting with new fields values
by Kurganov-Tadmor.
Velocity Correction by 2nd corrected pressure.
No
if(iter<innerIter)
Yes
Turbulence Eqn.
Density field update
No
Convergence?
Yes
Break loop
V=Vnew
P=Pnew
T=Tnew

Fig. 3 Pressure based U–p–h coupled algorithm

Furthermore, this study applied the flux-splitting technique suggested by Kim and Gill [2] to flux calculation to more accurately predict discontinuous flow phenomena. This method was introduced by Kraposhin et al. [3] and was developed to apply the central flux splitting method at pressure based algorithm.

3 Results

3.1 One-Dimensional Euler Problem

To verify the analysis ability of the solver to which the developed pressure–velocity–enthalpy coupled algorithm has been applied, three one-dimensional Euler problems were selected. The Sod [11], Lax [12], and Shu [13] problems, which are used for verification of discretization and higher-order methods, were applied to the developed solver.

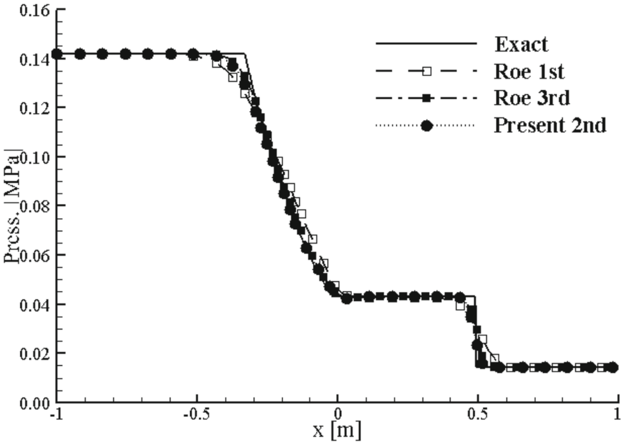

Fig. 4 Sod problem pressure distribution

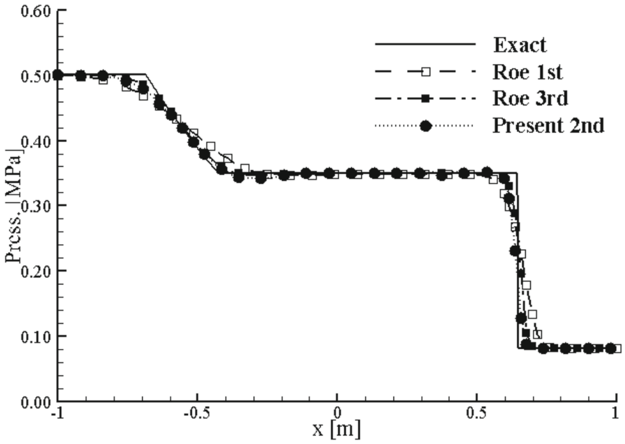

Fig. 5 Lax problem pressure distribution

The OpenFOAM performs dimensional analysis unlike the dimensionless in-house solver. Therefore, the results were compared with those of the dimensionless in-house solver by assuming the reference temperature (288 K) and reference pressure (101,325 Pa).

3.1.1 Sod Problem

The dimensionless initial conditions of the Sod [11] standard shock wave tube problem are as follows.

$$(\rho, u, p) = \begin{cases} (1, 0, 1) \text{ if} x \leq 0 \\ (0.125, 0, 0.1) \text{ if} x > 0 \end{cases}.$$

The extrapolation condition was given as the output boundary condition at both ends $x = \pm 0.5$ m. The dimensionless calculation time is $t_f = 0.14$ (8.23×10^{-4} s for dimensional calculation time), and the number of used grids is $N = 100$.

Figure 4 shows the analysis results by the developed solver and the in-house solver at the dimensionless calculation time $t_f = 0.14$ together with the exact solution. The second-order accuracy analysis result of the developed solver (present) shows higher agreement with the exact solution in the expansion zone and contact face than the Roe 1st (1st order accuracy of Roe FDS and MUSCL), and is similar to the Roe 3rd (Roe FDS, MUSCL 3rd order accuracy).

3.1.2 Lax Problem

The Lax [12] problem is similar to the Sod [11] standard shock wave tube problem, but there is a difference of 6 times between the pressure conditions at both ends, and the dimensionless initial condition is as follows:

$$(\rho, u, p) = \begin{cases} (0.445, 0.698, 3.528) \text{ if} x \leq 0 \\ (0.5, 0, 0.571) \text{if} x > 0 \end{cases}.$$

The extrapolation condition was given as the output boundary condition at both ends $x = \pm 0.5$ m. The dimensionless calculation time is $t_f = 0.13$ (7.64×10^{-4} s for dimensional calculation time), and the number of used grids is $N = 100$.

As with the result of the Sod problem, in the Lax problem applying the developed solver, the developed solver shows better gradient capture ability than the Roe 1st and the result is similar to results of the Roe 3rd, which is the third order accuracy as shown in Fig. 5.

3.1.3 Shu Problem

This is an interaction problem with the density field where the shock waves are distributed as the sine waves [13], and the flow field is created in continuous and discontinuous sections. The given dimensionless condition is as follows:

$$(\rho, u, p) = \begin{cases} (3.857, 2.629, 10.33) \text{if} - 5 \leq x < -4 \\ (1 + 0.2 \sin 5x, 0, 1) \text{if} - 4 \leq x \leq 5 \end{cases}.$$

The extrapolation condition was given as the output boundary condition at both ends $x = \pm 0.5$ m. The dimensionless calculation time is $t_f = 1.8$ (5.29×10^{-3} s for dimensional calculation time), and the number of used grids is $N = 400$. Since no exact solution to this problem is known, the calculation result using Roe FDS with 1600 grid points was regarded as the exact solution.

