48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition
4 - 7 January 2010, Orlando, Florida

AIAA 2010-1272

Performance of Low-Dissipation Euler Fluxes and Preconditioned Implicit Schemes in Low Speeds

Keiichi Kitamura,[*] Keiichiro Fujimoto[†] and Eiji Shima[‡]
JAXA, Sagamihara, Kanagawa, 229-8510, Japan

and

Z.J. Wang[§]
Iowa State University, Ames, IA 50011

In low speed flow computations, compressible finite-volume solvers are known to a) fail to converge in acceptable time and b) reach unphysical solutions. These problems are known to be cured by A) preconditioning on the time-derivative term, and B) control of numerical dissipation, respectively. There have been several methods of A) and B) proposed separately. However, it is unclear which combination is the most accurate, robust, and efficient for low speed flows. We carried out a comparative study of several well-known or recently-developed low-dissipation Euler fluxes coupled with a preconditioned LU-SGS (Lower-Upper Symmetric Gauss-Seidel) implicit time integration scheme to compute steady flows. Through a series of numerical experiments, accurate, efficient, and robust methods are suggested for low speed flow computations.

Nomenclature

c	=	speed of sound
C_D	=	drag coefficient
$\mathrm{c_p}$	=	specific heat at constant pressure
C_p	=	pressure coefficient
c_2	=	pressure stabilization coefficient in All-Speed-Roe, 0.05
δ	=	minimum spacing of grid, 1.e-3 in viscous cases
E	=	total energy
F, **Fv**	=	inviscid (Euler) and viscous flux vectors
ε	=	preconditioning coefficient
γ	=	specific heat ratio, 1.4
$\mathbf{\Gamma}$	=	preconditioning matrix
H	=	total enthalpy
i, j	=	cell indices
K	=	coefficient in preconditioning matrix, 1.0
κ	=	thermal conductivity, $\kappa = \mu \mathrm{c_p}/\mathrm{Pr}$
M	=	Mach number
$\dot{m}$	=	mass flux, $\dot{m} = \rho u$
μ	=	molecular viscosity
$\boldsymbol{n}$	=	normal vector to the cell-interface, $(n_x, n_y, n_z)^T$
p	=	pressure

[*] Researcher, JAXA's Engineering Digital Innovation (JEDI) Center, 3-1-1 Yoshinodai, and AIAA Member.
[†] Researcher, JAXA's Engineering Digital Innovation (JEDI) Center, 3-1-1 Yoshinodai (currently at 2-1-1 Sengen, Tsukuba, Ibaraki, 305-8505, Japan), and AIAA Member.
[‡] Senior Researcher and Director, JAXA's Engineering Digital Innovation (JEDI) Center, 3-1-1 Yoshinodai, and AIAA Member.
[§] Professor, Department of Aerospace Engineering, 2271 Howe Hall, and AIAA Associate Fellow.

$\widetilde{p}$	=	pressure flux
Pr	=	Prandtl number, 0.72
$\mathbf{Q}$	=	(conservative) state vector
Re	=	Reynolds number
ρ	=	density
T	=	temperature
u, v, w	=	velocity components in x, y, z-directions, respectively
x, y, z	=	Cartesian coordinates
V_i	=	volume of cell i
V_n	=	velocity component normal to the cell-interface, $V_n = (u, v, w)\boldsymbol{n} = un_x + vn_y + wn_z$
$\overline{(\ )}$	=	arithmetic averaged value
(^)	=	Roe averaged value
( )'	=	preconditioned value

Subscripts

co	=	cutoff
L, R	=	left and right running wave components
∞	=	freestream condition
*	=	maximum value in whole computational domain in All-Speed-Roe
1/2	=	cell-interfacial value

Superscripts

m	=	value at m-th sub-iteration
n	=	value at n-th timestep
*	=	critical value in AUSM-type schemes

I. Introduction

IN recent years, compressible finite-volume methods (FVMs) have been used in a wide range of flow regimes, even for low speed flows in which compressibility plays no significant role. The application of compressible flow solvers to low speeds has been motivated by the fact that users need only slight modifications to the existing (compressible) codes for computations of such low speed flows, and that this extension has the following potential applications of engineering interests:

- Analysis of flows involving both low speeds (M<0.1) and high speeds ($M\approx$ 10 or even 100), e.g., a cavitating flow in a rocket engine [1,2]
- Aeroacoustic analysis in low speed flows [3]

When applied to low speed flow computations, however, compressible solvers are known to a) fail to converge in acceptable time (stiffness problem), and b) reach unphysical solutions. These problems are known to be cured by A) preconditioning on the time-derivative term so that acoustic wave speed is properly scaled, and B) control of dissipation in numerical fluxes, respectively. There have been several methods of A) [4,5] and B) [6-9] proposed separately. However it is unclear which combination is the most accurate, robust, and efficient in low speed flows. It is difficult to prove this mathematically because, for instance, the amount of dissipation added to the computation is dependent not only on the adopted methods, but on the computational grid, flow conditions, and so forth. If a combination of methods A) and B) has insufficient dissipation for the given conditions, the calculation will suffer from numerical oscillation/instability, and may eventually diverge. If the method is too dissipative, on the other hand, its accuracy is significantly lost.

Therefore, in the present paper, we pursue an experimental approach by performing a comparative study of different methods of A) along with B) for different grids and different flow conditions of low speeds. We will pay particular attention to several well-known or recently-developed low-dissipation Euler fluxes coupled with a preconditioned LU-SGS (Lower-Upper Symmetric Gauss-Seidel) implicit scheme [10,11] in the framework of steady flows. Similar comparisons have already been conducted by others (in [12], for example), but their discussions were limited to only a few methods/cases and lacked concrete conclusions. In this study, through an extensive series of numerical experiments, accurate, efficient, and robust methods among 12 different approaches will be suggested for low speed flow computations.

II. Numerical Methods and Flow Conditions

A. Governing Equations

The governing equations are the compressible Navier-Stokes equations as follows, including the preconditioning matrix $\mathbf{\Gamma}$ of Weiss and Smith [4]. $\mathbf{\Gamma}$ is given in the Appendix A, and we keep using conservative variables as dependent variables as in [8,13], instead of using primitive ones. In the non-preconditioned form, on the other hand, $\mathbf{\Gamma}$ is simply eliminated.

$$\mathbf{\Gamma}\frac{\partial \mathbf{Q}}{\partial t}+\frac{\partial \mathbf{F}_k}{\partial x_k}=\frac{\partial \mathbf{Fv}_k}{\partial x_k} \tag{2.1a}$$

$$\mathbf{Q}=\begin{bmatrix}\rho\\ \rho u_l\\ \rho E\end{bmatrix},\quad \mathbf{F}_k=\begin{bmatrix}\rho u_k\\ \rho u_l u_k+p\delta_{lk}\\ \rho u_k H\end{bmatrix},\quad \mathbf{Fv}_k=\begin{bmatrix}0\\ \tau_{lk}\\ u_m\tau_{mk}+\kappa\dfrac{\partial T}{\partial x_k}\end{bmatrix} \tag{2.1b}$$

$$\tau_{lk}=\mu\left(\frac{\partial u_l}{\partial x_k}+\frac{\partial u_k}{\partial x_l}\right)-\frac{2}{3}\mu\frac{\partial u_n}{\partial x_n}\delta_{lk} \tag{2.1c}$$

where ρ is the density, u_i velocity components in Cartesian coordinates, E total energy, p pressure, H total enthalpy ($H = E + (p/\rho)$), and T temperature. The working gas is air approximated by the calorically perfect gas model with the specific heat ratio γ=1.4. The Prandtl number is Pr=0.72. The molecular viscosity μ and thermal conductivity κ are related as $\kappa=c_p\mu/\mathrm{Pr}$ where c_p is specific heat at constant pressure.

Eq.(2.1) is solved with a finite-volume code, and can be written in the delta form as:

$$\frac{V_i}{\Delta t_i}\Delta\mathbf{Q}_i+\mathbf{\Gamma}_i^{-1}\sum_j(\mathbf{F}_{i,j}-\mathbf{Fv}_{i,j})S_{i,j}=0 \tag{2.2}$$

B. Numerical Methods

The computational code employed here is “LS-FLOW:” JAXA’s in-house, unstructured, compressible Navier-Stokes solver for arbitrary polygons/polyhedra. LS-FLOW has many options for spatial reconstruction and temporal evolution. Included in Table 1 are only the methods adopted for the present study. The second order of spatial accuracy is guaranteed (Appendix B presents accuracy study for vortex preservation).

The Euler fluxes and the implicit schemes are summarized in Table 2. Viscous fluxes are computed by using Wang’s second-order method [16]. Formulation of each Euler flux is briefly introduced below, followed by that of time evolution methods (for details, see the original literatures).

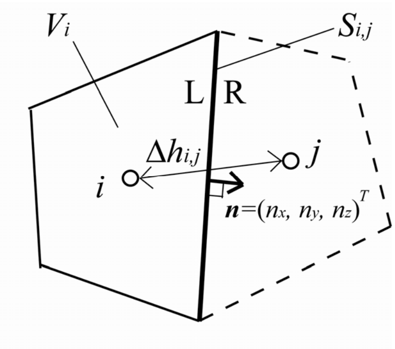


Figure 1: Schematic of cell geometric properties.

Table 1: Numerical methods.

Governing Eqs.		Compressible Euler/Navier-Stokes Eqs.
Spatial Discretization		Cell-centered FVM
Reconstruction	Gradients	Green-Gauss Method[14,15] [††] (without slope limiter)
	Inviscid Term	(see Table 2, “Euler Fluxes”)
	Viscous Term	Wang[16]
Temporal Evolution		(see Table 2, “Implicit Schemes”)

Table 2: Euler fluxes and implicit schemes.

		Baseline	Low-Dissipation/ Preconditioned
Euler Fluxes		Roe[17]	A-Roe[9]
		AUSM+[20]	AUSM$^+$-up[6]
		SHUS[21]	SLAU[8]
Implicit Schemes		LU-SGS[10]	pLU-SGS[4]

[††] The Green-Gauss method is suitable for body-fitted (= structured-type) grids as used here [14,15], whereas not for Cartesian grids having hanging-nodes.

- **Euler Fluxes**

Inviscid numerical fluxes at cell-interfaces $\mathbf{F}_{1/2}$ are calculated by one of the following Euler fluxes.

