6th European Conference on Computational Mechanics (ECCM 6)
7th European Conference on Computational Fluid Dynamics (ECFD 7)
11 – 15 June 2018, Glasgow, UK

A SMAC LIKE NOVEL EFFICIENT IMPLISIT MUSCL METHOD FOR ALL MACH NUMBER

EIJI SHIMA [1] and KEIICHI KITAMURA [2]

[1] Japan Aerospace Exploration Agency (JAXA)
Sagamihara, Kanagawa, 252-5210, Japan, shima.eiji@jaxa.jp

[2] Yokohama National University
Yokohama, Kanagawa, 240-8501, Japan,
kitamura@ynu.ac.jp

Key Words: SLAU, AUSM, Flux Function, Time Integration Method.

Abstract. The formulation of a new all-speed compressible CFD (Computational Fluid Dynamics) scheme, which unifies a compressible CFD schemes of implicit MUSCL (Monotonic Upstream-Centered Scheme for Conservation Laws) and an incompressible CFD scheme of SMAC (Simplified Marker and Cell) methods, SMUC (SMAC-inspired Mach Uniform Compressible scheme), is presented. New scheme has no Mach dependent adjustable parameter such as a cut-off Mach number. Numerical examples show improvements by this scheme, such as several times speed up to the convergence in low Mach number range. It is also shown that the procedure of SMAC method is theoretically derived from the operator factorization of minimum errors.

NOMENCLATURES

c : Sound speed

C : Reference sound speed

e : Total energy per unit volume

e_i : Internal energy per unit volume

$\hat{\mathbf{E}}$: Outward normal inviscid flux at cell boundary

$\tilde{\mathbf{E}}$: Outward normal numerical inviscid flux at cell boundary

h : Enthalpy

Q

$\dot{m}$: Mass flux

M : Mach number

M_c : Cut-off Mach number

$\mathbf{n}$: Outward normal vector at cell boundary (x_n, y_n, z_n)

$\hat{\mathbf{R}}$: Viscous flux at cell boundary

$\tilde{\mathbf{R}}$: Numerical viscous flux at cell boundary

p : Pressure

$\tilde{p}$: Numerical pressure at cell boundary

$\mathbf{Q}$: Vector of Conservative variables $(\rho, \rho u, \rho v, \rho w, e)^T$

$\mathbf{q}$: Vector of Conservative variables $(p, u, v, w, s)^T$

s : Entropy

S : Area

t : Time

$\mathbf{u}$: Velocity vector $(u, v, w)^T$

U : Reference advection speed

V_n : Outward normal velocity at cell boundary

x, y, z : Cartesian coordinates

Ω : Cell volume

ρ : Density

μ : Molecular viscosity coefficient

μ_T : Eddy viscosity coefficient

Lower superscript

i : Cell index

i, j : The 'j' th cell boundary or neighboring cell of cell 'i'

L : Left side of cell boundary

R : Right side of cell boundary

1 INTRODUCTION

All real fluids are compressible, but the influence of compressibility differs depending on M (Mach number).

In principle, the compressible NS (Navier-Stokes) equation is a governing equation of fluids for all *M*, but incompressible NS equations are often used at low *M* flows of approximately $M < 0.2$. In CFD, the basic schemes are different between them, and this has made their different progresses historically [1]-[6].

In the flow field above subsonic speed, it is essential to use compressible CFD. Also, even for low *M* flow, compressibility needs to be considered when directly handling sound waves. For example, in the rocket engine combustor flow, a wide range of *M* change appears as a series of phenomena, from low *M* flow of liquid fuel/oxidizer to supersonic flow after nozzle. Furthermore, a simultaneous analysis of flow and sound waves is necessary for phenomena such as combustion oscillation in which sound affects flow fields. Therefore, it is necessary to deal with a broad spectrum of *M* flows and to solve flows and sound waves simultaneously.

The compressible CFD schemes that can simulate flows and sounds simultaneously can be roughly divided into density-based and pressure-based ones. In the former, the conservation law is solved using conservative variables, which are density, momentum and total energy; and then, pressure is obtained from the equation of state from density and internal energy.

In this category of computational algorithms, various schemes combining MUSCL [4] (Monotone Upwind Scheme for Conservation Laws)-FVM (Finite Volume Method), Riemann flux and δ-form implicit scheme for time integration, are widely used in the aerospace field.

These schemes are flexible to cell (grid) geometry and applicable both to structured and unstructured grids. In addition, a δ-form implicit scheme permits various combinations of its L.H.S. (numerical) and R.H.S (physical). For example, high spatial accuracy and high computational efficiency are achieved using a higher-order scheme on R.H.S. and a first-order scheme in L.H.S.

The Riemann flux used in MUSCL is the key to realize robustness and clear discontinuity capturing without adjustment of numerical dissipation, and the AUSM (Adjustment Upstream Splitting Method)[6] family schemes have been developed to achieve this without complexity.

However, in low *M* regions, the following problems are reported.

(a) Stiffness due to a large ratio of sound speed and advection speed

(b) Excessive numerical dissipation at low M included in numerical scheme

(c) Round-off errors due to extremely small fluctuation such as pressure and density

Among them, (c) can be cured by using fluctuations as variables, thus, the essential problems are (a) and (b). In response to these problems, time-derivative pre-conditioning methods [7][8], and their corresponding implicit

schemes [9], and all-speed Riemann fluxes that control numerical dissipation appropriately for low M, have been developed. By using these schemes, now it is possible to compute all M flows[10][11]. However, as M decreases, the dissipation against pressure becomes stronger, which hinders the acoustic wave computations expected for compressive CFD schemes. Furthermore, there are still issues for adjustment of problem-dependent M_c (cutoff M) which is necessary for these schemes.

On the other hand, the pressure-based schemes are based on incompressible CFD scheme inherited from MAC (Marker and Cell) method[1]. In incompressible fluids, pressure cannot be calculated from the equations of state, thus, in incompressible CFD, the elliptic equation derived from the equation of momentum and the incompressible condition are solved to obtain pressure. Through this procedure, an efficient method that solves issue (a) has been realized, and it is widely used including various higher order accuracy schemes [12, 13].

An extension of the pressure-based scheme to compressible flows [14]-[17] uses conservative variables, combines a density equation and a momentum equation, and approximates an equation of state to derive implicit pressure equations, similarly to the MAC method. The equation of total energy is added, but the equations of density and momentum are not much different from those of incompressible ones. Computational efficiency at M <0.1 which depends on details of numerical dissipation, and accuracy of simultaneous analysis of fluids and sound waves are not clear. However, in Ref. [14], adjustable parameters similar to M_c were reported necessary for improvement of calculation efficiency at low M. Therefore, there appears to be a problematic like that of the density-based schemes.

On the other hand, Ref. [18] solves time evolution equations of velocity-pressure rather than conservation law. By taking advantage of the fact that the density fluctuation is small in low M flows and sound propagation, the density and energy equations are omitted and simultaneous analysis at low M is realized by a scheme similar to SMAC (Simplified MAC) [2]. Although its extension to general compressible fluids involving density fluctuations is not trivial, the use of the time evolution equation to pressure leads to a simpler formulation which can be extended to M=0 without adjustable parameter.

A comprehensive comparison is difficult because each characteristic is different for each individual scheme. However, it can be said that a density-based scheme using Riemann flux is more suitable to non-oscillatory and sharp capturing of discontinuities including shock waves, whereas pressure-based schemes are more efficient for low Mach number flow computations as their incompressible versions.

Therefore, the combination of the advantages of both seems promising, although the following differences

are concerns;

- Operator splitting: In a pressure-based scheme, each variable and each term are separated by operator splitting or fractional step, and each variable is updated in stages; whereas in the density-based schemes, the conservative variable vector is updated at once. The Riemann flux, playing a key role in density-based schemes, computes the fluxes of all the components at once, which makes it difficult to directly conduct the operator splitting.
- Calculation of pressure: Pressure is calculated by equation of state in density-based ones, and the elliptic equation in pressure-based ones. The latter corresponds to the implicit scheme of pressure [17], but its relationship with the former is unclear.

Therefore, the motivation and purpose of this research are the following two.

(1) Integrate the density-based and pressure-based compressible CFD schemes, in which their mutual relationships are not clear, in the framework of the density-based scheme and δ-form implicit scheme.