Figure 6 shows a graph representing the result of the Shu problem that applies the solver developed at the dimensionless calculation time of $t_f = 1.8$, the result of the in-house solver, and the exact solution. It seems that each solver derives a similar result as the exact solution. However, all the solvers cannot accurately capture the exact solution in the sharp gradient that occurs at the beginning of the sine wave region, but after the shock wave has passed and the sine wave is restored, all the solvers follow the exact solu-

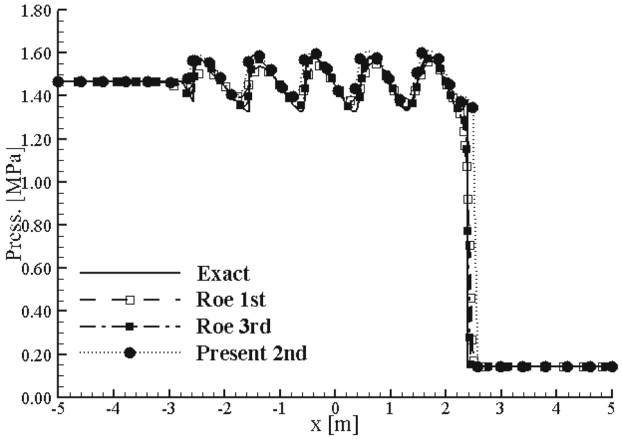

Fig. 6 Shu problem pressure distribution

tion well. It can be seen that the developed solver matches better with the exact solution than Roe 1st.

3.2 Inviscid Flow of Two-Dimensional Circular Bump

The bump flow is generally used to examine the accuracy and convergence of a preconditioning matrix. The developed solver is a pressure-based solver and has been modified to be suitable for low-speed and high-speed flows. Therefore, the inviscid flow of a two-dimensional circular bump was analyzed to verify the ability of the solver in various Mach number regions. The tests were performed for the flow velocities of subsonic, transonic, and supersonic regions with Mach numbers of 0.01, 0.1, 0.675, and 2.0. In the subsonic and transonic regions, the zero gradient condition (Neumann condition) was applied for the outlet velocity and temperature boundary conditions, and the fixed boundary condition (Dirichlet condition) applied for pressure. In the supersonic region, the zero gradient condition was applied for all the outlet velocity and temperature, and pressure.

The results of the developed solver were compared with the previous analysis results of the preconditioned compressible solver [15] using the preconditioning matrix of Choi and Merkle [14] and the compressible solver using a low Mach number modification method with no preconditioning matrix [16, 17]. The methods used in the analysis are outlined in Table 1. The meshes using at both solvers were shown in Fig. 7.

3.2.1 Mach number 0.01

As shown in the results of Mach number 0.01 in Fig. 8, the Mach number distribution around the bump shows a symmetrical characteristic for preconditioned in-house solver (Precon. Roe) and present solver (present), but is biased to the right side for LM Roe. However, for the developed solver, the flow is distorted in the downstream. The energy equation in the present solver was not performed in this case.

Table 1 Numerical schemes

Solver	Precon. Roe/LM Roe	Present
Spatial Discretization	Upwind Roe FDS 2nd minmod	2nd Central Difference Kurganov-Tadmor minmod
Time Integration	LU-SGS	1st order Euler

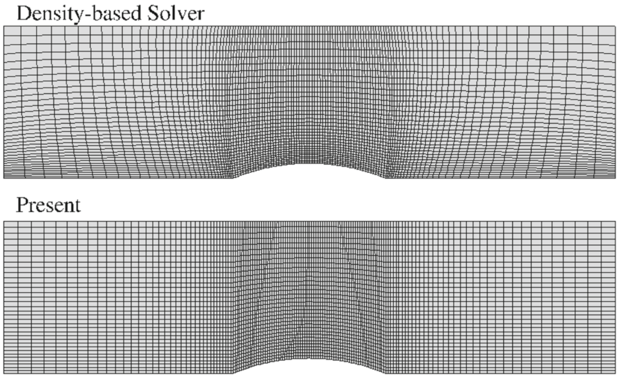

Fig. 7 Meshes of both solver for circular bump

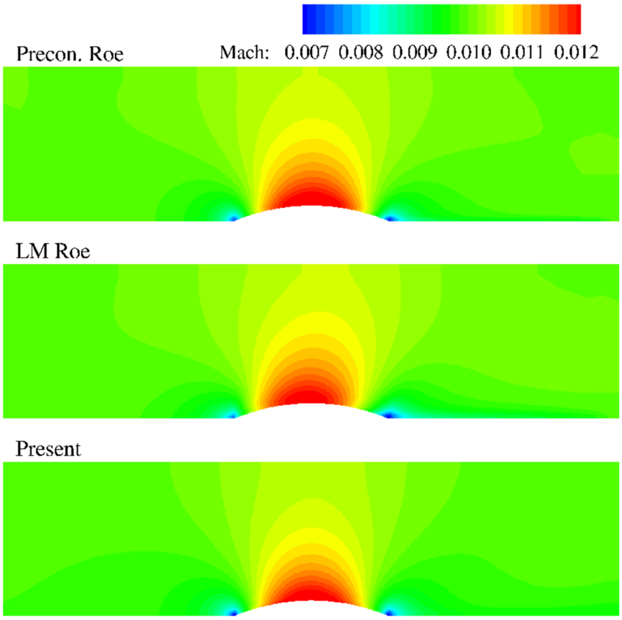

Fig. 8 Mach distribution at Mach 0.01 case

To examine the bottom wall pressure distribution in Fig. 9, for all solvers, the pressure increases in front of the bump, drops to the maximum at the peak of the bump, and then restores toward the downstream. However, for Present, the spatial discretization updates the pressure with the central difference, and the characteristics of a checkerboard appear