1) Roe [17] and All-Speed-Roe (A-Roe) [9]: Using the difference of variables *Δ*( ) = ( )$_R$ - ( )$_L$ and the Roe-averaged [17] values (^), the Roe flux is expressed in the following form of Liu and Vinokur [18].

$$\mathbf{F}_{1/2} = \frac{1}{2}\left(\mathbf{F}_L + \mathbf{F}_R - \hat{\mathbf{R}}\left|\hat{\mathbf{\Lambda}}\right|\hat{\mathbf{L}}\mathbf{\Delta Q}\right) \tag{2.3a}$$

$$\hat{\mathbf{R}}\left|\hat{\Lambda}\right|\hat{\mathbf{L}}\mathbf{\Delta Q} = \left|\hat{\lambda}_1\right|\mathbf{\Delta Q} + \delta_1\hat{\mathbf{Q}}^* + \delta_2\mathbf{N} \tag{2.3b}$$

where

$$\left.\begin{aligned}
&\hat{\mathbf{Q}}^* = \begin{pmatrix}1 & u & v & w & H\end{pmatrix}^T \\
&\mathbf{N} = \begin{pmatrix}0 & n_x & n_y & n_z & V_n\end{pmatrix}^T \\
&\mathbf{\Lambda} = diag(\lambda_1, \ \lambda_2, \ \ldots \ \lambda_5) \\
&\delta_1 = \left(\lambda^+ \cdot (\Delta p/\hat{c}) + \lambda^- \cdot \hat{\rho}\Delta V_n\right)/\hat{c} \\
&\delta_2 = \lambda^+ \cdot \hat{\rho}\Delta V_n + \lambda^- \cdot (\Delta p/\hat{c}) \\
&\lambda^+ = -\left|\hat{\lambda}_1\right| + \frac{\left|\hat{\lambda}_2\right| + \left|\hat{\lambda}_3\right|}{2}, \quad \lambda^- = \frac{\left|\hat{\lambda}_2\right| - \left|\hat{\lambda}_3\right|}{2}
\end{aligned}\right\} \tag{2.3c}$$

$$\lambda_{1,4,5} = V_n, \ \lambda_2 = V_n + c, \ \lambda_3 = V_n - c \tag{2.3d}$$

$$V_n = un_x + vn_y + wn_z \tag{2.3e}$$

The mass flux (first row of $\mathbf{F}_{1/2}$), for example, is written as follows:

$$\dot{m}_{1/2} = \frac{1}{2}\left(\rho_L V_{nL} + \rho_R V_{nR} - \left|\hat{V}_n\right|\Delta\rho - \frac{\left|\hat{M}+1\right| + \left|\hat{M}-1\right| - 2\left|\hat{M}\right|}{2}\cdot\frac{\Delta p}{\hat{c}} - \frac{\left|\hat{M}+1\right| - \left|\hat{M}-1\right|}{2}\cdot\hat{\rho}\Delta V_n\right) \tag{2.3f}$$

The Roe's approximate Riemann solver is one of the most widely-used numerical fluxes, but this flux is known to suffer from the carbuncle phenomenon [19] or expansion shock at high speeds, and as shown later, unphysical oscillations at low speeds.

All-Speed-Roe (A-Roe), which was developed recently by Li and Gu [9], modified the Roe flux by introducing the switching function *f*(*M*) for all speeds as follows:

$$\mathbf{F}_{1/2} = \frac{1}{2}\left(f(M)(\mathbf{F}_L + \mathbf{F}_R) + [1 - f(M)]\mathbf{F}_c^{press} - \hat{\mathbf{R}}\left|\hat{\mathbf{\Lambda}}^{A-Roe}\right|\hat{\mathbf{L}}\mathbf{\Delta Q}\right) \tag{2.4a}$$

$$f(M) = \min\left(M\frac{\sqrt{4 + (1 - M^2)^2}}{1 + M^2}, \ 1\right) \tag{2.4b}$$

$$\mathbf{F}_c^{press} = U_c\mathbf{\Phi} + \mathbf{P} \tag{2.4c}$$

$$U_c = (V_{nL} + V_{nR})/2 - c_2/(\rho_* u_*)\cdot(p_R - p_L) \tag{2.4d}$$

$$\mathbf{\Phi} = (\rho, \rho u, \rho v, \rho w, \rho H)^T, \quad \mathbf{P} = (0, pn_x, pn_y, pn_z, 0)^T \tag{2.4e}$$

$$\lambda_{1,4,5}^{A-Roe} = V_n, \ \lambda_2^{A-Roe} = V_n + f(M)c, \ \lambda_3^{A-Roe} = V_n - f(M)c \tag{2.4f}$$

where $\mathbf{F}_c^{\text{press}}$ is a pressure stabilization term with $c_2 = 0.05$ and $\rho_* u_* = \rho_\infty u_\infty$. According to the original paper[9], these values with (*) should be the maximum values in the whole computational domain; however, they are simply set to be freestream values here, since no shock discontinuities are present in this work.

This scheme does not rely on "cutoff Mach number M_{co}," which is typically borrowed from preconditioning matrix $\mathbf{\Gamma}$ (see Eq.(A.2) in Appendix) and included in some other all speed schemes (e.g, preconditioned Roe in [4]), though reference values $\rho_* u_*$ should be specified.

2) AUSM+ [16] and AUSM+-up [6]: AUSM-family schemes [6-8,20,21] are another set of widely-used fluxes featuring simplicity and relative robustness against shock-related anomalies (e.g., carbuncle phenomenon [22]). Among AUSM-family, we first introduce two representative methods, i.e., AUSM+ and its all-speed extension, AUSM$^+$-up.

Formulation of AUSM+ is given as:

$$\mathbf{F}_{1/2} = \frac{\dot{m} + |\dot{m}|}{2}\mathbf{\Psi}^+ + \frac{\dot{m} - |\dot{m}|}{2}\mathbf{\Psi}^- + \tilde{p}\mathbf{N} \tag{2.5a}$$

$$\mathbf{\Psi} = (1, u, v, w, H)^T, \quad \mathbf{N} = (0, n_x, n_y, n_z, 0)^T \tag{2.5b}$$

$$\dot{m}_{1/2} = M_{1/2} c_{1/2} \begin{cases} \rho_L & if\ M_{1/2} > 0 \\ \rho_R & otherwise \end{cases} \tag{2.5c}$$

$$M_{1/2} = f^{+}_{M\,L}\Big|_{\beta=1/8} + f^{-}_{M\,R}\Big|_{\beta=1/8} \tag{2.5d}$$

$$\tilde{p} = f^{+}_{p\,L}\Big|_{\alpha=3/16} p_L + f^{-}_{p\,R}\Big|_{\alpha=3/16} p_R \tag{2.5e}$$

$$f^{\pm}_{M}\Big|_{\beta} = \begin{cases} \dfrac{1}{2}(M \pm |M|), & if\ |M| \geq 1 \\ \pm\dfrac{1}{4}(M \pm 1)^2 \pm \beta(M^2 - 1)^2, & otherwise \end{cases} \tag{2.5f}$$

$$f^{\pm}_{p}\Big|_{\alpha} = \begin{cases} \dfrac{1}{2}(1 \pm sign(M)), & if\ |M| \geq 1 \\ \dfrac{1}{4}(M \pm 1)^2(2 \mp M) \pm \alpha M(M^2 - 1)^2, & otherwise \end{cases} \tag{2.5g}$$

where

$$c_{1/2} = \min(\tilde{c}_L, \tilde{c}_R), \quad \tilde{c} = c^{*2}/\max(c^*, |u|) \tag{2.5h}$$

and

$$c^{*2} = \frac{2(\gamma - 1)}{(\gamma + 1)} H \tag{2.5i}$$

This scheme was extended later for all speeds as AUSM$^+$-up, by introducing additional user-specified parameters:

$$\mathbf{F}_{1/2} = \frac{\dot{m} + |\dot{m}|}{2}\mathbf{\Psi}^+ + \frac{\dot{m} - |\dot{m}|}{2}\mathbf{\Psi}^- + \tilde{p}\mathbf{N} \tag{2.6a}$$

$$\mathbf{\Psi} = (1, u, v, w, H)^T, \quad \mathbf{N} = (0, n_x, n_y, n_z, 0)^T \tag{2.6b}$$

$$\dot{m}_{1/2} = M_{1/2} c_{1/2} \begin{cases} \rho_L & if\ M_{1/2} > 0 \\ \rho_R & otherwise \end{cases} \tag{2.6c}$$

$$M_{1/2} = f^{+}_{M\,L}\Big|_{\beta=1/8} + f^{-}_{M\,R}\Big|_{\beta=1/8} + M_p \tag{2.6d}$$

$$\widetilde{p} = f_{p\,L}^{+}\Big|_{\alpha} p_L + f_{p\,R}^{-}\Big|_{\alpha} p_R + p_u \tag{2.6e}$$

where

$$c_{1/2} = \min\left(\widetilde{c}_L, \widetilde{c}_R\right), \quad \widetilde{c}_L = c^{*2}/\max\left(c^*, u_L\right), \quad \widetilde{c}_R = c^{*2}/\max\left(c^*, -u_R\right) \tag{2.6f}$$

and

$$M_p = -\frac{K_p}{f_a}\max\left(1-\sigma\overline{M}^2, 0\right)\frac{p_R - p_L}{\rho_{1/2}c_{1/2}^2}, \quad \rho_{1/2} = \frac{\rho_L + \rho_R}{2} \tag{2.6g}$$

$$p_u = -K_u f_{p\,L}^{+} f_{p\,R}^{-}\left(\rho_L + \rho_R\right)\left(f_a c_{1/2}\right)\left(u_R - u_L\right) \tag{2.6h}$$

with

$$K_p = 0.25, \quad K_u = 0.75, \quad \sigma = 1.0 \tag{2.6i}$$

$$\overline{M}^2 = \frac{V_{nL}{}^2 + V_{nR}{}^2}{2c_{1/2}^2}, \quad \alpha = \frac{3}{16}\left(-4 + 5f_a^2\right) \tag{2.6j}$$

$$f_a\left(M_o\right) = M_o\left(2 - M_o\right) \tag{2.6k}$$

$$M_o{}^2 = \min\left(1, \max\left(\overline{M}^2, M_\infty^2\right)\right) \tag{2.6l}$$

This scheme also excludes “cutoff Mach number M_{co},” though freestream Mach number M_∞ is required.

3) SHUS [21] and SLAU [8]: SHUS (Simple High-resolution Upwind Scheme) is one of AUSM-family schemes, which replaced the mass flux of AUSM+ with that of Roe (Eq. 2.3f) with the use of arithmetic averaged values rather than Roe-averaged ones. This scheme achieved accuracy of Roe flux while keeping the robustness of AUSM+ against shock anomalies.

$$\mathbf{F}_{1/2} = \frac{\dot{m} + |\dot{m}|}{2}\mathbf{\Psi}^+ + \frac{\dot{m} - |\dot{m}|}{2}\mathbf{\Psi}^- + \widetilde{p}\mathbf{N} \tag{2.7a}$$

$$\mathbf{\Psi} = \left(1, u, v, w, H\right)^T, \quad \mathbf{N} = \left(0, n_x, n_y, n_z, 0\right)^T \tag{2.7b}$$

The mass flux function of SHUS is given as:

$$\dot{m} = \frac{1}{2}\left\{(\rho V_n)_L + (\rho V_n)_R - \left|\frac{V_{nL} + V_{nR}}{2}\right|\Delta\rho - \frac{\left|\overline{M}+1\right| - \left|\overline{M}-1\right|}{2}\overline{\rho}\Delta V_n - \frac{\left|\overline{M}+1\right| + \left|\overline{M}-1\right| - 2\left|\overline{M}\right|}{2\overline{c}}\Delta p\right\} \tag{2.7c}$$

where

$$\Delta(\ ) = (\ )_R - (\ )_L \tag{2.7d}$$

$$\overline{(\ )} = \frac{(\ )_L + (\ )_R}{2} \tag{2.7e}$$

The pressure flux is: (2.7f)

$$\widetilde{p} = f_{p\,L}^{+}\Big|_{\alpha=0} p_L + f_{p\,R}^{-}\Big|_{\alpha=0} p_R \tag{2.7g}$$