(2) By utilizing the features of the SMAC method, we propose a new scheme SMUC (SMAC-inspired Mach Uniform Compressible scheme) that enables simultaneous analysis of arbitrary low M flow and sound waves without adjusting M_c.

As the existing density-based schemes, SMUC consists of the MUSCL-FVM using δ-form implicit scheme and an all-speed Riemann flux.

First, we will introduce an SMAC like implicit time integration method named GC-SMAC (Generalized Compressible SMAC) using δ-form implicit scheme, using transformation to entropy variable and approximate factorization. In its derivation, it will also be shown that the procedure of SMAC is derived from an optimal factorization. Next, as an improved Riemann flux, we will propose a new Riemann flux: UD (Uniform Damping)-SLAU (Simple Low-dissipation AUSM) that can prevent unnecessary oscillations while maintaining the acoustic computation capability. The similarity in numerical dissipations included in GC-SMAC and UD-SLAU will also be explained. In addition, some numerical examples will demonstrate their effectiveness. Finally, a summary and future works will be given.

2 BASIC EQUATIONS AND IMPLICIT SOLVERS IN ENTROPY VARIABLES

Time derivative pre-conditioning method [7][8] that extends compressible CFD to low Mach numbers has realized a neat formulation by using entropy variables (velocity, pressure, entropy). Since the entropy is an only

thermodynamic variable that does not fluctuate by compression, the later formulation will become simple. Let us start from the compressible NS equation using entropy variables.

$$\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u} + \frac{1}{\rho}\nabla p + \mathbf{R}_u = 0 \tag{1}$$

$$p_t + \mathbf{u}\cdot\nabla p + \rho c^2 \nabla\cdot\mathbf{u} + R_p = 0 \tag{2}$$

$$s_t + \mathbf{u}\cdot\nabla s + R_p = 0 \tag{3}$$

$$\partial s \equiv \partial p - c^2 \partial\rho \tag{4}$$

$$\mathbf{R}_u = -\frac{1}{\rho}\nabla\cdot\boldsymbol{\tau} \tag{5}$$

$$R_p = -\left(\frac{\partial p}{\partial e_i}\right)_\rho \left(\frac{1}{\rho}\nabla\cdot\left(\boldsymbol{\tau}\cdot\mathbf{u} - \mathbf{q}\right)\right) \tag{6}$$

τ and q are the viscous tensor and the heat flux vector, respectively. The viscous term is handled as usual, and the details are omitted. Also, Eqs. (2)(3) are unchanged even when using the fluctuations from the reference values are used instead of the absolute values, and then they are free from the adverse effects of rounding errors at low Mach numbers. (Additional measures are necessary on the R.H.S., though)

The key to efficient computations of compressible CFD schemes at low M is the implicit time integration scheme that allows a large Courant number. Leaving the spatially differential form and introducing temporally implicit finite difference, the semi discrete form is derived as;

$$\mathbf{u}^{n+1} - \theta_1\mathbf{u}^n + \theta_2\mathbf{u}^{n-1} + \delta t'\left\{\mathbf{u}^{n+1}\cdot\nabla\mathbf{u}^{n+1} + \frac{1}{\rho^{n+1}}\nabla p^{n+1} + \mathbf{R}_u{}^{n+1}\right\} = 0 \tag{7}$$

$$p^{n+1} - \theta_1 p^n + \theta_2 p^{n-1} + \delta t'\left\{\mathbf{u}^{n+1}\cdot\nabla p^{n+1} + \left(\rho c^2\right)^{n+1}\nabla\cdot\mathbf{u}^{n+1} + R_p{}^{n+1}\right\} = 0 \tag{8}$$

$$s^{n+1} - \theta_1 s^n + \theta_2 s^{n-1} + \delta t'\left\{\mathbf{u}^{n+1}\cdot\nabla s^{n+1} + R_p{}^{n+1}\right\} = 0 \tag{9}$$

$$\left(\theta_1,\theta_2\right) = \left(\frac{2\theta+2}{\theta+2}, \frac{\theta}{\theta+2}\right) \tag{10}$$

$$\delta t' = \frac{2}{\theta+2}\delta t \tag{11}$$

When $\theta = 1$, it is second order temporal accurate, and $\theta = 0$ gives the first order accuracy. Newton iteration is introduced to solve the nonlinear equation for $\mathbf{u}^{n+1}$ etc. Setting Newton iteration count as k, and omitting the superscript k of the coefficients for brevity, and rewriting $\delta t'$ simply as δt;

$$\delta\mathbf{u} + \delta t\left\{\mathbf{u}\cdot\nabla\delta\mathbf{u} + \frac{1}{\rho}\nabla\delta p + \frac{\partial\mathbf{R}_u}{\partial\mathbf{u}}\delta\mathbf{u}\right\} = \delta\mathbf{u}^* \tag{12}$$

$$\delta p + \delta t\left\{\mathbf{u}\cdot\nabla\delta p + \rho c^2\nabla\cdot\delta\mathbf{u} + \frac{\partial R_p}{\partial p}\delta p\right\} = \delta p^* \tag{13}$$

$$\delta s + \delta t\left\{\mathbf{u}\cdot\nabla\delta s + \frac{\partial R_p}{\partial s}\delta s\right\} = \delta s^* \tag{14}$$

$$\mathbf{u}^{k+1} \equiv \mathbf{u}^k + \delta\mathbf{u} \tag{15}$$

$$p^{k+1} \equiv p^k + \delta p \tag{16}$$

$$s^{k+1} \equiv s^k + \delta s \tag{17}$$

Here, the left side viscous term is simplified considering utilization of diagonal approximation. Also, the R.H.S. is defined as follows.

$$\delta\mathbf{u}^* \equiv \mathbf{u} - \theta_1\mathbf{u}^n + \theta_2\mathbf{u}^{n-1} - \delta t\left\{\mathbf{u}\cdot\nabla\mathbf{u} + \frac{1}{\rho}\nabla p + \mathbf{R}_u\right\} \tag{18}$$

$$\delta p^* \equiv p - \theta_1 p^n + \theta_2 p^{n-1} - \delta t\left\{\mathbf{u}\cdot\nabla p + \rho c^2\nabla\cdot\mathbf{u} + R_p\right\} \tag{19}$$

$$\delta s^* \equiv s - \theta_1 s^n + \theta_2 s^{n-1} - \delta t\left\{\mathbf{u}\cdot\nabla s + R_p\right\} \tag{20}$$

The R.H.S. is obtained by variable conversion from discretization with an arbitrary set of variables. Specific examples will be described in Section 5. If Newton iteration converges, the R.H.S. = 0 is established, which means that the implicit scheme of the Eqs. (7)-(9) holds. In practice, set an appropriate threshold for convergence is prescribed, or the number of iterations is specified.

3 2 STEPS METHOD DERIVED FROM OPTIMUL OPERATOR SPLITTING

Vector notation is introduced and differential operators are grouped as follows.

$$\mathbf{q} = \begin{pmatrix} p \\ \mathbf{u} \\ s \end{pmatrix} \tag{21}$$

$${}_p\mathbf{A}_p = \begin{pmatrix} \mathbf{u}^n\cdot\nabla + \dfrac{\partial R_p}{\partial p} & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 \end{pmatrix} \tag{22}$$

$${}_p\mathbf{A}_u = \begin{pmatrix} 0 & \rho c^2\nabla\cdot & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 \end{pmatrix} \tag{23}$$

$$ {}_u\mathbf{A}_p = \begin{pmatrix} 0 & 0 & 0 \\ \dfrac{1}{\rho}\nabla & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 \end{pmatrix} \tag{24} $$

$$ {}_u\mathbf{A}_u = \begin{pmatrix} 0 & 0 & 0 \\ \mathbf{0} & \mathbf{u}^n \cdot \nabla + \dfrac{\partial \mathbf{R}_u}{\partial \mathbf{u}} & \mathbf{0} \\ 0 & 0 & 0 \end{pmatrix} \tag{25} $$

$$ {}_s\mathbf{A}_s = \begin{pmatrix} 0 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & \mathbf{u}^n \cdot \nabla + \dfrac{\partial R_p}{\partial s} \end{pmatrix} \tag{26} $$

Using these notations, we rewrite Eqs. (12)-(14) as follows.

$$ \left[\mathbf{I} + \delta t \left\{ {}_p\mathbf{A}_p + {}_p\mathbf{A}_u + {}_u\mathbf{A}_p + {}_u\mathbf{A}_u + {}_s\mathbf{A}_s \right\} \right] \delta\mathbf{q} = \delta\mathbf{q}^* \tag{27} $$