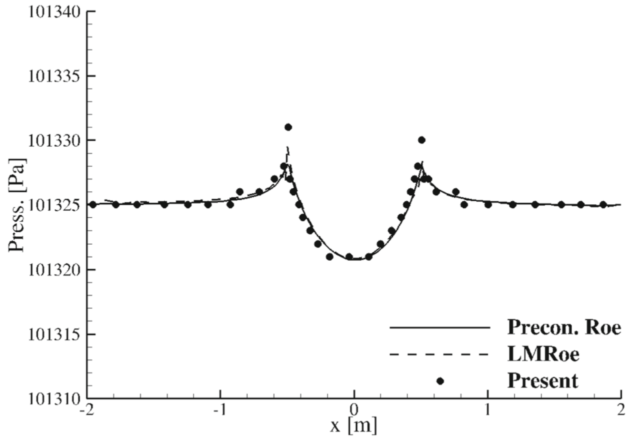

Fig. 9 Bottom wall pressure distribution at Mach 0.01 case

in the sections near bottom wall boundary condition. This seems to be a numerical vibration generated when the limiter, which is applied to the flux-splitting method used in the developed solver, is applied to a low-velocity flow. Some limiters have a tendency to oscillation at low speed flow. Therefore, a care should be taken when selecting a limiter of the flux-splitting method to analyze a very low velocity flow with little pressure gradient. The pressure difference at the corners of bump seems to be caused by difference in mesh density at the bump corners.

3.2.2 Mach Number 0.1

As shown in the results of Mach number 0.1 in Fig. 10, the Mach number distribution around the bump shows symmetrical characteristics for Precon. Roe and present solvers, but is biased to the right side for LM Roe, as with Mach number 0.01. Furthermore, as shown in Fig. 11, the bottom wall pressure distributions are similar for all the solvers. In this case, even if the central difference method is used for present, the characteristics of a checkerboard do not appear.

3.2.3 Mach Number 0.675

In the transonic region, as shown in Fig. 12, all solvers capture the shock wave well after the bump peak. The overall flow shapes are similar for Precon. Roe and LM Roe, but somewhat different for present. This is because the developed solver is a pressure-based solver, whereas the Precon. Roe and LM Roe solvers are density-based compressible solvers, and there is a difference in the flux method between them. Figure 13 shows that in spite of the differences in the solvers, the bottom wall pressure distributions are very similar.

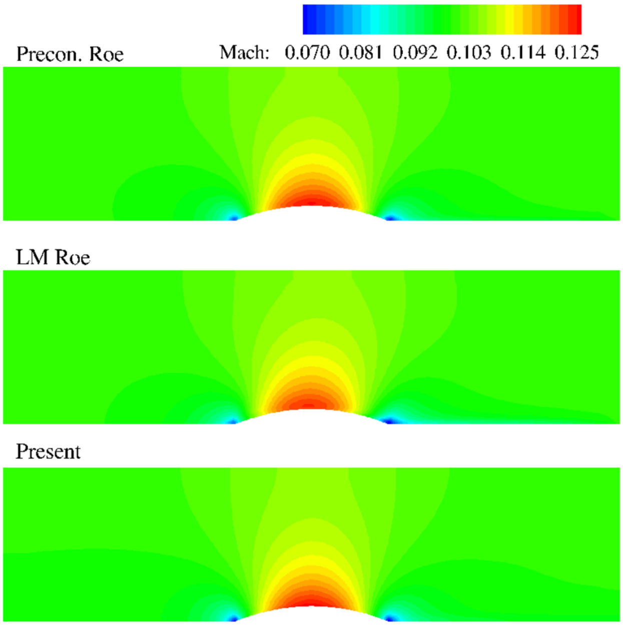

Fig. 10 Mach distribution at Mach 0.1 case

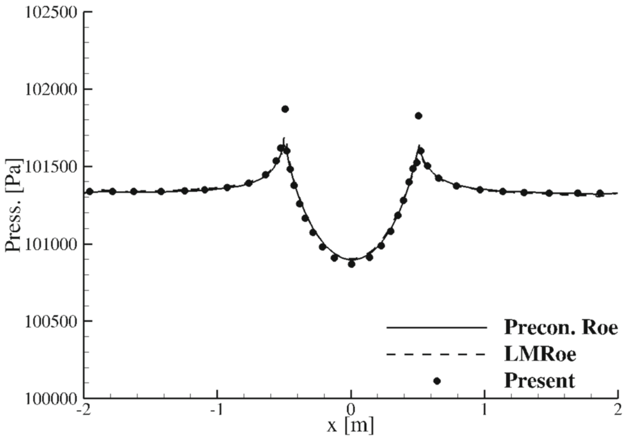

Fig. 11 Bottom wall pressure distribution at Mach 0.1 case

3.2.4 Mach Number 2.0

In the supersonic region, the Precon. Roe and LM Roe, which are density-based compressible solver, capture the oblique shock waves in front of the bump very sharply as shown in Fig. 14. However, the pressure-based present solver captures them somewhat thickly. Furthermore, in the bump wake flow, the density-based and pressure-based solvers capture the recompression waves well. For the density-based solver, in the top and bottom boundary conditions, the velocity was extrapolated for the inviscid wall condition, and for the pressure, density, and temperature. However, although the unattached wall condition provided by the OpenFOAM was applied to developed solver, reflection waves were generated from the bottom in the bump downstream as shown