SHUS was further developed to give more reliable solutions both at low and high speeds. The latest version is named SLAU (Simple Low-dissipation AUSM):

$$\mathbf{F}_{1/2} = \frac{\dot{m} + |\dot{m}|}{2}\mathbf{\Psi}^+ + \frac{\dot{m} - |\dot{m}|}{2}\mathbf{\Psi}^- + \widetilde{p}\mathbf{N} \tag{2.8a}$$

$$\mathbf{\Psi} = \left(1, u, v, w, H\right)^T, \quad \mathbf{N} = \left(0, n_x, n_y, n_z, 0\right)^T \tag{2.8b}$$

The mass flux function of SLAU is

$$\dot{m} = \frac{1}{2}\left\{\rho_L\left(V_{nL} + \left|\bar{V}_n\right|^+\right) + \rho_L\left(V_{nL} - \left|\bar{V}_n\right|^-\right) - \frac{\chi}{\bar{c}}\Delta p\right\} \tag{2.8c}$$

$$\left|\bar{V}_n\right| = \frac{\rho_L\left|V_{nL}\right| + \rho_R\left|V_{nR}\right|}{\rho_L + \rho_R} \tag{2.8d}$$

$$|\bar{V}_n|^+ = (1-g)|\bar{V}_n| + g|V_{nL}|, \quad |\bar{V}_n|^- = (1-g)|\bar{V}_n| + g|V_{nR}| \tag{2.8e}$$

$$g = -\max\left[\min\left(M_L, 0\right), -1\right] \cdot \min\left[\max\left(M_R, 0\right), 1\right] \quad \in [0, 1] \tag{2.8f}$$

and the pressure flux is

$$\tilde{p} = \frac{p_L + p_R}{2} + \frac{f_{pL}^+\big|_{\alpha=0} - f_{pR}^-\big|_{\alpha=0}}{2}\left(p_L - p_R\right) + \left(1-\chi\right)\left(f_{pL}^+\big|_{\alpha=0} + f_{pR}^-\big|_{\alpha=0} - 1\right)\frac{p_L + p_R}{2} \tag{2.8g}$$

$$\chi = (1-\hat{M})^2 \tag{2.8h}$$

$$\hat{M} = \min\left(1.0, \frac{1}{\bar{c}}\sqrt{\frac{u_L{}^2 + v_L{}^2 + w_L{}^2 + u_R{}^2 + v_R{}^2 + w_R{}^2}{2}}\right) \tag{2.8i}$$

$$M = \frac{V_n}{\bar{c}} = \frac{un_x + vn_y + wn_z}{\bar{c}} \tag{2.8j}$$

SLAU needs no cutoff Mach number M_{co} or freestream Mach number M_∞. To the best of the authors' knowledge, this flux is the only method among all speed schemes which is totally free from restrictions of specifying reference values. This property is desirable for computations of flows involving no uniform flow, such as turbopump internal flows [1].

- **Time Evolution Methods**

Inviscid numerical fluxes at cell-interfaces $\mathbf{F}_{1/2}$ are calculated by one of the following Euler fluxes.

1) LU-SGS and pLU-SGS (preconditioned LU-SGS) Implicit Schemes: Time integration is conducted by using LU-SGS implicit method or its preconditioned version, preconditioned LU-SGS[4], which is referred to as "pLU-SGS" for brevity here. Its formulation starts from Eq. (2.2), expressed with time step index *n* included

$$\frac{V_i}{\Delta t_i}\Delta\mathbf{Q}_i^n + \boldsymbol{\Gamma}_i^{-1}\sum_j(\Delta\mathbf{F}_{i,j}^n - \Delta\mathbf{Fv}_{i,j}^n)S_{i,j} = \boldsymbol{\Gamma}_i^{-1}\cdot\mathbf{Res}_i^n \tag{2.9a}$$

where

$$\Delta(\ )^n = (\ )^{n+1} - (\ )^n \tag{2.9b}$$

and $\mathbf{Res}_i$ is the right-hand side residual,

$$\mathbf{Res}_i^n = \sum_j(\mathbf{F}_{i,j}^n - \mathbf{Fv}_{i,j}^n)S_{i,j} \tag{2.9c}$$

Again, in the case without preconditioning, $\boldsymbol{\Gamma}^{-1}$ is simply dropped.

Then, Eq. (2.9) is rewritten in the form of Gauss-Seidel (GS) iterative method by decomposition of new (updated) and old (non-updated) values

$$\begin{aligned}&\Delta\mathbf{Q}_i^{n+1}\\ &= \mathbf{D}_i^{n-1}\cdot\left[\boldsymbol{\Gamma}_i^{-1}\sum_{j\in Lower}S_{i,j}\mathbf{A}_{j,i}^{+\ new/old}\Delta\mathbf{Q}_j^{new/old} + \boldsymbol{\Gamma}_i^{-1}\sum_{j\in Upper}S_{i,j}\mathbf{A}_{j,i}^{+\ new/old}\Delta\mathbf{Q}_j^{new/old} - \boldsymbol{\Gamma}_i^{-1}\cdot\mathbf{Res}_i^n\right]\end{aligned} \tag{2.10a}$$

where

$$\mathbf{A}_{i,j} = \frac{\partial \mathbf{F}_{i,j}}{\partial \mathbf{Q}_i} - \frac{\partial \mathbf{Fv}_{i,j}}{\partial \mathbf{Q}_i} \tag{2.10b}$$

is flux Jacobian from cell i to cell j through the cell-interface $S_{i,j}$. $\mathbf{A}^+$ has only the positive components of the eigenvectors. The diagonal matrix $\mathbf{D}_i$ is given as

$$\mathbf{D}_i = \frac{V_i}{\Delta t_i}\mathbf{I} + \mathbf{\Gamma}_i^{-1}\sum_j S_{i,j}\mathbf{A}_{i,j}^+ \tag{2.10c}$$

Specifically, in LU-SGS (Lower Upper Symmetric Gauss-Seidel), Eq. (2.10a) is further rewritten as

$$\Delta\mathbf{Q}_i^* = \mathbf{D}_i^{-1} \cdot \left[\mathbf{\Gamma}_i^{-1} \sum_{j \in Lower} S_{i,j}\mathbf{A}_{j,i}^{+\,*}\Delta\mathbf{Q}_j^* - \mathbf{\Gamma}_i^{-1} \cdot \mathbf{Res}_i^n \right] \tag{2.11a}$$

$$\begin{aligned}\Delta\mathbf{Q}_i^{n+1} &= \mathbf{D}_i^{-1} \cdot \left[\mathbf{\Gamma}_i^{-1} \sum_{j \in Lower} S_{i,j}\mathbf{A}_{j,i}^{+\,*}\Delta\mathbf{Q}_j^* + \mathbf{\Gamma}_i^{-1} \sum_{j \in Upper} S_{i,j}\mathbf{A}_{j,i}^{+\,n+1}\Delta\mathbf{Q}_j^{n+1} - \mathbf{\Gamma}_i^{-1} \cdot \mathbf{Res}_i^n \right] \\ &= \Delta\mathbf{Q}_i^* + \mathbf{D}_i^{-1} \cdot \mathbf{\Gamma}_i^{-1} \sum_{j \in Upper} S_{i,j}\mathbf{A}_{j,i}^{+\,n+1}\Delta\mathbf{Q}_j^{n+1}\end{aligned} \tag{2.11b}$$

Then, $\mathbf{A}^+$ is approximated as the following as in Jameson and Turkel's LU-SGS [10] (this version is commonly referred to as "LU-SGS").

$$\mathbf{A}^+{}_{i,j} \approx \frac{\mathbf{A}_{i,j} + \sigma_{i,j}\mathbf{I}}{2} \tag{2.12a}$$

where

$$\begin{aligned}\sigma_{i,j} &= \sigma_{i,j}\left(\mathbf{A}_{i,j}\right) \\ &= \left|V_{n\,i,j}\right| + c_i + \frac{2\left(\mu_i + \mu_j\right)}{\left(\rho_i + \rho_j\right) \cdot \Delta h_{i,j}}\end{aligned} \tag{2.12b}$$

where $\Delta h_{i,j}$ is distance between cell-centers of i and j.

In pLU-SGS, the spectral radius $\sigma_{i,j}$ is scaled as $\sigma'_{i,j}$, thus,

$$\mathbf{\Gamma}_i^{-1}\mathbf{A}^+{}_{i,j} \approx \frac{\mathbf{\Gamma}_i^{-1}\mathbf{A}_{i,j} + \sigma'_{i,j}\mathbf{I}}{2} \tag{2.13a}$$

where

$$\begin{aligned}\sigma'_{i,j} &= \sigma'\left(\mathbf{\Gamma}_i^{-1}\mathbf{A}_{i,j}\right) \\ &= \frac{1}{2}\left\{(1+\varepsilon)\left|V_{n\,i,j}\right| + \sqrt{(\varepsilon - 1)^2 V_{n\,i,j}{}^2 + 4\varepsilon c_i^2}\right\} + \frac{2\left(\mu_i + \mu_j\right)}{\left(\rho_i + \rho_j\right) \cdot \Delta h_{i,j}}\end{aligned} \tag{2.13b}$$

The preconditioning coefficient ε, which should be of the order of $M_\infty{}^2$, appears in the preconditioning matrix as given by $\varepsilon = \min(1, \max(KM^2, M_{\text{co}}{}^2))$ (Eq.(A.2) in Appendix). Variables in this equation can be arbitrary chosen, and here, K is taken as unity and $M_{\text{co}} = M_\infty$, leading to

$$\varepsilon = \min(1, \max(M^2, M_\infty{}^2)) \tag{2.13c}$$

With the above approximation, the diagonal block $\mathbf{D}_i$ (in Eq. (2.10c)) is transformed into merely a scalar.

$$\begin{aligned}\mathbf{D}_i &= \frac{V_i}{\Delta t_i}\mathbf{I} + \sum_j S_{i,j}\boldsymbol{\Gamma}_i^{-1}\mathbf{A}^{+}_{i,j} \\ &= \left(\frac{V_i}{\Delta t_i} + \sum_j S_{i,j}\frac{\sigma'_{i,j}}{2}\right)\mathbf{I}\end{aligned} \tag{2.14a}$$

$$\because \ \sum_j S_{i,j}\mathbf{A}_{i,j} = 0 \tag{2.14b}$$

Substituting Eqs. (2.10b), (2.13a), and (2.14a), Eqs. (2.11a) and (2.11b) becomes

$$\begin{aligned}\Delta\mathbf{Q}^*_i &= \left(\frac{V_i}{\Delta t_i} + \sum_j S_{i,j}\frac{\sigma'_{i,j}}{2}\right)^{-1}\cdot\left[\boldsymbol{\Gamma}_i^{-1}\sum_{j\in Lower} S_{i,j}\mathbf{A}^{+}_{j,i}\Delta\mathbf{Q}^*_j - \boldsymbol{\Gamma}_i^{-1}\cdot\mathbf{Res}^n_i\right] \\ &= \left(\frac{V_i}{\Delta t_i} + \sum_j S_{i,j}\frac{\sigma'_{i,j}}{2}\right)^{-1}\cdot\left[\sum_{j\in Lower} S_{i,j}\frac{\boldsymbol{\Gamma}_i^{-1}\mathbf{A}_{j,i} + \sigma'_{j,i}\mathbf{I}}{2}\Delta\mathbf{Q}^*_j - \boldsymbol{\Gamma}_i^{-1}\cdot\mathbf{Res}^n_i\right] \\ &= \left(\frac{V_i}{\Delta t_i} + \sum_j S_{i,j}\frac{\sigma'_{i,j}}{2}\right)^{-1}\cdot\left[\frac{1}{2}\sum_{j\in Lower} S_{i,j}\left(\boldsymbol{\Gamma}_i^{-1}\left(\Delta\mathbf{F}^*_{j,i} - \Delta\mathbf{Fv}^*_{j,i}\right) + \sigma'_{j,i}\Delta\mathbf{Q}^*_j\right) - \boldsymbol{\Gamma}_i^{-1}\cdot\mathbf{Res}^n_i\right]\end{aligned} \tag{2.15a}$$

$$\begin{aligned}\Delta\mathbf{Q}^{n+1}_i &= \Delta\mathbf{Q}^*_i + \left(\frac{V_i}{\Delta t_i} + \sum_j S_{i,j}\frac{\sigma'_{i,j}}{2}\right)^{-1}\cdot\boldsymbol{\Gamma}_i^{-1}\sum_{j\in Upper} S_{i,j}\mathbf{A}^{+}_{j,i}{}^{n+1}\Delta\mathbf{Q}^{n+1}_j \\ &= \Delta\mathbf{Q}^*_i + \left(\frac{V_i}{\Delta t_i} + \sum_j S_{i,j}\frac{\sigma'_{i,j}}{2}\right)^{-1}\cdot\frac{1}{2}\sum_{j\in Upper} S_{i,j}\left(\boldsymbol{\Gamma}_i^{-1}\left(\Delta\mathbf{F}^{n+1}_{j,i} - \Delta\mathbf{Fv}^{n+1}_{j,i}\right) + \sigma'_{j,i}\Delta\mathbf{Q}^*_j\right)\end{aligned} \tag{2.15b}$$