For an example, the R.H.S. can be approximately factorized as follows;

$$ \left[\mathbf{I} + \delta t \left\{ {}_u\mathbf{A}_u + {}_s\mathbf{A}_s \right\} \right] \left[\mathbf{I} + \delta t \left\{ {}_u\mathbf{A}_p + {}_p\mathbf{A}_p + {}_p\mathbf{A}_u \right\} \right] \delta\mathbf{q} = \delta\mathbf{q}^* \tag{28} $$

Errors are produced by approximate factorization in general. For example, the Eq. (28) includes the following factorization error.

$$ \begin{aligned} &\left[\mathbf{I} + \delta t \left\{ {}_u\mathbf{A}_u + {}_s\mathbf{A}_s \right\} \right] \left[\mathbf{I} + \delta t \left\{ {}_u\mathbf{A}_p + {}_p\mathbf{A}_p + {}_p\mathbf{A}_u \right\} \right] - \left[\mathbf{I} + \delta t \left\{ {}_u\mathbf{A}_u + {}_u\mathbf{A}_p + {}_p\mathbf{A}_p + {}_p\mathbf{A}_u + {}_s\mathbf{A}_s \right\} \right] \\ &= \delta t^2 \, {}_u\mathbf{A}_u \, {}_u\mathbf{A}_p \end{aligned} \tag{29} $$

Here the following relation was used;

$$ {}_a\mathbf{A}_b \, {}_c\mathbf{A}_d \begin{cases} = 0 & b \neq c \\ \neq 0 & b = c \end{cases} \tag{30} $$

Therefore, the entropy term produces no factorization error in any factorization. Thus, the entropy term will be omitted in the following discussion for brevity.

For factorization error evaluation, dimensional analysis of each differential operators is performed. Consider high *Re* number flow and neglect the influence of viscosity term.

$$ {}_p\mathbf{A}_p \delta\mathbf{q} = O\left(\mathbf{u}^n \cdot \nabla \mathbf{u} \right) = O\left(\frac{U \Delta P}{L} \right) = O\left(\frac{\rho U^3}{L} \right) \tag{31} $$

$$ {}_p\mathbf{A}_u \delta\mathbf{q} = O\left(\rho c^2 \nabla \cdot \mathbf{u} \right) = O\left(\frac{\rho C^2 U}{L} \right) \tag{32} $$

$$ {}_u\mathbf{A}_p\delta\mathbf{q} = O\left(\frac{1}{\rho}\nabla p\right) = O\left(\frac{\Delta P}{\rho L}\right) = O\left(\frac{U^2}{L}\right) \tag{33} $$

$$ {}_u\mathbf{A}_u\delta\mathbf{q} = O\left(\mathbf{u}^n\cdot\nabla\delta\mathbf{u}\right) = O\left(\frac{U^2}{L}\right) \tag{34} $$

Here, C and U are Reference scales of sound speed and advection speed, and the following relationship is used for the pressure fluctuation.

$$ O\left(p-p_\infty\right) = O\left(\rho U^2\right) \tag{35} $$

Eq. (31)(32) are for pressure equation, and Eq. (33)(34) are for velocity equation, thus, these dimensions are different. At low M (C>>U), i.e., only Eq. (32) is significantly large. Therefore, at low M, it is necessary to ensure that ${}_p\mathbf{A}_u$ does not remain in the error. There is only one factorization into three or more terms satisfying this condition;

$$ \left[\mathbf{I}+\delta t\,{}_u\mathbf{A}_u\right]\left[\mathbf{I}+\delta t\left\{{}_u\mathbf{A}_p+{}_p\mathbf{A}_u\right\}\right]\left[\mathbf{I}+\delta t\,{}_p\mathbf{A}_p\right]\delta\mathbf{q} = \delta\mathbf{q}^* \tag{36} $$

A Factorization into 2 terms can be obtained by combining the 1st and 2nd, or 2nd and 3rd terms of Eq. (36). Eq. (28) is obtained by a combination of 2nd and 3rd terms. Another possibility is to combine the 1st and 2nd;

$$ \left[\mathbf{I}+\delta t\left\{{}_u\mathbf{A}_u+{}_u\mathbf{A}_p+{}_p\mathbf{A}_u\right\}\right]\left[\mathbf{I}+\delta t\,{}_p\mathbf{A}_p\right]\delta\mathbf{q} = \delta\mathbf{q}^* \tag{37} $$

Considering the actual numerical schemes, to solve the first step of Eq. (37) is as difficult as to solve the original Eq. (27). Therefore, it is not attractive as a numerical scheme and Eq. (28) is only feasible factorization into 2 terms. On the other hand, it is feasible to solve the Eq. (36), however, more factorizations make more factorization error. (The fact that matrix of pressure becomes symmetric can be beneficial, though.)

As a result, Eqs. (12)-(14) are solved, by using approximate factorization of the differential operators expressed by Eq. (28) in two steps including the entropy variable.

First step (advection):

$$ \delta\mathbf{u}^{**}+\delta t\left\{\mathbf{u}\cdot\nabla\delta\mathbf{u}^{**}+\frac{\partial\mathbf{R}_u}{\partial\mathbf{u}}\delta\mathbf{u}^{**}\right\} = \delta\mathbf{u}^* \tag{38} $$

$$ \delta p^{**} = \delta p^* \tag{39} $$

$$ \delta s^{**}+\delta t\left\{\mathbf{u}\cdot\nabla\delta s^{**}+\frac{\partial R_p}{\partial s}\delta s^{**}\right\} = \delta s^* \tag{40} $$

Second step (pressure):

$$\delta\mathbf{u} + \delta t \frac{1}{\rho}\nabla\delta p = \delta\mathbf{u}^{**} \tag{41}$$

$$\delta p + \delta t\left(\mathbf{u}\cdot\nabla + \frac{\partial R_p}{\partial p}\right)\delta p + \delta t\rho c^2\nabla\cdot\delta\mathbf{u} = \delta p^{**} \tag{42}$$

$$\delta s = \delta s^{**} \tag{43}$$

Substituting Eq. (41) into Eq. (42), we have;

$$\delta p + \delta t\left(\mathbf{u}\cdot\nabla + \frac{\partial R_p}{\partial p}\right)\delta p + \delta t\rho c^2\nabla\cdot\left(\delta\mathbf{u}^{**} - \delta t\frac{1}{\rho}\nabla\delta p\right) = \delta p^{**} \tag{44}$$

The left side density difference is approximated to obtain the following equation;

$$\delta p + \delta t\left(\mathbf{u}\cdot\nabla + \frac{\partial R_p}{\partial p}\right)\delta p - \delta t^2 c^2\nabla\cdot\nabla\delta p = \delta p^{**} - \delta t\rho c^2\nabla\cdot\delta\mathbf{u}^{**} \tag{45}$$

The calculation procedure is summarized as follows:

i) calculate the R.H.S.; ii) convert it to an entropy variable; iii) solve Eq. (45) with δp as an only unknown variable after step 1 (Eqs. (38)-(40)); iv) solve Eq. (41) to obtain the final velocity variation; v) When the variation of the entropy variable vector is obtained, convert it again to a conservative variable and proceed to the next time step. (See Section 5)

If $c \to \infty$ and the discretization schemes of the L.H.S. and the R.H.S. are the same, this method is reduced to almost the same scheme as SMAC method [2] of incompressible CFD. Therefore, it can be regarded as generalization of SMAC method including the original one as a special example (See APPENDIX A.). Similarly, if the Newton iteration is limited to twice with first order temporal accuracy (θ=0), it is mathematically equivalent to the incompressible version of PISO (Pressure-Implicit with Splitting of Operators) [17] in a semi-discrete form, however, it does not match with the fully discretized form shown in the latter sections. In general, the pressure-based compressible CFD scheme can be regarded as a generalization of the incompressible MAC type CFD scheme, but the scheme directly using the conservation law[14]-[17] does not asymptotically approach to SMAC even if $c \to \infty$. (Since SMAC is not necessarily the best incompressible CFD scheme, this does not mean which scheme is better or worse.) In this scheme, on the other hand, simplicity of SMAC method is maintained in terms of velocity-pressure by constructing a δ-form implicit scheme employing entropy variables. This scheme is named GC-SMAC (Generalized Compressible SMAC).