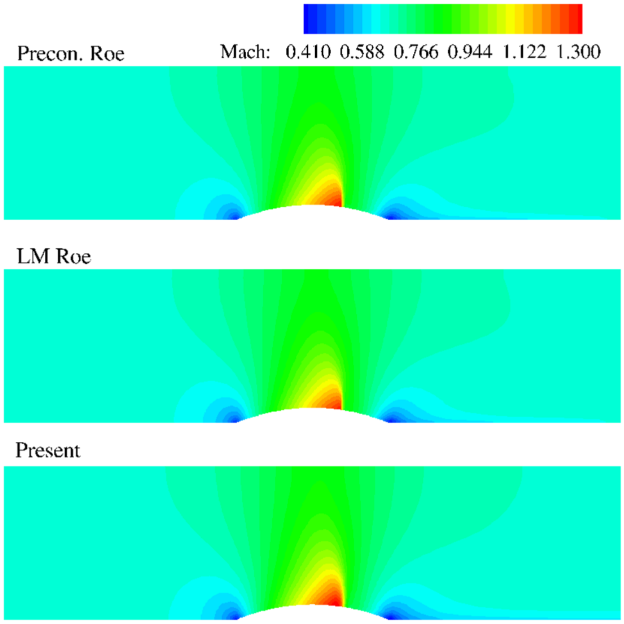

Fig. 12 Mach distribution at Mach 0.675 case

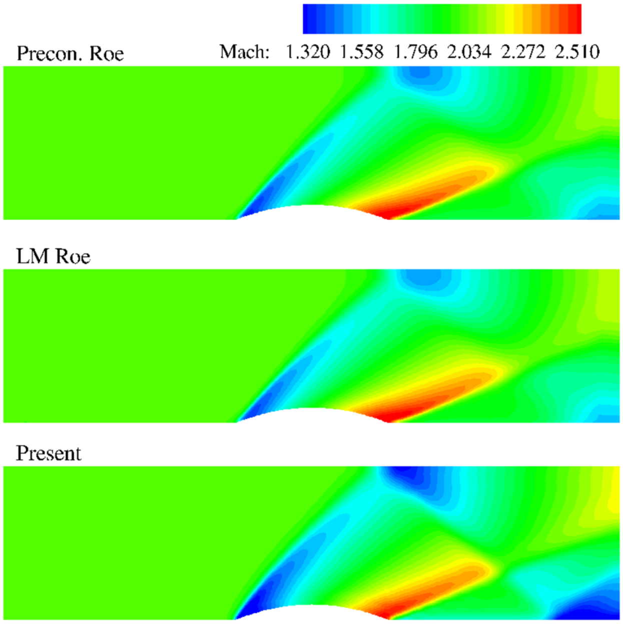

Fig. 14 Mach distribution at Mach 2.0 case

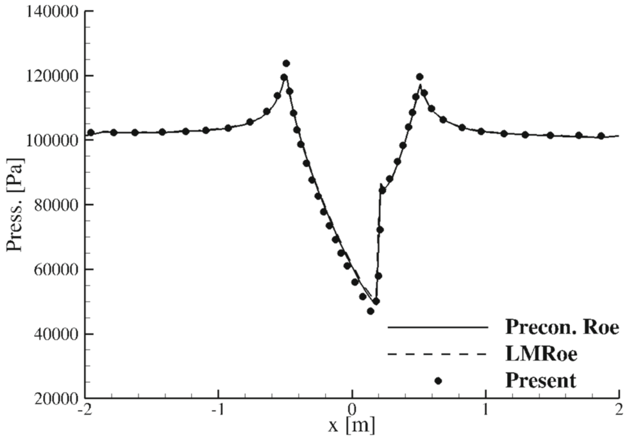

Fig. 13 Bottom wall pressure distribution at Mach 0.675 case

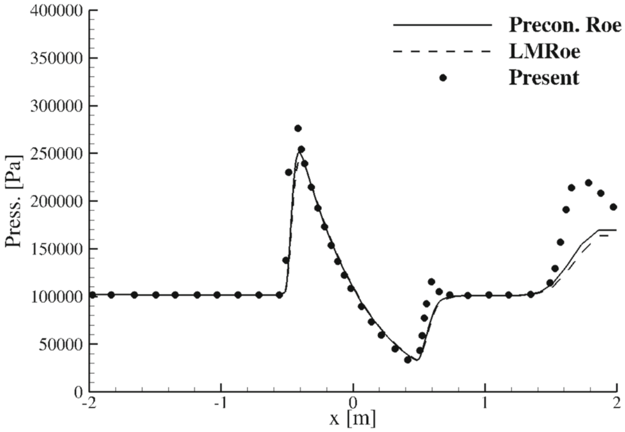

Fig. 15 Bottom wall pressure distribution at Mach 2.0 case

in Fig. 14. This reflection wave was occurred by the strong shock wave than that of density-based solvers. In Fig. 15, the pressure after the oblique shock wave in the present solver is overestimated by approximately 5% compared with the density-based solver. This caused a phenomena of the above-mentioned.

3.3 Inviscid Flow of Two-Dimensional JPL Nozzle

The flow of the NASA JPL transonic nozzle was analyzed to verify the flow of the developed solver. The nozzle used for this verification is a contraction–expansion nozzle with a contraction angle of 45° and an expansion angle of 15°. It is used to verify the axisymmetric flow and the transonic flow near the nozzle neck [18]. For inlet conditions, the experimental conditions of Cuffel et al. [18], that is, the pre-pressure of 4.826 × 105 Pa (= 70 psia) and the pre-temperature of 300 K (= 540°R) were applied. Figure 16 shows the Mach number and pressure. It correctly shows the characteristics of nozzle flow that the pressure decreases as the flow is gradually accelerated through the contraction part and the flow is further accelerated through the expansion part.

The inviscid flow analysis results show that the Mach numbers at the nozzle neck wall and centerline near the transonic region are 1.248 and 0.806, respectively. These values are similar to the Mach number of the nozzle neck wall obtained by the isentropic calculation of Cuffel et al. [18] and the Mach number of the centerline obtained by measurement, which are 1.4 and 0.8, respectively.