Note that the computational cost for the implementation of $\boldsymbol{\Gamma}^{-1}\Delta\mathbf{F}$ is trivial according to Turkel [5], by using the following form.

$$\boldsymbol{\Gamma}^{-1}d\mathbf{Q} = d\mathbf{Q} - \frac{(1-\varepsilon)dp}{c^2}\begin{pmatrix}1 & u & v & w & H\end{pmatrix}^T \tag{2.16a}$$

$$dp = (\gamma - 1)\left(\frac{u^2+v^2+w^2}{2}d\rho - ud(\rho u) - vd(\rho v) - wd(\rho w) + dE\right) \tag{2.16b}$$

2) Time step: Time step Δt_i is given by the following formula.

$$\Delta t_i = CFL * V_i \Big/ \left[\sum_j \sigma_{i,j}S_{i,j}\right] \tag{2.17}$$

where CFL is Courant number, and the spectral radius σ can be replaced by σ' for preconditioned systems.

The use of Eq. (2.17) is called local time stepping, whereas the global time stepping takes the form of

$$\Delta t = CFL / \max_{i,j}\left(\sigma_{i,j}/\Delta h_{i,j}\right) \tag{2.18}$$

3) Sub-iteration procedure: The sub-iteration (sometimes called Newton-iteration) is used to enhance the convergence rate outside the LU-SGS loop. Eq. (2.9a),

$$\frac{V_i}{\Delta t_i}\Delta\mathbf{Q}^n_i + \boldsymbol{\Gamma}_i^{-1}\sum_j(\Delta\mathbf{F}^n_{i,j} - \Delta\mathbf{Fv}^n_{i,j})S_{i,j} = \boldsymbol{\Gamma}_i^{-1}\cdot\mathbf{Res}^n_i \tag{2.9a}$$

with $\Delta\mathbf{Q}^n \rightarrow \Delta\mathbf{Q}^m$ and applying three-point backward difference (subscripts *i* and *j* are omitted for clarity), leads to

$$\frac{3\mathbf{Q}^{m+1}-4\mathbf{Q}^{n}+\mathbf{Q}^{n-1}}{2}=\frac{3\Delta\mathbf{Q}^{m}+3\mathbf{Q}^{m}-4\mathbf{Q}^{n}+\mathbf{Q}^{n-1}}{2}=-\frac{\Delta t}{V}\mathbf{\Gamma}^{-1}\cdot\left\{\mathbf{Res}^{m}-(\Delta\mathbf{F}^{n}-\Delta\mathbf{Fv}^{n})S\right\} \quad (2.19a)$$

Thus,

$$\Delta\mathbf{Q}^{m}=-\frac{2}{3}\Delta t\left[\frac{3\mathbf{Q}^{m}-4\mathbf{Q}^{n}+\mathbf{Q}^{n-1}}{2}+\frac{\Delta t}{V}\mathbf{\Gamma}^{-1}\cdot\left\{\mathbf{Res}^{m}-(\Delta\mathbf{F}^{n}-\Delta\mathbf{Fv}^{n})S\right\}\right] \quad (2.19b)$$

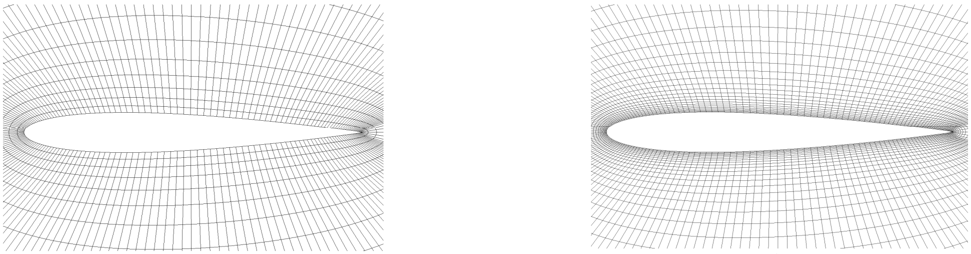
where m is the number of sub-iterations, and when m reaches the specified maximum iteration number or the $\Delta\mathbf{Q}^{m}$ reduced to the threshold value, the sub-iteration process is terminated as $\Delta\mathbf{Q}^{m}\rightarrow\Delta\mathbf{Q}^{n}$. Note that this procedure achieves second-order temporal accuracy if Δt is frozen throughout the computation. In addition, with preconditioning matrix $\mathbf{\Gamma}$, dual time stepping is usually adopted for unsteady calculations [4]. However, we did not take this strategy and used sub-iterations only to accelerate and stabilize computations of steady flows.[‡‡]

In the subsequent sections, Courant number is chosen as CFL = 20 in consideration of both stability and efficiency, and no sub-iterations (= one sub-iteration) or three sub-iterations are employed, if not mentioned otherwise. The global time stepping technique is usually used (unless stated otherwise). Based on the flow conditions explained below (in Table 3), no slope limiters or turbulence models are used.

C. Flow Conditions

Computations are conducted for a subsonic or a low-Mach-number flow over NACA0012 airfoil, under the conditions given in Table 3. The airfoil has no angle-of-attack throughout the present study. The following two grids are generated (Fig. 2):

- Two-dimensional, O-type, structured grids.
- 201 points in the circumferential direction, and 31 points (inviscid) or 51 points (viscous) in the radial (wall-normal) direction, respectively.
- The minimum spacing near the wall for viscous cases is δ= 1.0e-3, based on the chord length of 1. This spacing achieves sufficient resolution for boundary-layers considered here.
- Far field boundary is 50 times chord length away from the wall.


(a)

(b)

Figure 2: Computational grids for a) inviscid (201 x 31 points) and b) viscous (201 x 51 points) simulations.

Table 3: Test cases and conditions.

Cases	Conditions	Comments
1) Viscous	M_∞=0.5, Re_∞=5,000[23,24]	Moderate Speed (Validation)
2) Inviscid	M_∞=0.001 - 0.1	Low Speeds
3) Viscous	M_∞=0.01, Re_∞=2,000[3,11]	Low Speed

[‡‡] We found that three inner-iterations helped to accelerate convergence and/or to stabilize computations even for steady cases by significant reduction (two orders drop) of RHS residual achieved at the third inner loop.

III. Results and Discussions

The results are summarized in Tables 4-5 in which the following notations are used:

- **S** (Stable): The L2-norm of density residual dropped at least four orders with physically correct solution.
- **U** (Unphysical): The solution reached to unphysical one with poor quality, and/or the residual remained significant (=oscillatory).
- **F** (Failure): The calculation diverged.

The result of each case will be discussed in the subsequent sub-sections.

Table 4: Computed results of *no sub-iteration* cases (CFL=20), **S** (Successful), **U** (Unphysical or oscillatory), and **F** (Failure, diverged).

Sub-iterations: 1		Case 1	Case 2			Case 3
Implicit Schemes	Euler Fluxes	M_∞=0.5	A) M_∞=0.1	B) M_∞=0.01	C) M_∞=0.001	M_∞=0.01
		Re_∞=5,000	Re_∞= ∞ (Inviscid)			Re_∞=2,000
LU-SGS	Roe	**S**	**U**			**U**
	AUSM+					
	SHUS					
	A-Roe	**U**	**U**	**F**	**F**	**F**
	AUSM$^+$-up	**S**	**F**	**F**	**F**	**F**
	SLAU	**S**	**U**	**U**	**U**	**U**
pLU-SGS	Roe	**S**	**F**			**F**
	AUSM+	**F**				
	SHUS	**S**				
	A-Roe	**U**	**F**	**F** (**F**: CFL=200)	**F**	**S** (**S**: CFL=200)
	AUSM$^+$-up	**F**	**S**	**S** (**S**: CFL=200)	**S**	**S** (**S**: CFL=200)
	SLAU	**S**	**S**	**S** (**F**: CFL=200)	**S**	**S** (**F**: CFL=200)

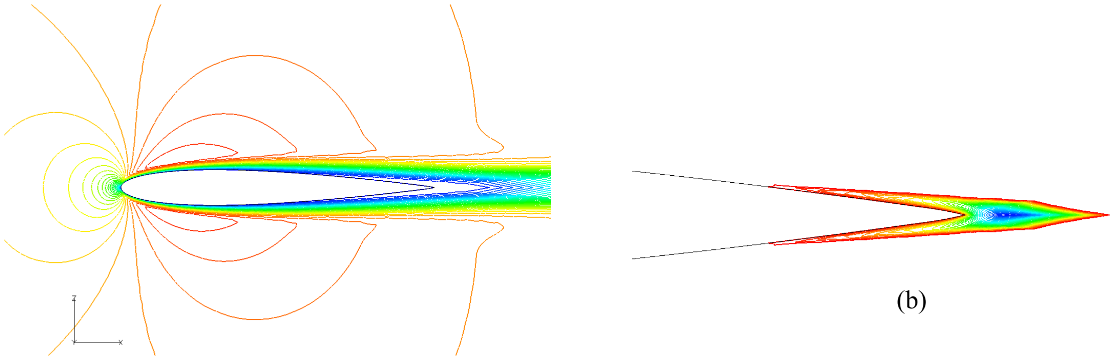
Table 5: Computed results of *three sub-iteration* cases (CFL=20), **S** (Successful), **U** (Unphysical or oscillatory), and **F** (Failure, diverged).