In addition, the derivation of this scheme as an optimal approximate factorization shows the necessity of

SMAC calculation procedure. In other words, the reverse procedure of "advection step" after "pressure step" could have been an alternative scheme; but it is inappropriate since this causes a huge factorization error.

4 NUMERICAL METHOD FOR THE SECOND STEP

In the discretization of the L.H.S., the first step includes only advection and diffusion, and it is easy to make implicit schemes that maintains diagonal dominance by using the first order upwind scheme. (See Eq.(60)(61)) On the other hand, the second step is not that simple. It can be shown difference among many implicit schemes for compressible CFD including time derivative pre-conditioning methods are explained by differences for solving the second step. (See APPENDIX B.)

Regarding the second step, in the inviscid case, if the finite volume method is used, as a discretization method applicable to both structural and unstructured meshes, to Eq. (45), it can be written as follows;

$$\delta p_i + \frac{\Delta t}{\Omega}\mathbf{u}_i \cdot \sum_j \delta p_{i,j}\mathbf{n}_{i,j}dS_{i,j} - \frac{c^2\Delta t^2}{\Omega}\sum_j \nabla \delta p_{i,j} \cdot \mathbf{n}_{i,j}dS_{i,j} = R.H.S. \tag{46}$$

Then, the relationship and definition of the following equation are used.

$$\sum_j \mathbf{n}_{i,j}dS_{i,j} = 0 \tag{47}$$

$$V_{ni,j} \equiv \mathbf{u}_i \cdot \mathbf{n}_{i,j} \tag{48}$$

The advection of pressure is calculated by first order upwind difference to maintain the diagonal dominance and the difference at the cell boundary is defined as follows.

$$\delta p_{i,j}\mathbf{u}_i \cdot \mathbf{n}_{i,j} = \frac{V_{ni,j} + |V_{ni,j}|}{2}\delta p_i + \frac{V_{ni,j} - |V_{ni,j}|}{2}\delta p_j \tag{49}$$

$$\left(\nabla \delta p \cdot \mathbf{n}\right)_{i,j} = \frac{\delta p_j - \delta p_i}{\Delta h_{i,j}} \tag{50}$$

where $\Delta h_{i,j}$ is the projection of the distance between the cell centers to the cell-interfacial normal vector [corresponding to cell-to-cell distance in an orthogonal (Cartesian) mesh]. Using these, Eq. (46) can be written as follows.

$$\delta p_i + \frac{\Delta t}{V}\sum_j \left[\frac{|V_{ni,j}|}{2}\delta p_i + \frac{V_{ni,j} - |V_{ni,j}|}{2}\delta p_j + c^2\Delta t\frac{\delta p_i - \delta p_j}{\Delta h_{i,j}}\right] ds_{i,j} = R.H.S. \tag{51}$$

Here, the following relationship was used.

$$\sum_j V_{ni,j}dS_{i,j} = 0 \tag{52}$$

Equation (50) can be rewritten as follows.

$$\delta p_i + \frac{\Delta t}{\Omega}\sum_j \left[\frac{\sigma_{i,j}}{2}\delta p_i + \frac{\hat{V}_{ni,j} - \sigma_{i,j}}{2}\delta p_j \right] dS_{i,j} = R.H.S. \tag{53}$$

$$\sigma_{i,j} = \left| V_{ni,j} \right| + 2Cnl_{i,j}c \tag{54}$$

$$Cnl_{i,j} = \frac{c\Delta t}{\Delta h_{i,j}} \tag{55}$$

Here, C_{nl} corresponds to the local Courant number based on the sound speed. In order to discretize the R.H.S. for pressure, numerical dissipation inherent in the R.H.S. should be included, and then σ is corrected as follows including the simple handling of viscosity.

$$\sigma_{i,j} = \left| V_{ni,j} \right| + d_\mu + c\max\left(2Cnl_{i,j},1\right) \tag{56}$$

$$d_\mu = \frac{2(\mu + \mu_T)dS_{i,j}}{\rho_i \Omega_i} \tag{57}$$

The following terms necessary for the evaluation of the Eqs. (41)(42) are given in a central difference manner as follows.

$$\nabla \cdot \delta \mathbf{u}^{**} \approx \frac{1}{2\Omega}\sum_j \delta \mathbf{u}_j^{**} \cdot \mathbf{n}_{i,j} dS_{i,j} \tag{58}$$

$$\nabla \delta p \approx \frac{1}{2\Omega}\sum_j \delta p_j \mathbf{n}_{i,j} dS_{i,j} \tag{59}$$

In addition, the advection terms in the first step (Eqs.(38)(39)) are discretize similarly as follows;

$$\delta \mathbf{u}_i + \frac{\Delta t}{V}\sum_j \left[\frac{\left|V_{ni,j}\right| + d_\mu}{2}\delta \mathbf{u}_i + \frac{V_{ni,j} - \left|V_{ni,j}\right| - d_\mu}{2}\delta \mathbf{u}_j \right] ds_{i,j} = R.H.S. \tag{60}$$

$$\delta s_i + \frac{\Delta t}{V}\sum_j \left[\frac{\left|V_{ni,j}\right| + d_\mu}{2}\delta s_i + \frac{V_{ni,j} - \left|V_{ni,j}\right| - d_\mu}{2}\delta s_j \right] ds_{i,j} = R.H.S. \tag{61}$$

5 EVALUATION OF THE R.H.S. BY MUSCL-FVM

As GC-SMAC is written in δ-form, various discretization with different stability and accuracy can be used on the R.H.S. Here, we show an example of MUSCL for conservative variables.

Compressible NS equation can be written in integral form as;

$$\iiint \mathbf{Q}_t d\Omega + \oiint (\hat{\mathbf{E}} - \hat{\mathbf{R}}) dS = 0 \tag{62}$$

In this paper we focus on the evaluation of the inviscid flux. The inviscid flux can be written as;

$$\hat{\mathbf{E}} = \dot{m}\mathbf{\Phi} + p\mathbf{N} \tag{63}$$

$$\dot{m} = \rho V_n \tag{64}$$

$$\mathbf{\Phi} = \left(1, u, v, w, h\right)^T \tag{65}$$

$$\mathbf{N} = \left(0, x_n, y_n, z_n, 0\right)^T \tag{66}$$

$$h = (e+p)/\rho \tag{67}$$

By applying this to the computational cell of polyhedron (polygon in two dimensions), the basic equation of the finite volume method common to the structured/unstructured mesh can be obtained.

$$\overline{\mathbf{Q}}_t + \frac{1}{\Omega_i}\sum_j (\widetilde{\mathbf{E}}_{i,j} - \widetilde{\mathbf{R}}_{i,j}) dS_{i,j} = 0 \tag{68}$$

Eq. (68) is exact if the flux is an accurate average. MUSCL realizes the higher-order accuracy by inner-cell reconstruction and generally calculates inviscid flux using Riemann flux from right and left physical quantities of cell boundary (including discontinuity). The R.H.S. of the entropy variable, Eqs. (18)-(20), is obtained from the above-mentioned discretization and variable transformation in which the entropy variable vector is defined as **s**, as follows. If Newton iteration converges, R.H.S. = 0 is valid.

$$\delta\mathbf{s}^* \equiv \frac{\partial \mathbf{s}}{\partial \mathbf{Q}}\left[\mathbf{Q} - \theta_1 \mathbf{Q}^n + \theta_2 \mathbf{Q}^{n-1} - \delta t \frac{1}{\Omega_i}\sum_j (\widetilde{\mathbf{E}}_{i,j} - \widetilde{\mathbf{R}}_{i,j}) dS_{i,j}\right] \tag{69}$$

When compared with an implicit scheme with the direct use of conservative variables, variable transformation and inverse transformation are required before and after the calculation of L.H.S., but this is only a small part of the whole implicit method computation. Since the calculation of L.H.S. is simpler in this case than that for conservative variables, the total computational amount rather decreases even if the conversion to entropy variable vector is combined[9].