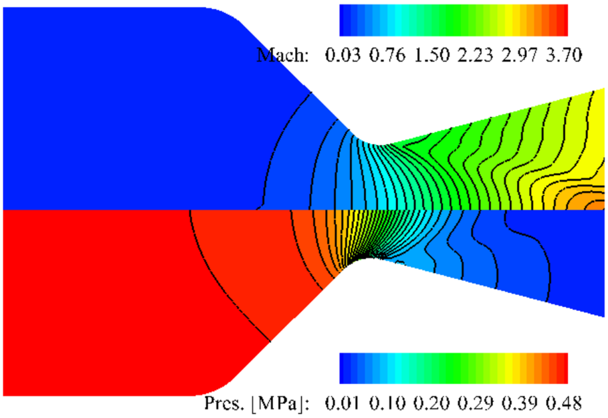

Fig. 16 Mach and pressure distribution

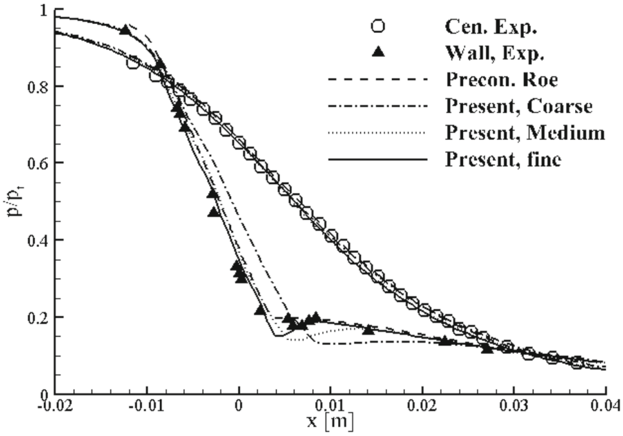

Fig. 17 Wall and centerline pressure distribution

Figure 17 compares the distributions of the pressure ratio (static/total pressure) at the nozzle centerline and wall among measured data obtained by Cuffel et al. [18], previous analysis results [15], and the developed solver. It seems that the experimental results, previous analysis results, and the present results agree well.

3.4 Viscous Flow of Three-Dimensional Missile

To verify the viscous flow analysis ability of the developed solver, the aerodynamic coefficient of a short-range air-to-air missile was analyzed. Menter's 2-equation k–ω SST RANS turbulence model was used for viscous flow analysis. Graves and Fournier experimented for the aerodynamic characteristics of a short-range air-to-air missile for various Mach numbers [19]. Their report shows the aerodynamic coefficient values according to the shape, dimension, and maneuver of air-to-air missiles.

Figure 18 shows a visualization of the pressure distribution on the surface of the missile and the flow field around the missile as interpreted at an angle of attack of 25° and a Mach number of 1.2.

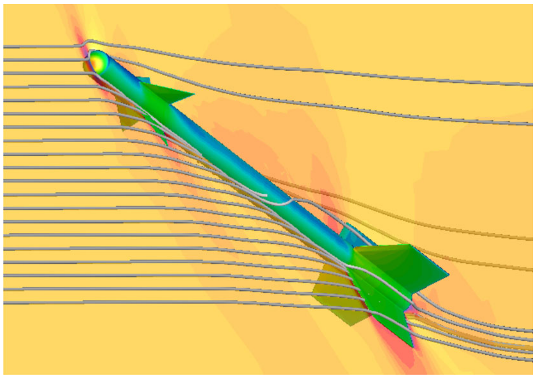

Fig. 18 Pressure distribution at AOA 25, Mach 1.2

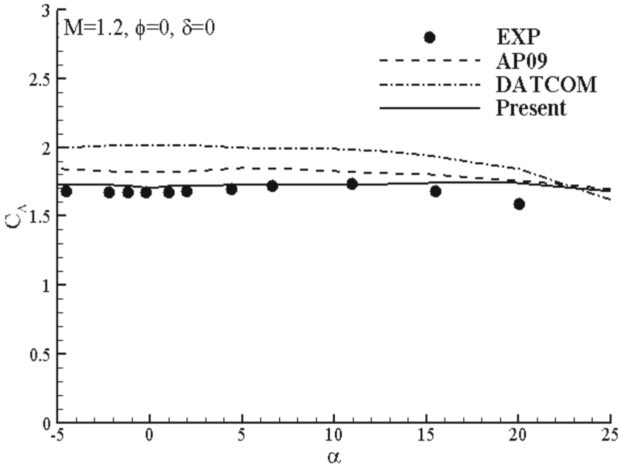

Fig. 19 Axial force coefficient at AOA sweep, Mach 1.2

The aerodynamic coefficients obtained by analyzing the flight conditions of the missile for various angles of attack at a Mach number of 1.2, a roll angle of 0° (ϕ: roll angle) and a canard deflection angle of (δ: canard deflection angle) were compared with the calculations and experiment results of previous researchers. The comparison results of the axial direction, normal direction, and pitching moment coefficient are shown in Figs. 19,20,21, respectively.

The axial force coefficient in Fig. 19 shows much higher performance at angles of attack between − 5° and 15° than the missile design tools AP09 [20] and DATCOM. However, in terms of the overall tendency, the present solver overestimates the values at higher angles of attack, but the values are similar to the level of the missile design tools.

The normal force coefficient in Fig. 20 matches well with the missile design tools and experiment values in the entire range of angle of attack. Furthermore, the pitching moment coefficient in Fig. 21 shows much higher performance than those of the missile design tools AP09 and DATCOM at the angles of attack between − 5° and 15° similarly to the