Sub-iterations: 3		Case 1	Case 2			Case 3
Implicit Schemes	Euler Fluxes	M_∞=0.5	A) M_∞=0.1	B) M_∞=0.01	C) M_∞=0.001	M_∞=0.01
		Re_∞=5,000	Re_∞= ∞ (Inviscid)			Re_∞=2,000
LU-SGS	Roe	**S**	**U**			**U**
	AUSM+					
	SHUS					
	A-Roe	**S**	**S**	**F**	**F**	**F**
	AUSM$^+$-up	**S**	**F**	**F**	**F**	**F**
	SLAU	**S**	**S**	**S**	**S**	**S**
pLU-SGS	Roe	**S**	**F**			**F**
	AUSM+	**F**				
	SHUS	**U**				
	A-Roe	**S**	**S**	**S** (**F**: CFL=200)	**S**	**S** (**S**: CFL=200)
	AUSM$^+$-up	**F**	**F**	**S** (**S**: CFL=200)	**S**	**S** (**S**: CFL=200)
	SLAU	**S**	**S**	**S** (**F**: CFL=200)	**S**	**S** (**F**: CFL=200)

A. Case 1: M_∞=0.5 Viscous (Laminar) Flow for Code Validation

This test case has been widely used as a benchmark [23,24]. The computations were conducted for 10,000 timesteps. Typical computed flow field is displayed in Fig. 3. Figure 4 shows histories of drag coefficient C_D and L2-norm of density residual for the successful cases.

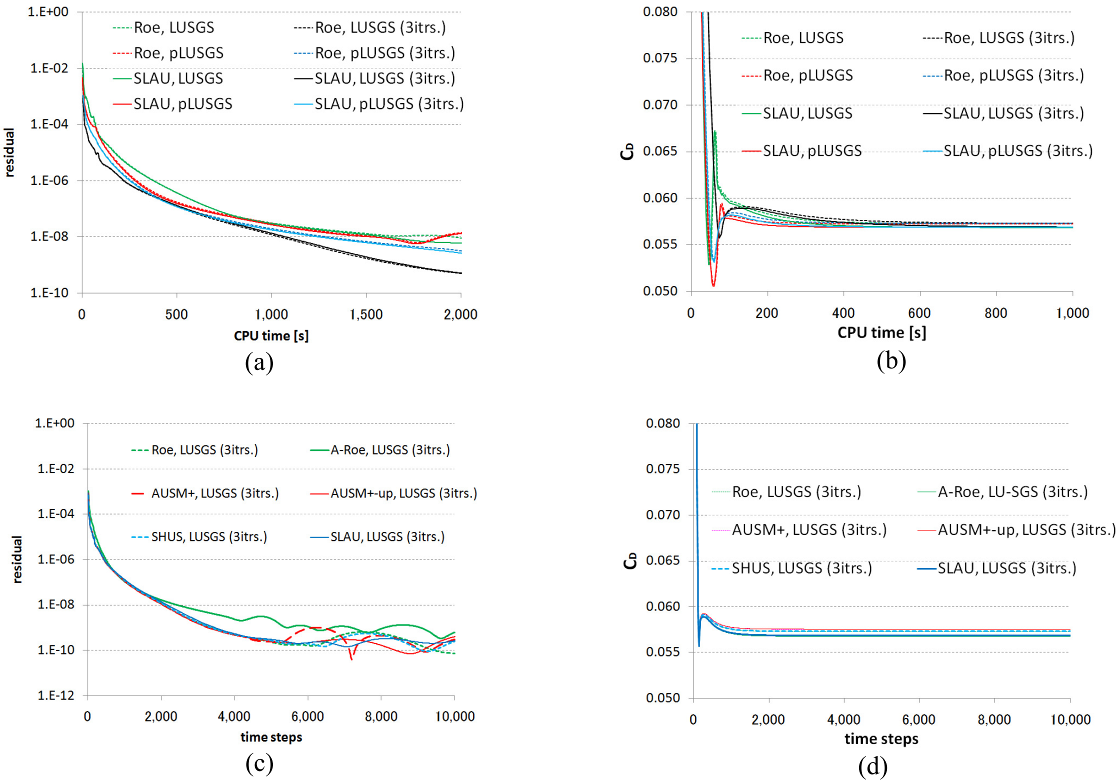
For all the successful cases, the computed flows were almost identical to each other, with slightly different separation points [23,24] near the trailing edge. These locations are in good agreement with reference separation points of 80%-89% chord length, which validates our code.



(b)

(a)

Figure 3: Computed flow field by LU-SGS/SLAU, no sub-iterations, Case 1 (M_∞=0.5, viscous): a) Iso-Mach-contours ($0 < M < 0.59$), b) u-velocity contours; blow-up view of separation region near the trailing-edge ($-0.01 < u < 0$)



(a)

(b)

(c)

(d)

Figure 4: Residual and drag coefficient histories, Case 1 (M_∞=0.5, viscous): a) Residual vs. CPU time, b) C_D vs. CPU time, c) Residual vs. time steps, and d) C_D vs. time steps.

From Tables 4 and 5, and Figs. 3 and 4, the following features are noteworthy:

- Coupled with LU-SGS, all the fluxes (except for A-Roe without sub-iterations) yielded physically correct solutions.
- The LU-SGS/A-Roe combination (without sub-iterations) exhibited an oscillatory solution with the separation point within the reference range. This oscillation was removed by employing three sub-iterations.
- Sub-iterations worked to stabilize the solutions, not to accelerate the convergence (Fig. 4a).

- In this test case, the convergence rate was not practically improved by preconditioning of LU-SGS, although histories of the drag coefficient and residual are slightly affected (Fig. 4a,b). Even worse, calculations diverged in some cases (see Table 4) unless sub-iterations were introduced. This would be because i) some combinations, such as pLU-SGS/AUSM+, resulted in an insufficient amount of dissipation production (explained later), or ii) the scaling function of AUSM$^+$-up did not work well in conjunction with pLU-SGS under the current flow conditions.
- Effect of Euler fluxes seemed to be minor (Fig. 4c,d), compared with the above mentioned factors.

B. Case 2: Low Speed (M_∞=0.1, 0.01, and 0.001), Inviscid Flow

In this section, inviscid computations were carried out for 2,000 timesteps with the freestream Mach number as a parameter: M_∞=0.1 (Case 2A), 0.01 (2B), and 0.001 (2C). Solutions and convergence rates are compared for different methods. Figure 5 shows the typical computed flowfields by LU-SGS/Roe, LU-SGS/SLAU and pLU-SGS/SLAU. In Fig. 6, drag coefficient histories are shown for the three sub-iteration cases. Under the current flow conditions, the computed drag is regarded as an indicator of numerical error. For example, in LU-SGS/Roe calculation, the drag coefficient history reached a plateau at a significant value (Fig. 6b) with an apparently unphysical solution shown in Fig. 5a, even though the corresponding density residual showed five orders of reduction (Fig. 7).

From those figures, the following general remarks are confirmed:

a) If no-preconditioned system of equations (Eq.(2.1) without Γ) is solved, such as LU-SGS/Roe, calculations do not diverge, but reach unphysical solutions due to excessive numerical dissipation of the method (Fig. 5a) [3,8].

b) If only preconditioning A) (time-derivative preconditioning) is used, such as pLU-SGS/Roe, calculations diverge (usually within a few time steps), because the dissipation in the numerical flux is not scaled properly [4,8,25].

c) If only preconditioning B) (numerical flux preconditioning) is used, such as LU-SGS/SLAU, calculations are sometimes unstable and/or oscillatory (Fig. 5b) [3,8]. These oscillations can be cured by introduction of sub-iterations, but this of course increases the computational cost.

d) If both preconditioning A) and B) are used, such as pLU-SGS/SLAU, physically correct solutions are obtained in most cases, with clearly improved convergence (Fig. 5c).

These remarks are summarized in Table 6.

According to Tables 4 and 5, performances of most of the methods presented here were independent on the Mach number. However, the drag coefficients for non-preconditioned cases increased with decreasing Mach number; Meanwhile, those values for the preconditioned cases stayed constant (Fig. 6). The drag coefficient, the error indicator, showed the least value for pLU-SGS/A-Roe of 0.0019, followed by pLU-SGS/SLAU (0.0037), and pLU-SGS/AUSM$^+$-up (0.0049), in the three sub-iteration cases (Table 7). Therefore, from the present standard, *pLUSGS/A-Roe produced the most accurate solution,* if it successfully worked.

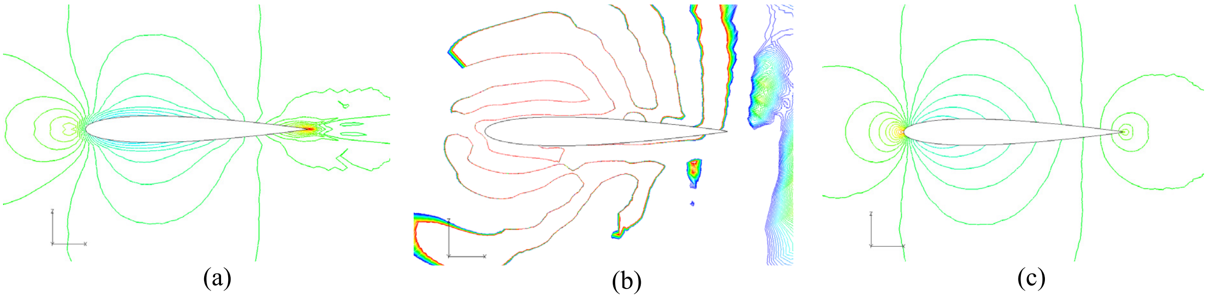
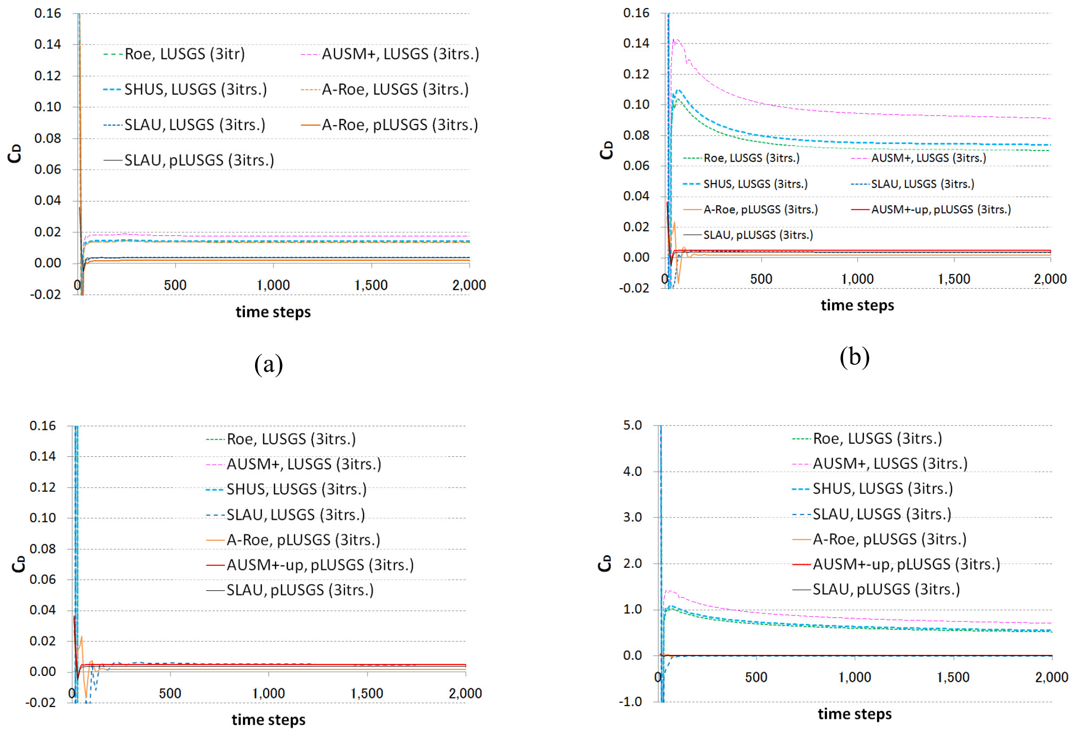


Figure 5: Computed flow fields (-1 < C_p < 1), no sub-iterations, Case 2B (M_∞=0.01, inviscid): a) LU-SGS/Roe (Unphysical), b) LU-SGS/SLAU (Oscillatory), and c) pLU-SGS/SLAU (Stable).