6 SLAU TYPE NUMERICAL FLUX SCHEME

A Riemann flux is based on SLAU [10] which is a simple and robust AUSM [6] family, having less dissipation at low *M*. The Riemann flux of the AUSM family schemes can be written in the following form.

$$_{AUSM-Family}\widetilde{\mathbf{E}} = \frac{\dot{m} + |\dot{m}|}{2}\mathbf{\Phi}_L + \frac{\dot{m} - |\dot{m}|}{2}\mathbf{\Phi}_R + \tilde{p}\mathbf{N} \tag{70}$$

Transport due to mass flux is separated from pressure, and it is upwinded by its sign. Various AUSM family fluxes are defined by differences in mass flux and interfacial pressure. In SLAU, it is defined as follows. (Countermeasures against strong asymmetric expansion are omitted.)

$$_{SLAU}\dot{m} = \mathrm{ave}(\rho V_n) - \left|\overline{V}_n\right| \mathrm{diff}(\rho) - \frac{f_p}{\overline{c}} \mathrm{diff}(p) \tag{71}$$

$$_{SLAU}\widetilde{p} = \mathrm{ave}(p) - \left(\beta_L^+ - \beta_R^-\right)\mathrm{diff}(p) + (1-\chi)\left(\beta_L^+ + \beta_R^- - 1\right)\mathrm{ave}(p) \tag{72}$$

here,

$$\mathrm{ave}\,(q) = \frac{1}{2}(q_R + q_L) \tag{73}$$

$$\mathrm{diff}\,(q) = \frac{1}{2}(q_R - q_L) \tag{74}$$

$$\overline{c} = \mathrm{ave}(c) \tag{75}$$

$$V_n = \mathbf{u} \cdot \mathbf{n} \tag{76}$$

$$M = V_n / \overline{c} \tag{77}$$

$$\left|\overline{V}_n\right| = \mathrm{ave}\left(\rho |V_n|\right) / \mathrm{ave}\left(\rho\right) \tag{78}$$

$$\beta_{L/R}^{\pm} = \begin{cases} \dfrac{1}{4}(2 \mp M_{L/R})(M_{L/R} \pm 1)^2, & |M_{L/R}| < 1 \\ \dfrac{1}{2}(1 + sign(\pm M_{L/R})) & ,\text{otherwise} \end{cases} \tag{79}$$

χ is defined as follows, $\chi = 1 \rightarrow 0$ when $M = 0 \rightarrow 1$, and controls numerical dissipation at subsonic speeds.

$$\widehat{M} = \min\left(1.0, \frac{1}{\overline{c}}\sqrt{\mathrm{ave}(|\mathbf{u}|^2)}\right) \tag{80}$$

$$\chi = (1 - \widehat{M})^2 \tag{81}$$

7 IMPROVEMENT OF SLAU FOR DAMPING OF PRESSURE NOISE

The problem in case of $M << 1$ is how to define the dimensionless coefficient f_p of the pressure difference term in the mass flux. In the original SLAU, it is defined as follows, and for low M, $f_p \approx 1$.

$$f_p = \chi \tag{82}$$

The pressure difference term in the mass flux acts as numerical dissipation for pressure through density variation suppression. Considering the case of one-dimensional, spatially first order accuracy with zero fluid velocity, as an example. Density variation according to the mass flux difference is written as;

$$\rho_t \approx \frac{f_p}{2\overline{c}\Delta x}(p_{i+1} - 2p_i + p_{i-1}) \approx \frac{\Delta x f_p}{2\overline{c}} p_{xx} \tag{83}$$

Furthermore, assuming isentropy;

$$p_t = c^2 \rho_t \approx \frac{\overline{c}\Delta x f_p}{2} p_{xx} \tag{84}$$

This is the diffusion equation of pressure, and it can be seen that f_p controls the diffusion amount of pressure. Eq. (84) has the following analytical solution. Here, T_d is the damping characteristic time.

$$p = p_0 \exp(-\frac{t}{T_d} + ikx) \tag{85}$$

$$T_d = \frac{2}{c\Delta x f_p k^2} \tag{86}$$

Substituting the relation of sound waves with T as the period and L as the wavelength;

$$T_d = \frac{1}{2\pi^2 f_p}\frac{L}{\Delta x}T \tag{87}$$

This means that if f_p is a constant, the damping characteristic time depends on the grid size: The damping is fast (strong) in the coarse grid, and it is slow (weak) in the fine part even for unnecessary disturbances. Therefore, we introduce the following form of f_p using the control variable T_c of the time dimension, in which the damping characteristic time is independent of the grid size.

$$f_p = \frac{cT_c}{\Delta x} \tag{88}$$

When this is used, the damping characteristic time can be expressed as follows.

$$T_d = \frac{2}{c\Delta x f_p k^2} = \frac{2}{c^2 T_c k^2} = \frac{1}{2\pi^2}\frac{T}{T_c}T \tag{89}$$

Therefore, the damping characteristic time is sufficiently longer than the period, irrespective of the grid size, if $T_c << T$.

Therefore, if T_c is sufficiently smaller than the period of sound waves, the damping characteristic time becomes sufficiently longer than the period, and the sound wave can be calculated without strong damping. Conversely, when the period is order of T_c or less, the damping increases irrespective of the grid size, and the effect of noise suppression is expected. Also, no matter how small the T_c is, we cannot calculate the waves of the grid size or less, and hence, considering the smooth connection from the original SLAU, we will use the following.

$$f_p = \chi \max\left(1, \frac{\bar{c}T_c}{\Delta x}\right) \tag{90}$$

Since the damping of sound waves becomes uniform irrespective of the grid size, we call this numerical flux scheme UD (Uniform Damping)-SLAU.

It is obvious that sound waves with a period shorter than the time step of the unsteady computation cannot be resolved, and thus, it is desirable that such sound waves be damped since they can only be noises. Therefore,

in the case of unsteady computation, we define the following as the default value of T_c;

$$T_c \approx \Delta t \tag{91}$$

As is clear from the expression Eq. (90), this selection also means recovering the original SLAU when the Courant number based on the speed of sound is 1 or less, and this is natural in terms of smooth transition with the high-speed side scheme. Note that numerical experiments have shown that variable T_c led to numerical oscillations, therefore, T_c needs to be fixed even when allowing the variation of Δt.

8 COMBINATION OF UD-SLAU WITH IMPLICIT TIME INTEGRATION METHOD

In UD-SLAU, the numerical dissipation in the R.H.S. will be strengthened. Therefore, the spectral radius in L.H.S should be also augmented. For example, the spectral radius used in LU-SGS[5] implicit scheme etc. should be modified as follows;

$$|V| + c + d_\mu \rightarrow |V| + c \max\left(1, \frac{\overline{c}T_c}{\Delta x}\right) + d_\mu \tag{92}$$

On the other hand, in the case of GC-SMAC, it can be seen, from the comparison with Eqs. (56),(57), stability is guaranteed by defining as follows;

$$\sigma_{i,j} = \left|V_{ni,j}\right| + d_\mu + c \max\left(\frac{\overline{c}T_c}{\Delta x}, 2Cnl_{i,j}, 1\right) \tag{93}$$

From the comparison of Eqs. (54)(55) and (90), if the following holds, the GC - SMAC is stable without any change.

$$T_c < 2\Delta t \tag{94}$$

As a result, if the definition of Eq. (91) is applied, the spectral radius used in GC-SMAC is simply denoted as;

$$\sigma_{i,j} = \left|V_{ni,j}\right| + d_\mu + c \max\left(2Cnl_{i,j}, 1\right) \tag{95}$$

Therefore, the compatibility of UD-SLAU with GC-SMAC is excellent.

9 NUMERICAL EXAMPLES

9.1 Inviscid flow around two-dimensional airfoil (NACA0012)

As an example of convergence acceleration to steady solutions, we show convergence in an inviscid flow around NACA0012 airfoil. An O-type mesh is used and the time step is set so that the maximum Courant number based on the advection speed becomes about 1. The temporal accuracy is the first-order, in which

Newton iteration is limited to once, without the local time method.

Figures 1-3 show the convergence history of the velocity field at $M = 0.5, 0.1, 0.01$ by schemes combining SLAU& MFGS [10] (without time derivative pre-conditioning), SLAU&TC-PGS1[9], which is an implicit scheme having flavor of the time derivative pre-conditioning method and the fastest in our previous study[9], SLAU&GC-SMAC and UD-SLAU&GC-SMAC (SMUC). In each case the number of time steps is fixed at 10000, and the horizontal axis shows CPU time. Here MFGS is similar to LU-SGS with more inner iteration where LU-SGS[5] has only one symmetric sweep.

At $M = 0.5$, the convergence history and the computational time per step do not differ too much. As the flow speed is going down to $M = 0.1, 0.01$, the superiority of GC-SMAC becomes clear. MFGS's computational time is long because internal Gauss-Seidel iteration is less likely to converge and the number of required iterations for the convergence [i.e., the residual being 1/10 of the initial value] is increasing. SSOR is used in GC - SMAC.