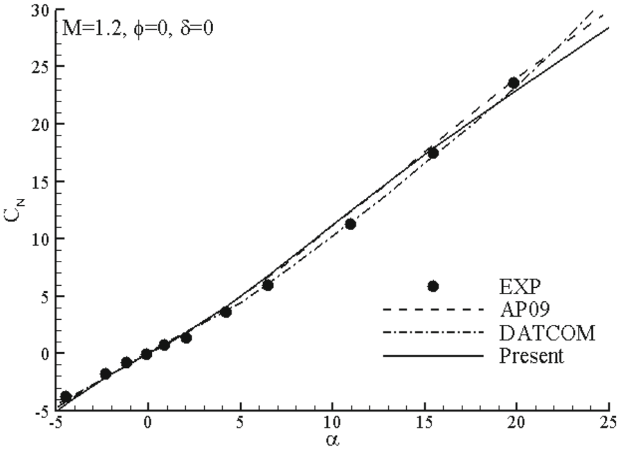

Fig. 20 Normal force coefficient at AOA sweep, Mach 1.2

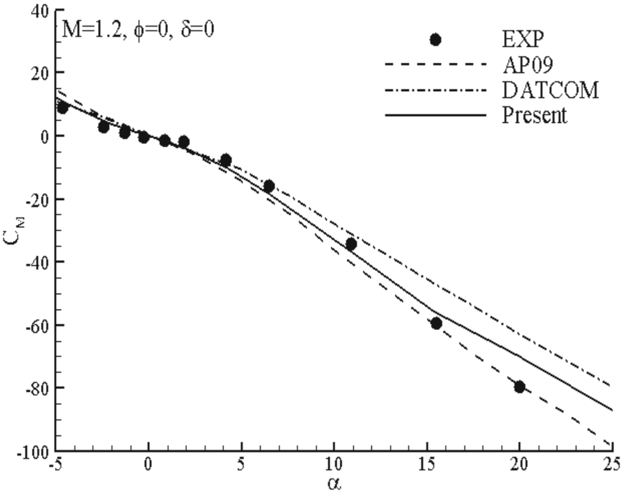

Fig. 21 Pitching moment at AOA sweep, Mach 1.2

axial force coefficient results in Fig. 18. However, the present developed solver underestimates the values compared with AP09 and experiment at the angle of attack above 20°.

3.5 OneraM6 Three-Dimensional Wing

To verify the developed solver, the Onera M6 wing case was employed. The flow conditions and base mesh were obtained from NASA NPARC website [21]. The coarse mesh was generated by converting the PLOT3D format mesh to OpenFOAM mesh. Figure 22 is coarse mesh obtained from NASA NPARC. The fine mesh was obtained by splitting cells of the coarse mesh by 2 times along each i.j.k directions of PLOT3D format. The fine mesh cells number is 2.4 million. The characteristic boundary condition was adopted to far boundary

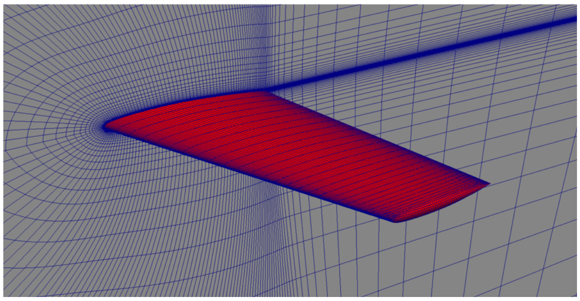

Fig. 22 NASA presented mesh of Onera M6 wing

and Spalart–Allmaras turbulence model was employed. The freestream Mach number is 0.84 and freestream Reynolds number is 11.76×10^6. The angle of attack is 3.06° and freestream temperature is 460 R. The local time stepping time integration scheme was used for steady state analysis. The CFL number and inner iterative number were set up 5 and 2 respectively while the iteration number is 5000 for coarse mesh and 10,000 for fine mesh.

Figure 23 shown the iterative residuals of both meshes. Based on Fig. 23, it can be observed that the simulation heading into a converged solution where the increasing frequency at early iteration steps of each variable residual continually trending towards steady state as the iterations progressed at both meshes. Figure 24 shown the pressure coefficient results around OneraM6 wing of both analysis meshes. As shown as in these figure, the λ shock on upper wing surface appears at both meshes but more clearly at density mesh. Especially the following shock of density mesh is captured better than coarse mesh.

Figure 25 shown the pressure coefficients on wing surface at specified position according to spanwise. Each positions on spanwise were selected form the experiment. It indicated that the calculated results from both meshes were seem as like with other numerical and experiment data. However, the pressure coefficients on leading edge were underestimated at coarse mesh than other plotted results. As like Fig. 24, the second shock of λ shock on upper wing surface were weakly estimated at coarse mesh. This phenomena was occurred when the calculated node points were not sufficient to estimate the discontinuous phenomenon as like shock. Whereas, the calculated result with fine mesh were similar compared with the experiment data. Therefore, it indicated that the developed solver with suggested algorithm has the ability to analyze the transonic region problem including the shock phenomenon if proper mesh provided. In addition, it shown that the problems with discontinuous flow could be calculated with high CFL number by the suggested algorithm.

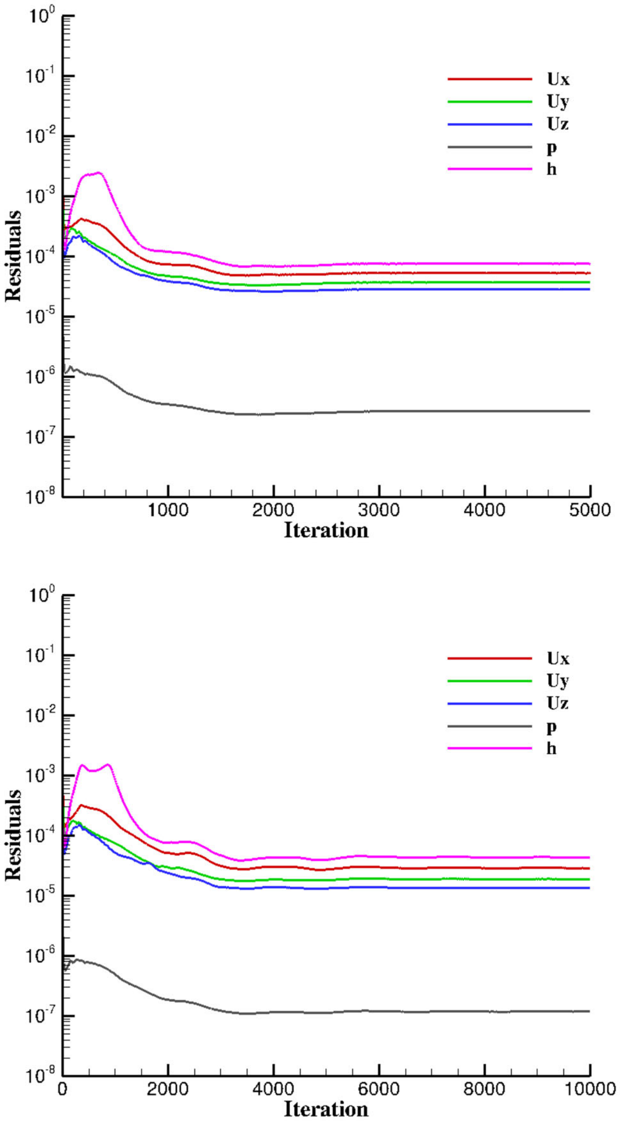

Fig. 23 Iterative residuals of coarse mesh (upper figure) and fine mesh (lower figure)