(a)

(b)

(c)

(d)

Figure 6: Drag coefficient histories, Case 2 (inviscid), 3 sub-iterations: a) Case 2A (M_∞=0.1), b) Case 2B (M_∞=0.01), c) Case 2C (M_∞=0.001), and d) Case 2C (M_∞=0.001, wider scale for vertical axis)

Table 6: Summary of computed results (expect for a few exceptions), **S** (Successful), **U** (Unphysical or oscillatory), and **F** (Failure, diverged).

		Euler Fluxes	
		Baseline	Low-Dissipation
Implicit Schemes	LU-SGS	**U**	**S** (slow convergence), **U** or **F**
	pLU-SGS	**F**	**S** (fast convergence)

Table 7: Calculated drag coefficient in inviscid flow (= numerical error) over NACA0012 airfoil.

		No-sub-iterations	Three-sub-iterations
pLU-SGS	A-Roe	(Diverged)	0.0019
	AUSM$^+$-up	0.0049	0.0049
	SLAU	0.0037	0.0037

In addition, it is confirmed that a pLU-SGS/Low-Dissipation-Flux combination can handle even M_∞=0.001 flow. Specifically, with three sub-iterations, pLU-SGS/SLAU produced more successful results than other methods in a range of M_∞= 0.001 – 0.5.

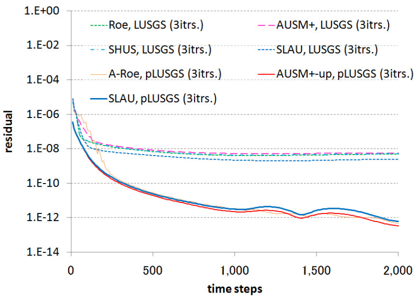
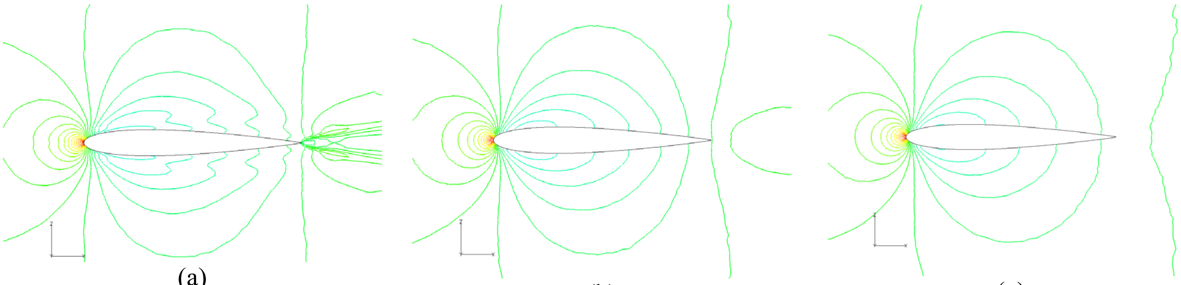


Figure 7: Residual histories, Case 2B (M_∞=0.01, inviscid), 3 sub-iterations.

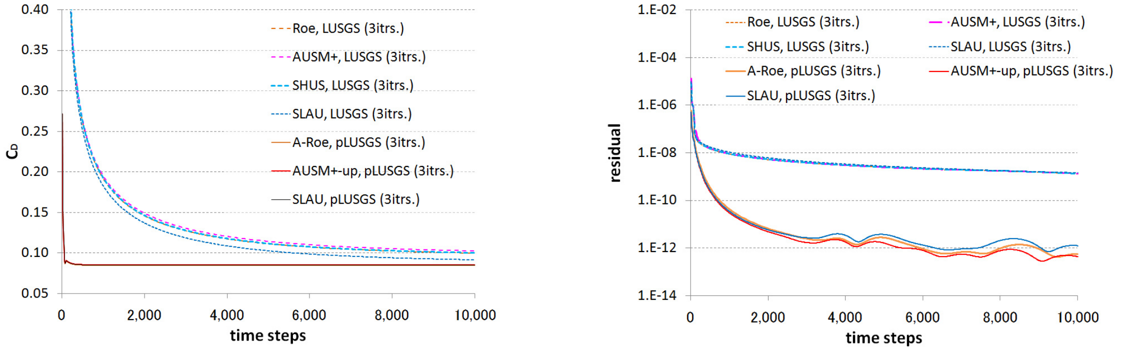
C. Case 3: Low Speed (M_∞=0.01), Viscous Flow

This test case has also been used to investigate the effects of preconditioning [3,11]. Here, however, we focus on the viscous effects. Again, typical computed flow fields and drag/residual histories are shown in Figs. 8 and 9, respectively. As can be seen from these figures and Tables 4 and 5, these computations behaved in a broadly similar manner to their inviscid counterparts (Case 2B; Figs. 5, 6b, and 7), for both aspects of robustness and efficiency. In other words, viscous effects played a minor role in the present cases withfew exceptions. For example, A-Roe flux (without sub-iterations) yielded a satisfactory solution only in the viscous case, probably because its pressure stabilization term (Eqs. 2.4c-2.4e) in combination with the viscous source term (**Fv** in Eq. 2.1b) had a favorable contribution to the solution.



(a) (b) (c)

Figure 8: Computed flow fields (-1 < C_p < 1), 3 sub-iterations, Case 3 (M_∞=0.01, viscous): a) LU-SGS/Roe (Unphysical), b) LU-SGS/SLAU (Stable, slow convergence), and c) pLU-SGS/SLAU (Stable, fast convergence).



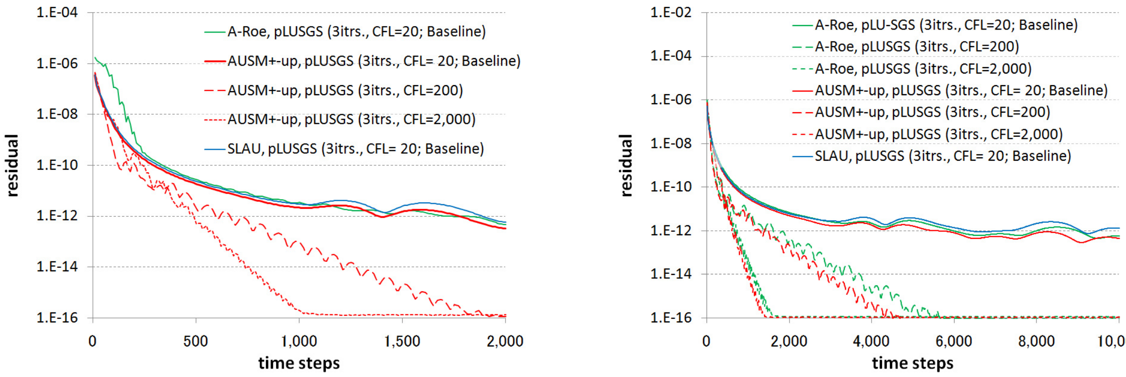
(a) (b)

Figure 9: Drag coefficient and residual histories, Case 3 (M_∞=0.01, viscous), 3 sub-iterations: a) C_D, b) Residual.

D. Effect of Courant Numbers

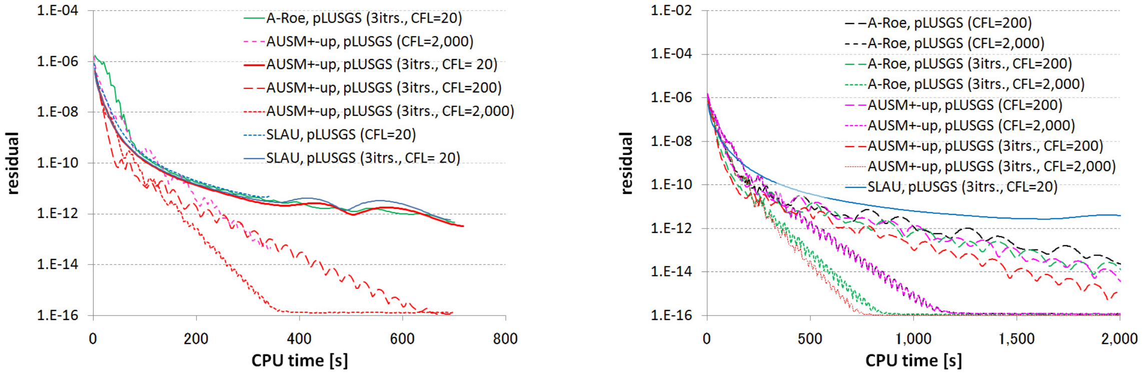
We compared convergence rates of pLU-SGS/Low-dissipation-flux combinations with different Courant numbers ranging from 2 to 2,000 for M_∞=0.01, both in inviscid (Case 2B) and viscous (Case 3) cases. According to the results included also in Tables 4 and 5, the larger the Courant number is, the more the computation tends to be oscillatory or unstable. Figure 10 shows residual histories for the cases with sub-iterations (diverged cases are excluded, e.g., pLU-SGS/SLAU with CFL=200). Judging from this figure, pLU-SGS/AUSM$^+$-up with CFL=2,000 gave the fastest convergence (to machine zero) with a satisfactory solution both in the inviscid and the viscous cases, whereas this combination with CFL=200 showed faster convergence rate until four-order reduction of residual is achieved (about 100 time steps; Figs. 10a,b), which is as twice fast as that of the CFL=20 case. Courant numbers larger than such optimum values led to oscillatory or unstable solutions.

In addition, from Fig. 11 in which both the cases with and without sub-iterations are shown together (as "residual histories versus CPU time"), it is confirmed that using sub-iterations generally yield faster convergence. With the effect of number of sub-iterations taken into account, the choice of pLU-SGS/AUSM$^+$-up with CFL=2,000 (3 sub-iterations) (Fig. 11), again, showed the fastest convergence rate towards machine zero; for 4-order drop of residual, pLU-SGS/AUSM$^+$-up with CFL=200 (3 sub-iterations) is the fastest. Thus, *in terms of efficiency, pLU-SGS/AUSM$^+$-up appeared to be the best with the maximum allowable Courant numbers*. Based on this limited set of results, numerical dissipation in AUSM$^+$-up is compatible with that produced by LU-SGS (or pLU-SGS) for large Courant numbers, [§§] probably due to its use of M_∞.



(a) (b)

Figure 10: Residual histories with different Courant numbers for M_∞=0.01, 3 sub-iterations: a) inviscid (Case 2B) and b) viscous (Case 3) computations.



(a) (b)

Figure 11: Residual histories with different Courant numbers, with and without sub-iterations, for M_∞=0.01: a) inviscid (Case 2B) and (b) viscous (Case 3) computations.