Figures 4 and 5 show the pressure distribution around the 10% chord length of the airfoil trailing edge for SLAU&GC-SMAC and SMUC (UD-SLAU&GC-SMAC) at M = 0.01. With SLAU, wiggles due to grid distortion due to O-type mesh is generated, but in UD - SLAU, they are eliminated by damping of disturbance in small cells near the airfoil.

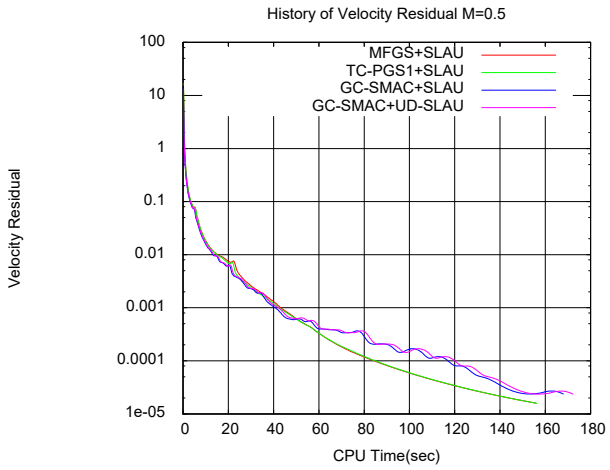

Figure 1 History of velocity residual of flow around NACA0012 at Mach=0.5.

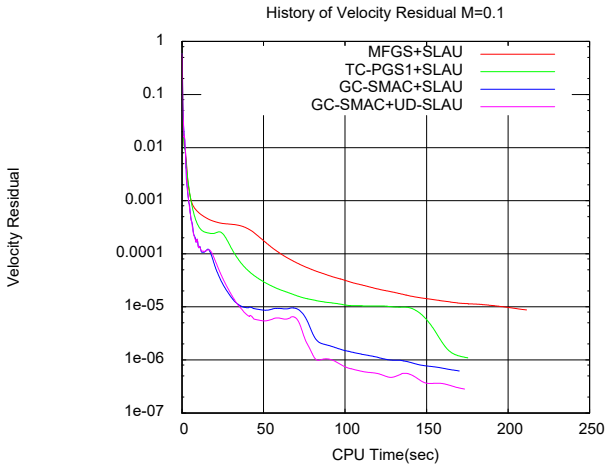

Figure 2 History of velocity residual of flow around NACA0012 at Mach=0.1.

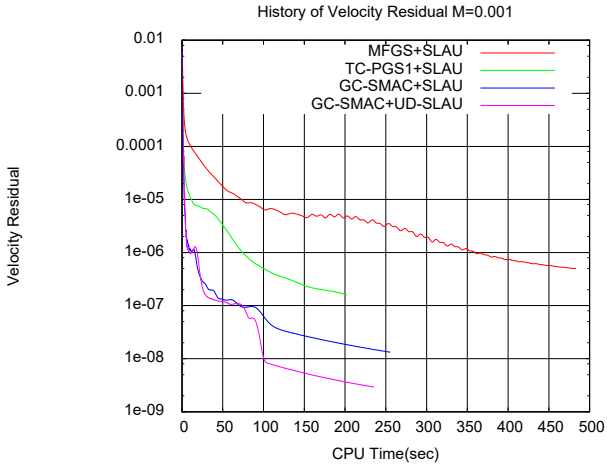

Figure 3 History of velocity residual of flow around NACA0012 at Mach=0.01.

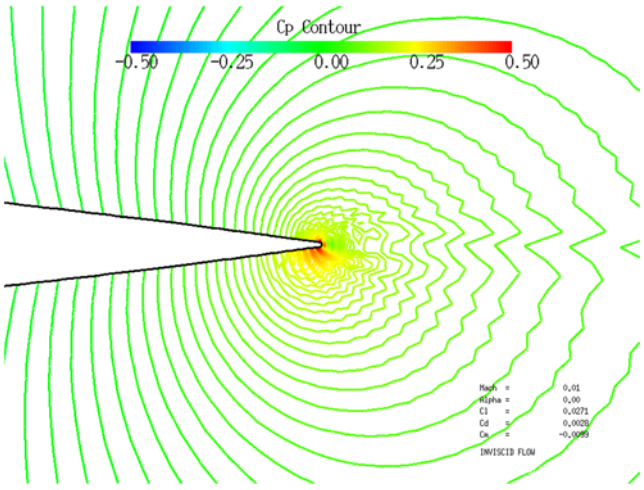

Figure 4 Enlarged view of the contour of the pressure coefficient at 10% chord of the trailing edge of NACA0012 airfoil computed using GC-SMAC and SLAU. Mach=0.01.

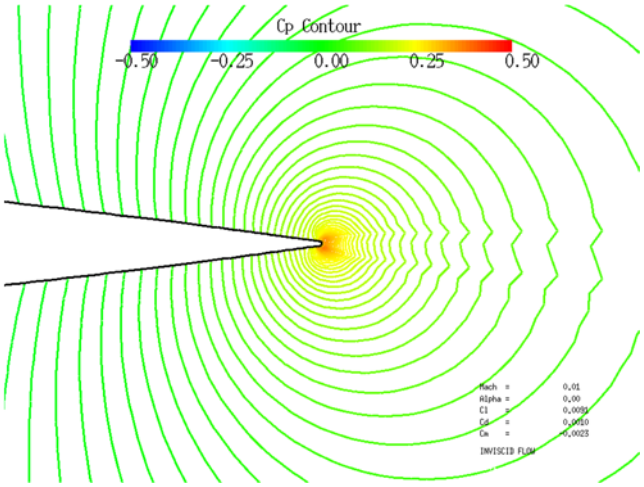

Figure 5 Enlarged view of the contour of the pressure coefficient at 10% chord of the trailing edge of NACA0012 airfoil computed using GC-SMAC and UD-SLAU (SMUC). Mach=0.01.

9.2 Computation of 1D sound wave by implicit methods

Consider propagation of sound generated by low M fluid motion. In this case, since the frequency of the sound is the same as that of the fluid motion, the time step determined considering the flow field resolution is sufficiently smaller than the sound period. On the other hand, if the Strouhal number is constant, the wavelength of the sound is proportional to 1 / M. Therefore, the grid size needed for resolving the flow is much smaller than the wavelength. In addition, if the time step size is constant, Courant number based on sound speed increases in proportion to 1 / M.

Therefore, in this case, the Courant number is large, but the time / space step is sufficiently smaller than the sound wave period / wavelength. A validation under such a condition will be shown considering calculation of sound waves by the implicit scheme. Here, an example of a one-dimensional acoustic wave (third order accuracy in space and second order accuracy in time) with 400 cells per wavelength, Courant number 10 ($T/\Delta t$ = 40) is shown.

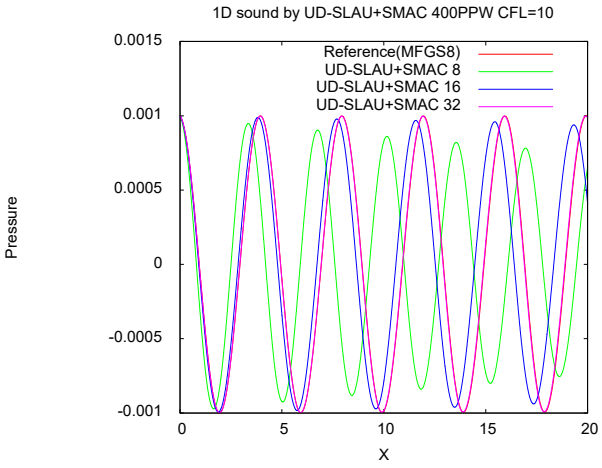

Figure 6 1D sound propagation computed using SMUC.

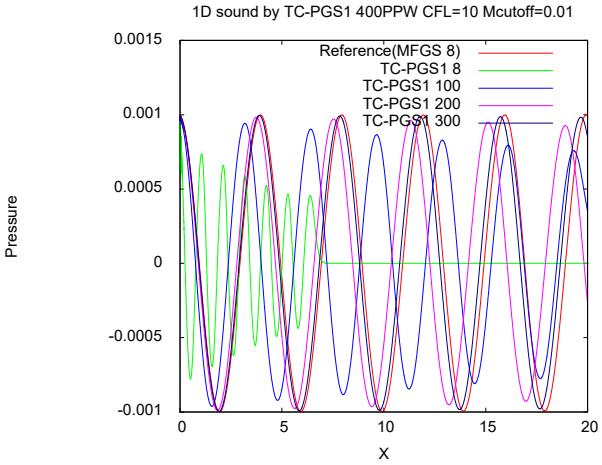

Figure 7 1D sound propagation computed using TC-PGS1.