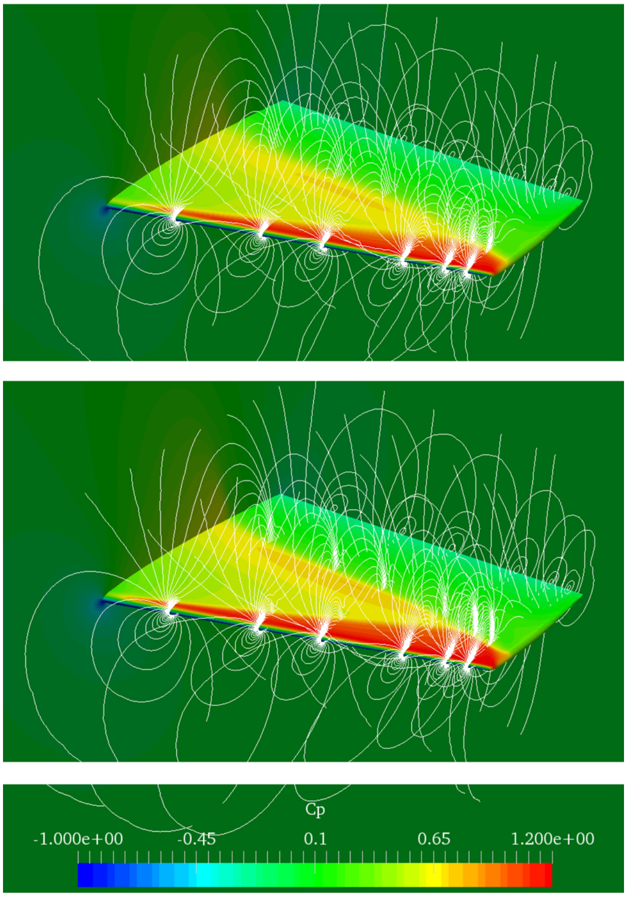

Fig. 24 Pressure coefficient around wing with coarse mesh (upper figure) and fine mesh (lower figure)

4 Conclusions

To evaluate the analysis ability of the developed solver, the one-dimensional Euler problem, the two-dimensional inviscid bump flow, the two-dimensional inviscid JPL nozzle flow, and the three-dimensional missile aerodynamics were analyzed.

The analysis results of the two-dimensional inviscid bump flow show that the checkerboard characteristics appear at very low velocities, but similar results to those of the compressible solver that applied preconditioning were obtained at higher velocities. Furthermore, the results in the overall flow field were better than the low Mach number modification method. Therefore, the developed solver could be applied in a wide velocity range although there are some limitations in the low velocity range. In particular, the developed solver demonstrated an ability to simulate a contraction–expansion flow around the nozzle neck such as the two-dimensional JPL nozzle problem in the transonic range.

For viscous flow, when the developed solver was applied to the aerodynamic analysis of a three-dimensional missile, the results matched with the experiment values better than those of missile design programs. For wing case, the developed was able to analyze at the transonic flow with high CFL numbers.

The solver was developed on the basis of OpenFOAM, which is an open-source CFD code, so as to use it as a basic aerodynamic solver for analysis of rocket engine combustion flow field. It will be applied to the analysis of rocket engine turbulence combustion flow field in combination with the flamelet model that is under development. In additional, it will be performed to a quantitative error analysis at various cases to confirm the performance estimation abilities of solver.