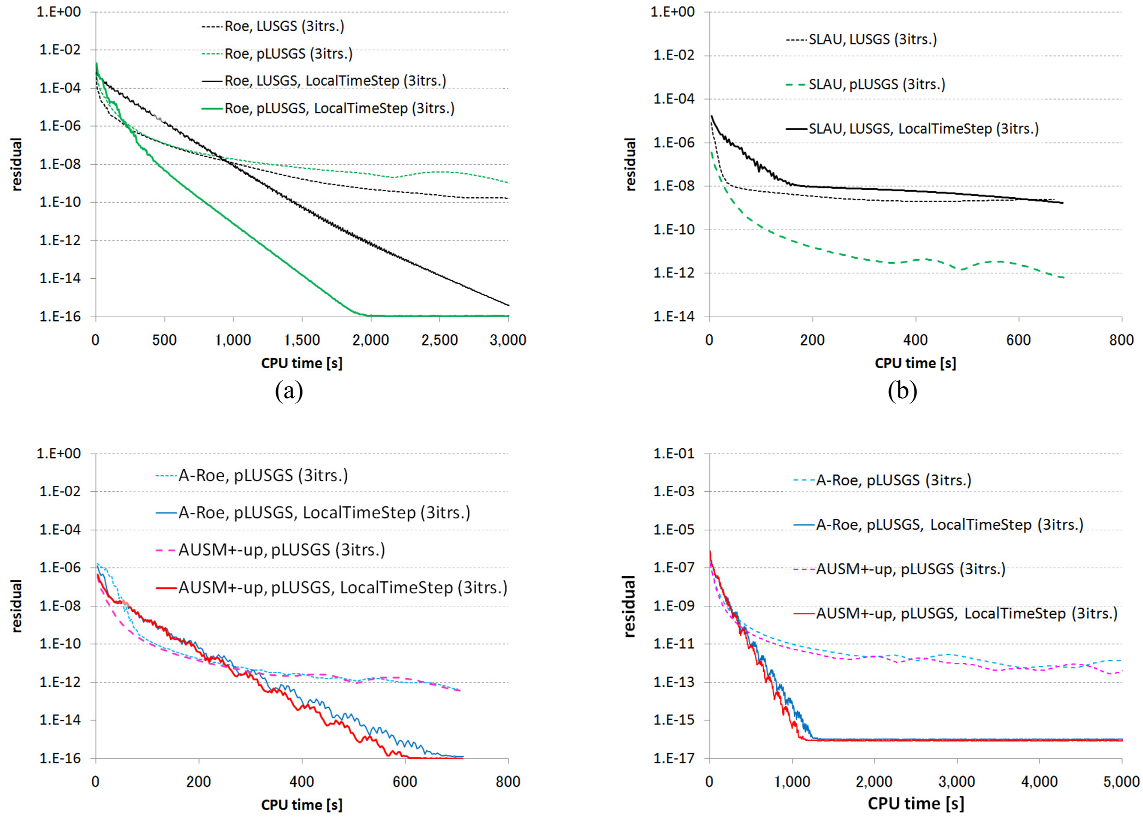
E. Effect of Local Time-Stepping

It is commonly known that the local time-stepping technique (Eq. 2.17) enhances convergence rate for steady flow computations, and, we employed this technique for our test cases here.

[§§] From Eqs. 2.14 and 2.17, the larger CFL is, the smaller the scalar **D** becomes, degrading its diagonal dominance and hence, introducing more numerical dissipation into the system of equations.

Table 8: Computed results of three sub-iteration cases (CFL=20, with local time stepping), **S** (Successful), **U** (Unphysical or oscillatory), and **F** (Failure, diverged); time step sizes Δt_i are also included for selected cases.

Sub-iterations: 3		Case 1	Case 2B	Case 3
Implicit Schemes	Euler Fluxes	M_∞=0.5 Re_∞=5,000	M_∞=0.01 Re_∞= ∞ (Inviscid)	M_∞=0.01 Re_∞=2,000
LU-SGS	Roe	**S** ($0.00456 \le \Delta t_i \le 5.30$)	-	-
	A-Roe	**S**	**F**	**F**
	AUSM$^+$-up	**S**	**F**	**F**
	SLAU	**S**	**S** ($0.00165 \le \Delta t_i \le 5.54$)	**U**
pLU-SGS	Roe	**S** ($0.00911 \le \Delta t_i \le 10.5$)	-	-
	A-Roe	**S**	**S**	**S**
	AUSM$^+$-up	**F**	**S** ($1.37 \le \Delta t_i \le 534$)	**S** ($0.465 \le \Delta t_i \le 526$)
	SLAU	**U**	**F**	**F**



(a) (b)

(c) (d)

Figure 12: Residual histories with different Courant numbers, with and without sub-iterations: a) viscous, M_∞=0.5 (Case 1), b), c) inviscid, M_∞=0.01 (Case 2B), and d) viscous, M_∞=0.01 (Case 3) computations.

The computations were conducted for selected cases and methods, and three sub-iterations were adopted. The results are summarized in Table 8, and residual histories are shown in Fig. 12.

- The portions of successful cases and others are roughly similar to the global time-stepping cases shown in Table 5.
- As shown in Fig. 12a, the local time-stepping clearly accelerated the convergence for viscous, moderate Mach number flow of M_∞=0.5 (Case 1). At this flow speed, local time-stepping appeared to be more effective than preconditioning, and this is explained from the formulation of Eqs. (2.12b), (2.13b), and (2.17). The spectral radius, $\sigma = |V_n| + c +$ (viscous term), is dominated by $|V_n|$ along with c, and $|V_n|$ varies from one cell to another with the order of c. This change is amplified by changes of cell sizes of the order of 10 or more (in this case, about 100; Fig. 2b), significantly affecting the time step Δt_i (= 0.00456-5.30, i.e., three orders magnification at maximum, as included in Table 8), compared with preconditioning $\sigma \rightarrow \sigma'$ with which the order of the spectral radius is only doubled (Δt_i = 0.00911-10.5, Table 8).
- At low Mach numbers, on the other hand, the local time-stepping is less effective than time-derivative preconditioning (Fig. 12b). Again, this is clearly explained from Eqs. (2.12b), (2.13b), and (2.17). If preconditioning technique is used at low speeds [e.g., in "SLAU, pLU-SGS (3itrs.)" in Fig. 12b (M_∞=0.01; Case 2B)], c is reduced to c' with the order of $|V_n|$ (which is orders smaller than original c), resulting in orders larger time steps in the whole computational domain (Δt = 4.70 in this case); on the contrary, if only the local time-stepping is used [as in "SLAU, LU-SGS, LocalTimeStep (3itrs.)" in Fig. 12b in which Δt_i = 0.00456-5.30], this technique has little effect on the spectral radius $\sigma = |V_n| + c$ since $|V_n| << c$, and the cell sizes play a major role. As a result, the maximum Δt_i = 5.30 (at cells away from the wall) is only slightly larger than Δt = 4.70 which is achieved by the preconditioning standalone for all the cells.
- Combination of the time-derivative preconditioning and the local time stepping is appeared to be quite effective (Figs. 12c,d), but this set led to unstable or oscillatory solutions under some conditions.

F. The "Best" Scheme

From the view point of the inviscid/subsonic drag generation as an overall error indicator, *pLUSGS/A-Roe* produced the most *accurate* solution if it successfully worked.

In terms of *efficiency, pLU-SGS/AUSM$^+$-up* appears to be the best with the maximum allowable Courant numbers.

To compare *robustness*, we simply counted numbers of successful cases marked in Tables 4, 5, and 8: pLU-SGS/A-Roe (9), pLU-SGS/AUSM$^+$-up (10), pLU-SGS/SLAU (10). Thus, *pLU-SGS/SLAU* produced more successful cases than other methods in a range of M_∞= 0.001 – 0.5, and this combination seems the most robust among all the methods.

Table 9: Evaluation of Preconditioned LU-SGS Scheme/Low-Dissipation Euler Fluxes: 5 (Excellent), 4 (Very Good), 3 (Good), 2 (Fair), and 1 (Poor).

		Accuracy	Efficiency	Robustness
pLU-SGS	A-Roe	**5**	**4**	**3**
	AUSM$^+$-up	**3**	**5**	**3**
	SLAU	**4**	**2**	**4**

Based on all the discussions above, overall ratings for each combination of preconditioned LU-SGS scheme and a low-dissipation flux is presented in Table 9. All in all, in low speed flow computations, each method is suggested for use in the following occasions:

- pLU-SGS/A-Roe: Obtaining the lowest drag error is the top priority.
- pLU-SGS/AUSM$^+$-up: One seeks the fastest convergence.
- pLU-SGS/SLAU: One is not sure whether the computation reaches stable solutions, and/or there is no reference (uniform) flow present.

Therefore, it is expected that a promising flux function can be developed if, for instance, SLAU is improved by incorporating numerical dissipation while its robustness is maintained, by using reference flow values as in A-Roe or AUSM$^+$-up only when they are available.

IV. Conclusions

We carried out a comparative study for several well-known or recently-developed low-dissipation Euler fluxes coupled with preconditioned LU-SGS (pLU-SGS) implicit scheme in the framework of steady flows. It is confirmed that pLU-SGS along with low-dissipation Euler fluxes gave accurate solutions with significant improvement of the computational efficiency. The system of non-preconditioned counterparts, on the other hand, suffered from unphysical solutions (no preconditioning at all), oscillation/slow convergence (control of dissipation in numerical flux only), or divergence of calculations (preconditioning of time integration only). The following features suggest that pLU-SGS/A-Roe, pLU-SGS/SLAU or pLU-SGS/AUSM$^+$-up combination is the best for low speed computations in terms of accuracy, efficiency, or robustness, respectively:

- [Accuracy] pLU-SGS/A-Roe yielded the minimum numerical error among the methods tested here.
- [Efficiency] The maximum allowable Courant number for pLU-SGS/AUSM$^+$-up is much larger than one for pLU-SGS/A-Roe or pLU-SGS/SLAU. With this Courant number, pLU-SGS/AUSM$^+$-up showed the fastest convergence rate.
- [Robustness] In most cases, with its optimum Courant number, pLU-SGS/SLAU gave satisfactory solutions for any Mach numbers ranging from 0.001 to 0.5.

In addition, SLAU is the only all-speed scheme which is totally free from restrictions of specifying reference values, such as cutoff Mach number M_{co} or freestream Mach number M_∞.

Therefore, it is expected that a promising flux function can be developed if, for instance, SLAU is improved by incorporating numerical dissipation while its robustness is maintained, by using reference flow values as in A-Roe or AUSM$^+$-up only when they are available.

Furthermore, local time stepping technique was proven to be effective to accelerate convergence, but its effect decreased with decreasing Mach number M_∞. At low speeds, the effect of local time stepping is recovered if it is coupled with preconditioning of time integration, but this combination led to unstable or oscillatory solutions under some conditions.

The use of other time integration methods such as pMFGS [8] or BLU-SGS [26] can be effective when coupled with the low dissipation fluxes, and performance assessment of those combinations is left as a future work.

Acknowledgments

This work was conducted as a joint research between JAXA and Iowa State University (ISU). The authors are grateful to Kozo Fujii, and other staff at JAXA and ISU both for the relevant support. The authors also thank Nobuyuki Tsuboi for having constructive discussions with us.

References

[1] Kiris, C.C., Kwak, D., Chan, W., and Housman, J.A., High-Fidelity Simulations of Unsteady Flow through Tubopumps and Flowliners," *Computers and Fluids*, Vol. 37, pp. 536-546, 2008.

[2] Tani, N., Tsuda, S., and Yamanishi, N., "Numerical Study of Cavitating Inducer in Cryogenic Fluid," 49th Japan Conference on Propulsion and Power, B14 (2009) (in Japanese).