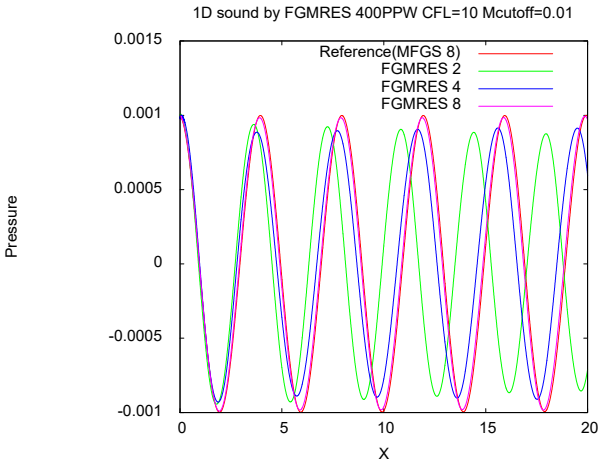

Figure 8 1D sound propagation computed using FGMRES.

As a reference value, a combination of MFGS implicit scheme and Newton iteration of 8 iterations is used. This is an example of the least numbers of Newton iterations resulting in almost the exact solution.

Figure 6 shows the result of changing numbers of Newton iterations of SMUC from 8, 16, to 32. Table 1 also

shows the summary of each case and the normalized computational time. SMUC requires more than 32 iterations. This is because implicit dissipation for sound waves included in SMUC is stronger than MFGS. However, MFGS is a scheme for high Mach number, and SMUC is not inferior in overall calculation efficiency in the simultaneous analysis for low Mach number flows. Considering computations for the flow field, MFGS requires Newton iterations several times more than SMUC for time accuracy for the flow field at low M, and hence, SMUC is more efficient overall. This is similar, even if Riemann flux is changed to SLAU.

The advantage of SMUC in the simultaneous analysis with low M flow is obvious when compared with the result of SLAU&TC-PGS1, using the concept of temporal derivative pre-conditioning method, as shown in Fig. 7. Here, $M_c = 0.01$ is set in consideration of application to a low M flow field. It takes 300 Newton iterations to obtain the same results as MFGS. As M_c becomes smaller, this situation becomes more severe. This is a big disadvantage compared to SMUC which does not require M_c. Also, it is a disadvantage that the influence of M_c remains even if the Courant number is reduced.

Figure 8 shows the results of F(Flexible)GMRES using TC-PGS 1 ($M_c = 0.01$) for matrix preconditioning, which showed the best efficiency in Ref. (9). FGMRES converges at about 8 iterations. The calculation time is about 63% of SMUC, which is the fastest in this case. However, since TC-PGS 1 is used for matrix preprocessing, it is necessary to adjust M_c in the same way. If the M further decreases, deterioration in calculation efficiency cannot be avoided.

In summary, SMUC can be said to have a great benefit in the simultaneous analysis of low M flow and sound waves in the sense that the adjustment of M_c is unnecessary and that the convergence improves automatically due to the decrease in numerical dissipation, when the local Courant number decreases.

Table 1 Schemes and relative CPU time of 1D sound propagation cases.

CASE	Scheme	#Newton	Rel. CPU
MFGS 8	SLAU+MFGS	8	1.00
UD-SLAU+SMAC 8	SMUC	8	1.21
UD-SLAU+SMAC 16	SMUC	16	2.42
UD-SLAU+SMAC 32	SMUC	32	4.83
TC-PGS1 8	SLAU+TC-PGS1	8	1.00
TC-PGS1 100	SLAU+TC-PGS1	100	13.06
TC-PGS1 200	SLAU+TC-PGS1	200	26.42
TC-PGS1 300	SLAU+TC-PGS1	300	39.90
FGMRES 2	SLAU+FGMRES	2	0.78
FGMRES 4	SLAU+FGMRES	4	1.53
FGMRES 8	SLAU+FGMRES	8	3.07

10 SUMMARY AND FUTURE CHALLENGES

We presented formulation and numerical examples of SMUC (SMAC-inspired Mach Uniform Compressible scheme) scheme introducing the idea of SMAC method into framework of *δ-form* implicit scheme and MUSCL finite volume method.

- SMUC scheme consists of a combination of an implicit scheme (GC-SMAC) and a Riemann flux (UD-SLAU) in the framework of the MUSCL-FVM.
- GC-SMAC can compute very low Mach number flows stably without adjustment of flow-field-dependent parameters such as M_c, unlike the scheme based on application of time derivative pre-conditioning.
- With respect to computational efficiency, its superiority is clear to conventional schemes, including time derivative pre-conditioning method when M <0.1 or less, while efficiency equivalent to MFGS etc. above subsonic speed being maintained. It is a great benefit that several times faster speed can be realized around M=0.1, which is frequently used in low Mach number aerodynamic problems.
- In UD-SLAU, the pressure difference term in the mass flux term, which is the key to the low Mach number characteristic of the SLAU type scheme, has been corrected in a manner highly compatible with GC-SMAC. The adjustment of M_c, which was necessary for the stabilization, was removed. Furthermore, numerical examples show that simultaneous computations of low M flows and sound waves are possible with the same parameters.

The following issues can be raised as the future work.

- Although we could not fully describe it in this paper, MFGS, TC-PGS 1 and GC-SMAC have different properties with respect to internal linear iterations, and the knowledge obtained in MFGS is not sufficient. In MFGS, over-relaxation is not effective and 20 or fewer iterations to reduce the residual to 1/10 of its initial value is the optimum. On the other hand, over-relaxation is effective for GC - SMAC. Regarding the number of iterations, it takes 50-250 SSOR iterations to convergence to 1/10 of the initial residual.
- In the pressure step, the current scheme uses a simple SSOR method, but it may be effective to use a more sophisticated method used in the pressure-based schemes.
- Since the verification is limited to the basic flow field and sound propagation, demonstrations for a more realistic and complex fluid-acoustic coupling problem is necessary.

REFERENCES

[1] Harlow, F. H. and Welch, J. E.: Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface, Physics of Fluids, 8.12 (1965) 2182-2129.

[2] Amsden, A.A. and Harlow, F. H.: A simplified MAC technique for incompressible fluid flow calculations, Journal of Computational Physics, 6.2 (1970) 322-325.

[3] Beam, R.M., and Warming, R.F.: An implicit factored scheme for the compressible Navier-Stokes equations. AIAA journal, 16.4 (1978) 393-402.

[4] van Leer, B.: Towards the Ultimate Conservative Difference Scheme, V. A Second Order Sequel to Godunov's Method, Journal of computational physics, 32 (1979) 101–136.

[5] Jameson, A. and Turkel, E.: Implicit schemes and *LU* decompositions, Mathematics of Computation, 37.156 (1981) 385-397.

[6] Liou, M.-S. and Steffen, C.J. Jr.: A new flux splitting scheme, Journal of Computational Physics, 107.1 (1993) 23–39.

[7] Turkel, E.: Preconditioned methods for solving the incompressible and low speed compressible equations, Journal of Computational Physics, 72.2 (1987) 277-298.

[8] Weiss, J.M. and Smith, W.A.: Preconditioning Applied to Variable and Constant Density Flows, AIAA journal, 33.11 (1995) 2050-2057.

[9] Shima, E. and Kitamura, K.: New approaches for computation of low Mach number flows, Computers & Fluids, 85 (2013) 143-152.

[10] Shima E and Kitamura K.: Parameter-free simple low-dissipation AUSM-family scheme for all speeds, AIAA journal, 49.8 (2011) 1693-1709.

[11] Shima, E.: On the improvement of the all-speed flux scheme for very low Mach number flows, AIAA Paper 2013-2696 (2013).

[12] Kawamura, T. and Kuwahara, K.: Computation of High Reynolds Number Flow Around a Circular Cylinder with Surface Roughness, AIAA Paper 84-0340 (1984).