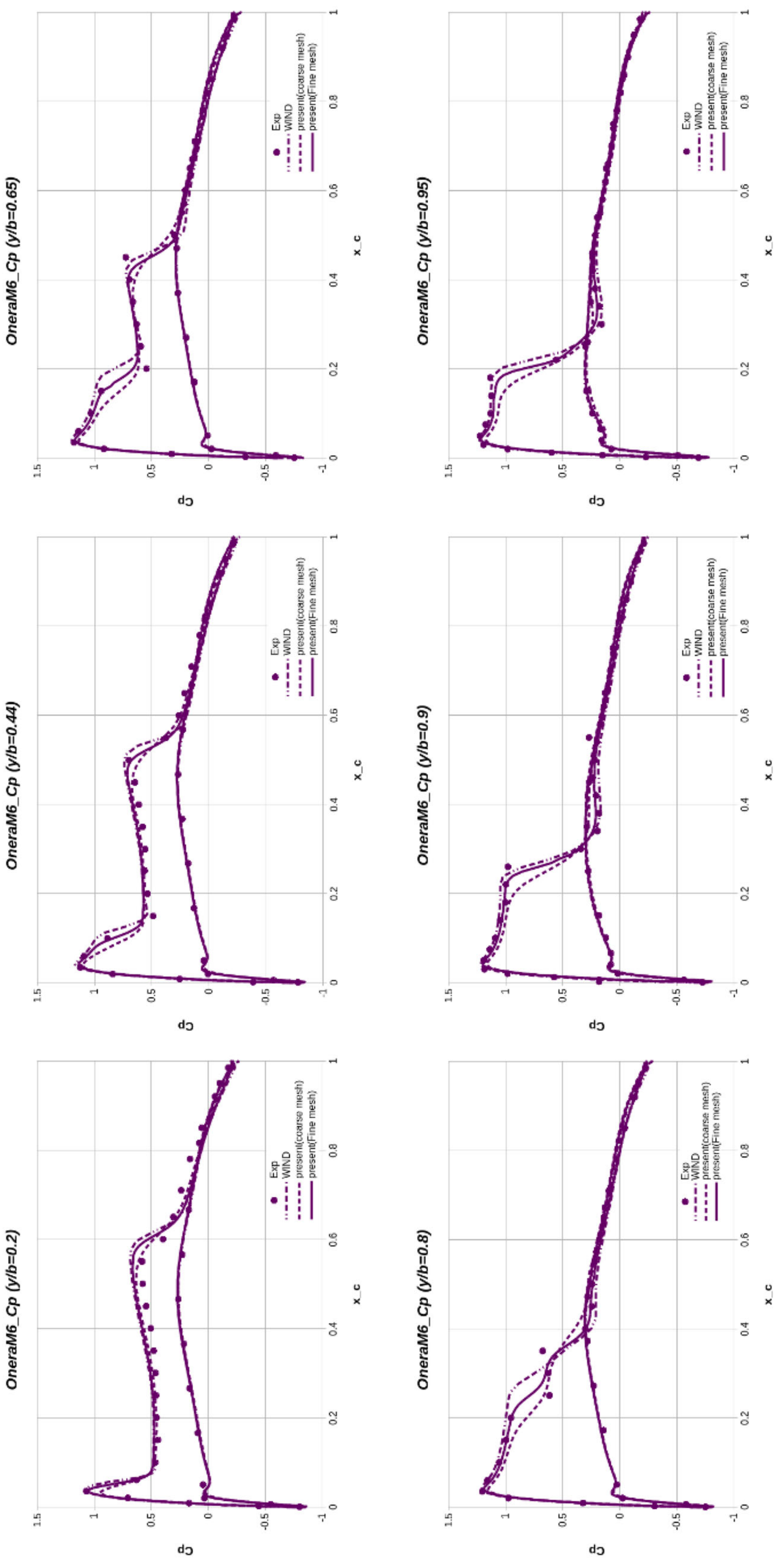

Fig. 25 Pressure coefficient on Onera M6 wing surface according to spanwise position

Acknowledgements This study was supported by the Aerospace Core Technology Development Project (2017M1A3A3A04016580) of the Korea Research Foundation with government funding (Ministry of Science and ICT).

References

1. Patankar J (1980) Numerical heat transfer and fluid flow. CRC Press, Boca Raton
2. Kim TW, Gill JH (2017) Development of improved pressure-based solver algorithm for compressible flow. KSCFE 22(4):28
3. Kraposhin M, Bovtrikova A, Strijhak S (2015) Adaptation of Kurganov–Tadmor numerical scheme for applying in combination with the PISO method in numerical simulation of flows in a wide range of mach numbers. Proc Comput Sci 66:43–52
4. Darwish M, Moukalled F (2014) A fully coupled Navier–Stokes solver for fluid flow at all speeds. Numer Heat Transf Part B 65(5):410
5. Mangani L, Darwish M, Moukalled F (2016) An OpenFOAM pressure-based coupled CFD solver for turbulent and compressible flows in turbomachinery applications. J Numer Heat Transf Part B Fundam 69(5):413
6. Mangani L, Buchmayr M, Darwish M, Moukalled F (2017) A fully coupled OpenFOAM® solver for transient incompressible turbulent flows in ALE formulation. Numer Heat Transf Part B Fundam 71(4):313
7. Emans M, Frolov SM, Lidskii B, Posvyanskii V, Basara B (2012) SIMPLE-H: a finite-volume pressure-enthalpy coupling scheme for flows with variable density. Int J Numer Methods Fluids 69(1):206
8. Emans M, Zunic Z, Basara B, Frolov SM (2012) A novel SIMPLE-based pressure-enthalpy coupling scheme for engine flow problems. Math Model Anal 17(1):1
9. Xiao C-N, Denner F, van Wachem B (2017) Fully-coupled pressure-based finite-volume framework for the simulation of fluid flows at all speeds in complex geometries. J Comput Phys 346:91
10. Denner F (2018) Fully-coupled pressure-based algorithm for compressible flows: linearisation and iterative solution strategies. Comput Fluids 175:53
11. Sod GA (1978) A survey of several finite difference methods for systems of non-linear hyperbolic conservation laws. J Comput Phys 27(1):1–159
12. Lax PD (1954) Weak solutions of nonlinear hyperbolic equations and their numerical computation. Commun Pure Appl Math 7:159–193
13. Shu C-W (1990) Numerical experiments on the accuracy of ENO and modified ENO schemes. J Sci Comput 5(2):127–149
14. Choi Y, Merkle C (1993) The application of preconditioning in viscous flows. J Comput Phys 107:207–223
15. Han SH, Kim MH, Choi JY (2007) Migration from compressible code to preconditioned code. J Korean Soc Aeronaut Space Sci 35(3):183–195
16. Rieper F (2011) A low-Mach number fix for Roe's approximate Riemann solver. J Comput Phys 230:5263–5287
17. Kim SD, Han SH, Shin JR, Choi JY (2012) Comparison of low-Mach number fixing methods for Roe scheme. In: Proceeding of KSAS spring conference, Rep. of Korea, p 4
18. Cuffel RF, Back LH, Massier PF (1969) Transonic flow field in a supersonic nozzle with small throat radius of curvature. AIAA J 7(7):1364–1366
19. Graves EB, Fournier RH (1974) Stability and control characteristics at Mach numbers from 0.20 to 4.63 of a cruciform air-to-air missile with triangular canard controls and a trapezoidal wing. NASA Technical Memorandum, NASA TM X-3070
20. Wee HC (2011) Aerodynamic analysis of a canard missile configuration using Ansys-Cfx. Naval Postgraduate School, Thesis, Monterey
21. NASA. NPARC Alliance Verification and Validation Archive. https://www.grc.nasa.gov/WWW/wind/valid/m6wing/m6wing.html