[3] Tsuboi, N., Fukiba, K., and Shimada, T., "Numerical Simulation on Unsteady Compressible Low-Speed Flow Using Preconditioning Method: Simulation in Combustion Chamber of Hybrid Rocket," 49th Japan Conference on Propulsion and Power, B05 (2009) (in Japanese).

[4] Weiss, J.M. and Smith, W.A., "Preconditioning Applied to Variable and Constant Density Flows," *AIAA Journal*, Vol. 33, No.11, pp. 2050-2057, 1995.

[5] Turkel, E., "Preconditioning Technique in Computational Fluid Dynamics," *Annu. Rev. Fluid Mech*, Vol. 31, pp. 385-416, 1999.

[6] Liou, M.S., "A Sequel to AUSM, Part II: AUSM+-up for All Speeds," *Journal of Computational Physics,* Vol. 214, pp. 137-170, 2006.

[7] Edwards, J.R., "Towards Unified CFD Simulation of Real Fluid Flows," AIAA Paper 2001-2524, 2001.

[8] Shima, E. and Kitamura, K., "On New Simple Low- Dissipation Scheme of AUSM-Family for All Speeds," AIAA Paper 2009-136, 2009.

[9] Li, X.S. and Gu, C.W., "An All-Speed Roe-Type Scheme and Its Asymptotic Analysis of Low Mach Number Behavior," *Journal of Computational Physics*, Vol. 227, pp. 5144-5159, 2008.

[10]Jameson, A. and Turkel, E., "Implicit Schemes and LU Decompositions," Mathematics of Computation, Vol. 37, pp. 385-397, 1981.

[11]Yamamoto, S., "Preconditioning Method for Condensate Fluid and Solid Coupling Problems in General Curvilinear Coordinates," *Journal of Computational Physics*, Vol. 207, pp.240-260, 2005.

[12]Xie, F., Song, W., and Han, Z., “Numerical Study of High-Resolution Scheme Based on Preconditioning Method,” *Journal of Aircraft*, Vol. 46, No. 2, pp. 520-525, 2009.
[13]Unrau, D. and Zingg, D.W., “Viscous Airfoil Computations using Local Preconditioning,” AIAA Paper 97-2027, 1997.
[14]Shima, E., Kitamura, K., and Fujimoto, K., “New Gradient Calculation Method for MUSCL Type CFD Schemes in Arbitrary Polyhedra,” 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Orlando, AIAA Paper 2010-1081, 2010.
[15]Mavriplis, D.J., “Revisiting the Least-Squares Procedure for Gradient Reconstruction on Unstructured Meshes,” AIAA Paper 2003-3986, 2003.
[16]Wang, Z.J., “A Quadtree-based Adaptive Cartesian/Quad Grid Flow Solver for Navier-Stokes Equations,” *Computers and Fluids*, Vol. 27, No. 4, pp. 529-549, 1998.
[17]Roe, P.L., “Approximate Riemann Solvers, Parameter Vectors and Difference Schemes,” *Journal of Computational Physics*, Vol. 43, pp. 357–372, 1981.
[18]Liu, Y., and Vinokur, M., “Upwind Algorithms for General Thermo-Chemical Nonequilibrium Flows,” AIAA Paper 89-0201, 1989.
[19]Peery, K.M., and Imlay, S.T., “Blunt-Body Flow Simulations,” AIAA Paper 88-2904, 1988.
[20]Liou, M.S., “A Sequel to AUSM: AUSM+,” *Journal of Computational Physics*, Vol. 129, pp. 364-382, 1996.
[21]Shima, E., “Role of CFD in Aeronautical Engineering (No.14) - AUSM Type Upwind Schemes-,” NAL-SP30, Proceedings of 13th NAL symposium on Aircraft Computational Aerodynamics, pp. 41-46, 1996.
[22]Kitamura, K., Roe, P., and Ismail, F., “Evaluation of Euler Fluxes for Hypersonic Flow Computations,” *AIAA Journal*, Vol. 47, No. 1, pp. 44-53, 2009.
[23]Sun, Y., Wang, Z.J. and Liu, Y., “Efficient Implicit Non-linear LU-SGS Approach for Compressible Flow Computation Using High-Order Spectral Difference Method,” *Commun. Comput. Phys.*, Vol. 5, pp. 760-778, 2009.
[24]Mavriplis, D.J., Jameson, A., and Martinelli, L., “Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes,” AIAA Paper 89-0120, 1989.
[25]Liou, M.S. and Edwards, J.R., “Numerical Speed of Sound and its Application to Schemes of all Speeds,” NASA TM-1999-09286, 1999; AIAA Paper 99-3268-CP, 1999.
[26]Chen, R.F. and Wang, Z.J., “Fast, Block Lower-Upper Symmetric Gauss-Seidel Scheme for Arbitrary Grids,” *AIAA Journal*, Vol. 38, No. 12, pp. 2238-2245, 2000.
[27]Shu, C.W. and Osher, S., “Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes,” *J. Comput. Phys.*, Vol. 77, pp. 439-471, 1988.
[28]Wang, Z.J., Zhang, L, and Liu, Y., “Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids IV – Extension to Two-Dimensional Systems,” *J. Comput. Phys.*, Vol. 194, pp.716-741, 2004.

Appendix A: Weiss and Smith Preconditioning Matrix for Conservative Vector

The preconditioner of Weiss and Smith is written as follows due to Turkel [4], although this form is not used in the actual implementation.

$$\mathbf{\Gamma}^{-1} = \mathbf{I} - \frac{(1-\varepsilon)(\gamma-1)}{c^2} \cdot diag(1, \quad u, \quad v, \quad w, \quad H) \cdot \begin{pmatrix} \dfrac{u^2+v^2+w^2}{2} & -u & -v & -w & 1 \\ \dfrac{u^2+v^2+w^2}{2} & -u & -v & -w & 1 \\ \dfrac{u^2+v^2+w^2}{2} & -u & -v & -w & 1 \\ \dfrac{u^2+v^2+w^2}{2} & -u & -v & -w & 1 \\ \dfrac{u^2+v^2+w^2}{2} & -u & -v & -w & 1 \end{pmatrix} \tag{A.1}$$

$$\varepsilon = \min\ (1, \max\ (KM^2, M_{\mathrm{co}}{}^2)) \tag{A.2}$$

where K is constant usually taken as 0.25 – 1.0, and M_{co} is cutoff Mach number which is as the same order as freestream Mach number M_∞.

Appendix B: Accuracy Study - Vortex Preservation

The static and propagating vortex problems, widely known benchmark tests for Euler equations [27,28], are solved to assess spatial accuracy of the code. Euler Eqs. are solved by the following methods: spatial reconstruction using G-G with no limiter (second order in space), SLAU inviscid flux, and Shu's TVD Runge-Kutta [27] (second order in time) for time integration.

The computational setup is exactly the same as in [28].

- Mean flow: $(\rho, u, v, w, p) = (1, U_\infty, 0, U_\infty, 1)$
  where U_∞=0 for static vortex, whereas for vortex advection problem U_∞=1.
- Perturbations representing isotropic vortex:

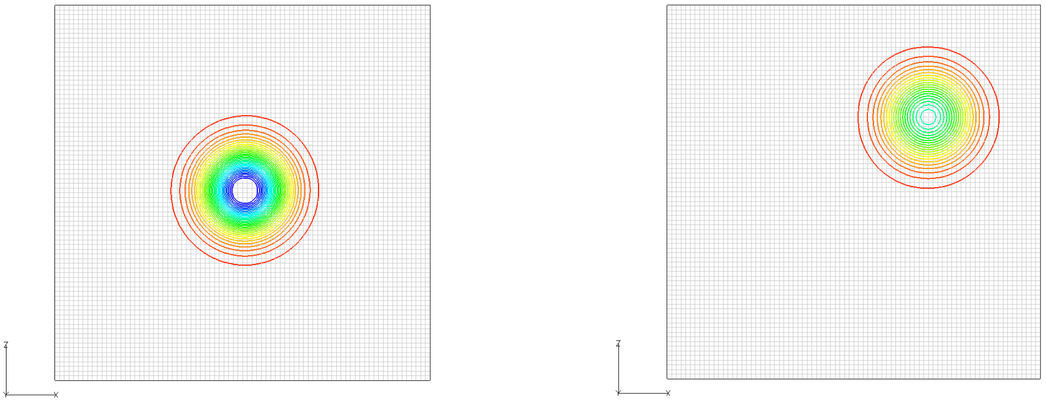
$$\left.\begin{aligned} &(\delta u,\quad \delta v,\quad \delta w) = \frac{\beta}{2\pi} e^{0.5(1-r^2)}(-z,\quad 0,\quad x), \\ &\delta T = -\frac{(\gamma-1)\beta^2}{8\gamma\pi^2} e^{1-r^2}, \quad \delta S = 0, \\ &r^2 = (x-x_0)^2 + (z-z_0)^2 \end{aligned}\right\} \qquad \text{(B.1)}$$

where center of the vortex is initially located at (x_0, y_0, z_0) = (5, 0, 5) in the computational domain of [0, 10] × [-0.5, 0.5] × [0, 10].

In the vortex advection case, the vortex moves with the speed U_∞=1 in the diagonal direction in the *x-z* plane.

As boundary conditions of what was referred to as characteristic conditions in [28], the exact solutions are imposed at all the ghost cells at each time step. Uniform Cartesian grids are used with different time step size Δt to keep Courant number almost unchanged. The numbers of cells used in each grid are presented in Tables B1 and B2. Computations are conducted until t=2.0 when the exact solution of the vortex center is located at (7, 0, 7).

Figure B1 shows the computed flows at t=2 of stationary and propagating vortex cases. In Tables B1 and B2, and Fig. B2, the L_1 norms in density difference between computed and exact values are presented for static and propagating vortex problems, respectively. One can see that the formal (second) order of accuracy is achieved in L_1 norms in both cases.



(a) **(b)**

Figure B1: Density contours of (a) initial and (b) propagated (t=2) vortex, with 80×80 grid.

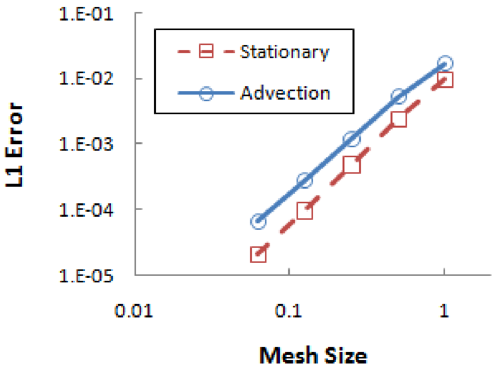


Figure B2: Solution errors for stationary and propagating vortex cases (t=2).

Table B1. Grid Refinement Study for Stationary Vortex

Grid	(Δt) × (time steps)	L_1 error	L_1 order
10 × 10	0.2 × 10	9.57E-3	-
20 × 20	0.1 × 20	2.37E-3	2.01
40 × 40	0.05 × 40	4.91E-4	2.20
80 × 80	0.025 × 80	9.70E-5	2.25
160×160	0.0125×160	2.12E-5	2.14

Table B2. Grid Refinement Study for Vortex Advection

Grid	(Δt) × (time steps)	L_1 error	L_1 order
10 × 10	0.2 × 10	1.67E-2	-
20 × 20	0.1 × 20	5.31E-3	1.77
40 × 40	0.05 × 40	1.20E-3	2.10
80 × 80	0.025 × 80	2.74E-4	2.09
160×160	0.0125×160	6.62E-5	2.03