[13] Morinishi, Y., Lund, T. S., Vasilyev, O. V., & Moin, P.: Fully conservative higher order finite difference schemes for incompressible flow, Journal of computational physics, 143.1 (1998) 90-124.

[14] Harlow, F.H, Amsden, A.A.: Numerical calculation of almost incompressible flow, Journal of Computational Physics, 3 (1968) 80-93.

[15] Harlow, F.H, Amsden, A.A.: A numerical fluid dynamics calculation method for all flow speed, Journal of Computational Physics, 8 (1971) 197-213.

[16] Karki, K.C., Patanker, S.V.: Pressure based calculation procedure for viscous flows at all speed in arbitrary configurations, AIAA Journal, 27 (1989) 1167-174.

[17] Issa, R.I.: Solution of the Implicitly Discretized Fluid Flow Equations by the Operator-Splitting, Journal of Computational Physics, 62 (1985) 40–65.

[18] Inagaki, M., Murata, O. and Kondoh, T.: Numerical prediction of fluid-resonant oscillation at low Mach number, AIAA journal, 40.9 (2002) 1823-1829.

APPENDIX A. Similarity with SMAC method

In the GC-SMAC shown in Section 2, we will use a temporal first-order implicit method with only once of Newton iteration. By setting the value at the previous step as the initial value, and substituting the Eq. (18)for Eq.(38) in the first step, the following is obtained;

$$\delta\mathbf{u}^{**} + \delta t\left\{\mathbf{u}^n \cdot \nabla \delta\mathbf{u}^{**} + \frac{\partial \mathbf{R_m}}{\partial \mathbf{u}}\delta\mathbf{u}^{**}\right\} = -\delta t\left\{\mathbf{u}^n \cdot \nabla \mathbf{u}^n + \frac{1}{\rho}\nabla p^n + \mathbf{R_m}^n\right\} \tag{A1}$$

Assuming the consistency of the left and right differentiation operators and the linearity of the viscous term, we can rewrite it as follows.

$$\mathbf{u}^{**} + \delta t\left\{\mathbf{u}^n \cdot \nabla \mathbf{u}^{**} + \frac{1}{\rho}\nabla p^n + \mathbf{R_m}^{**}\right\} = 0 \tag{A2}$$

where the following definitions are used;

$$\mathbf{u}^{**} \equiv \mathbf{u}^n + \delta\mathbf{u}^{**} \tag{A3}$$

It corresponds to advection step of SMAC method treating advection and diffusion implicitly.

Similarly, in the second step, substituting the Eqs. (19)(A3) into the Eq. (42) gives

$$\delta p + \delta t\left(\mathbf{u}^n \cdot \nabla + \frac{\partial R_p}{\partial p}\right)\delta p - \delta t^2 c^2 \nabla \cdot \nabla \delta p = -\delta t\left\{\mathbf{u}^n \cdot \nabla p^n + \rho c^2 \nabla \cdot \mathbf{u}^{**} + R_p{}^n\right\} \tag{A4}$$

If the left and right operators match and the viscosity term is linear, we can rewrite it as follows.

$$\delta p + \delta t\left(\mathbf{u}^{n} \cdot \nabla p^{n+1} + R_{p}^{n+1}\right) - \delta t^{2} c^{2} \nabla \cdot \nabla \delta p = -\delta t \rho c^{2} \nabla \cdot \mathbf{u}^{**} \tag{A5}$$

In the case of $c >> u$, only the term multiplied by c^2 remains, and it coincides with the pressure step of SMAC method as described below.

$$\nabla \cdot \nabla \delta p = \frac{\rho}{\delta t} \nabla \cdot \mathbf{u}^{**} \tag{A6}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^{**} - \frac{\delta t}{\rho} \nabla \delta p \tag{A7}$$

Therefore, GC-SMAC includes the SMAC method of incompressible CFD as a special case.

APPENDIX B. Relationship with other implicit schemes and time derivative preconditioning schemes

With the two-step method expressed by Eq. (38)-(43), there is no difficulty in numerical handling in the first step, and the treatment of the second step is the key. In this section, we show that the difference in the numerical methods for this second step produces a difference between SMAC-based pressure-based method and density-based compressible methods such as LU-SGS[5] method. What is important is the diagonal dominance of the coefficient matrix when discretizing Eq. (41)(42). The diagonal dominance is essential in classical methods such as SOR. Also, when using the Krylov subspace methods such as GMRES, matrix preconditioning is indispensable for efficient computation, where the diagonal dominance is still required.

In the SMAC method and the GC-SMAC method, by substituting Eq. (41) of the second step into Eq. (42), a Poisson-like equation having pressure as an only unknown variable is created. For incompressible flows, it becomes a pure Poisson equation, in which diagonal dominance is automatically satisfied; even in the case of compressible flows, diagonal dominance can be realized if the advection of pressure is upwinded. Thus, various linear solvers can be used.

In the GC-SMAC method, it is understood that the discrete equations with the coupled pressure of both sides (of the cell-interface) are obtained by evaluating Eq.(41) not at the cell center, but at the cell boundary (Eqs. (50)(58)), when substituting the Eq. (41) into the Eq. (42). Such a selection is possible, since the Eqs. (41),(42) are coupled before discretization of the differential equation.

On the other hand, in the compressible implicit schemes such as LU - SGS, velocity and pressure are solved simultaneously after discretizing the Eqs. (41),(42). In the discretization by the central difference, it becomes

diagonal dominance only under the CFL-like condition, and this is not realistic in the low M where the difference between the sound speed and the advection speed is large. For unconditional diagonal dominance, it is necessary to add numerical dissipation (=upwinding) according to the characteristic speed. To illustrate this, we consider the approximate factorization to the three terms Eq. (36) shown in the Section 3. Again, how to solve the second term is the key.

The second step is expressed as follows.

$$\delta\mathbf{u}^{***} + \delta t \frac{1}{\rho}\nabla\delta p^{***} = \delta\mathbf{u}^{**} \tag{B1}$$

$$\delta p^{***} + \delta t \rho c^2 \nabla\cdot\delta\mathbf{u}^{***} = \delta p^{**} \tag{B2}$$

Let us consider a one-dimensional case. Eqs. (C1, C2) can be expressed in matrix notation as follows.

$$\begin{pmatrix}\delta u^{***}\\ \delta p^{***}\end{pmatrix} + \delta t\mathbf{A}\begin{pmatrix}\delta u^{***}\\ \delta p^{***}\end{pmatrix}_x = R.H.S \tag{B3}$$

$$\mathbf{A} = \begin{pmatrix}0 & \frac{1}{\rho}\\ \rho c^2 & 0\end{pmatrix} \tag{B4}$$

The eigenvalues are as follows, and it is understood that the Eqs. (C1, C2) correspond to the equation of sound waves.

$$\lambda_{\pm} = \pm c \tag{B5}$$

Next, some examples of compressible implicit schemes are shown.

In case of LU-SGS

In LU - SGS, by using the spectral radius (sound speed in this case), it is upwinded for diagonal dominance. The upwinded flux Jacobian can be written as follows.

$$\mathbf{A}^{\pm} = \frac{\mathbf{A} \pm c\mathbf{I}}{2} \tag{B6}$$

A similar method for diagonal dominance is applicable even in a method without factorization (the eigenvalues are different, though), and then, a concise method can be realized. On the other hand, due to this

modification, numerical dissipation, whose coefficient is sound speed, is added. When $U \ll c$, this numerical dissipation is excessive for the velocity field, and the numerical dissipation becomes dominant. Such a problem does not occur at high M, but at low M, it causes strong damping on the velocity field.

In the case of Weiss-Smith's time derivative pre-conditioning method

Weiss-Smith[6]'s time derivative pre-conditioning method is equivalent to the scaling of the variation of the pressure field in this system, and the coefficient matrix becomes as follows.

$$\mathbf{A}' = \begin{pmatrix} 0 & \frac{1}{\rho} \\ \varepsilon^2 \rho c^2 & 0 \end{pmatrix} \tag{B7}$$

By this change, eigenvalues also change as follows;

$$\lambda'_{\pm} = \pm \varepsilon c \tag{B8}$$

Using these eigenvalues, diagonal dominance can be realized as LU-SGS. If ε is the order of the M, the eigenvalues will also be on the order of the advection speed and the numerical dissipation will be an appropriate scale for the velocity field. However, it corresponds to making the fluctuation rate of the pressure field ε^2 times, so that the physical temporal evolution of pressure, that is, sound waves, cannot be calculated accurately.