Commun. Comput. Phys.
doi: 10.4208/cicp.041109.160910a

Vol. **10**, No. 1, pp. 90-119
July 2011

Performance of Low-Dissipation Euler Fluxes and Preconditioned LU-SGS at Low Speeds

Keiichi Kitamura[1,*], Eiji Shima[1], Keiichiro Fujimoto[1,2]
and Z. J. Wang[3]

[1] *JAXA's Engineering Digital Innovation (JEDI) Center, Japan Aerospace Exploration Agency, 3-1-1 Yoshinodai, Chuuou, Sagamihara, Kanagawa, 252-5210, Japan.*
[2] *JAXA's Engineering Digital Innovation (JEDI) Center, Japan Aerospace Exploration Agency, 2-1-1 Sengen, Tsukuba, Ibaraki, 305-8505, Japan.*
[3] *Department of Aerospace Engineering, Iowa State University, 2271 Howe Hall, Ames, IA 50011, USA.*

Received 4 November 2009; Accepted (in revised version) 16 September 2010

Available online 7 March 2011

Abstract. 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.

AMS subject classifications: 65Z05, 76M12, 76N99

Key words: All-speed scheme, low-dissipation, preconditioning, LU-SGS.

1 Introduction

In recent years, compressible finite-volume methods (FVMs) have been used in a wide spectrum of flow regimes, including low speed flows in which compressibility plays no

[*]Corresponding author. *Email addresses:* kitamura.keiichi@jaxa.jp (K. Kitamura), shima.eiji@jaxa.jp (E. Shima), fujimoto.keiichiro@jaxa.jp (K. Fujimoto), zjw@iastate.edu (Z. J. Wang)

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:

1. 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].
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 14 different approaches will be suggested for low speed flow computations.

The paper is organized as follows: in Section 2, numerical methods and flow conditions adopted here will be described. Then, in Section 3, numerical results and discussions will be presented from a viscous, moderate speed case (Case 1: M_∞=0.5, Re_∞=5,000 in 3.1), inviscid, low speeds cases (Cases 2A-2C: $M_\infty=0.1-0.001$ in 3.2), and a viscous, low speed case (Case 3: M_∞=0.01, Re_∞=2,000 in 3.3). CFL effects will also be discussed in 3.4. These computations will be conducted with global time stepping so that discussions therein could be applied (or at least referenced) to unsteady flow computations with the use of dual-time stepping in which temporal convergence is attained in each time step [4] (not actually covered in this work, though). On the other hand, it is natural to use local time stepping technique if one is interested in steady solutions. Thus, we will address the local time stepping issue as a separate investigation in 3.5. Features of each method will be summarized in 3.6, and Section 4 will conclude the present article.

2 Numerical methods and flow conditions

2.1 Governing equations

The governing equations are the compressible Navier-Stokes equations as follows, including the preconditioning matrix Γ of Weiss and Smith [4]. $\mathbf{\Gamma}$ is given in the Appendix, and it is simply eliminated in the non-preconditioned form. In general, conservative variables are commonly used in compressible flow solvers and occasionally it is not straightforward to change variables used in the whole code (our code is not an exception). Thus, we kept using conservative variables $\mathbf{Q}$ as dependent variables [8,13], instead of using primitive ones $\mathbf{q}$, although some drawbacks of using conservative variables were pointed out in [14].

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

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

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

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

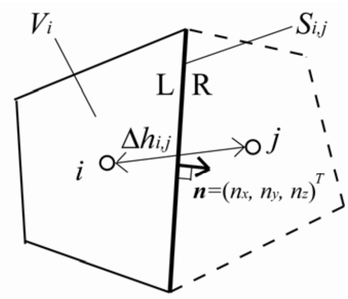


Figure 1: Schematic of cell geometric properties.

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

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

where V_i stands for the volume of the cell i, Δt_i the (local) time step, $\Delta\mathbf{Q}_i$ change of conservative variables in time, $\mathbf{F}_{i,j}$ and $\mathbf{Fv}_{i,j}$ the inviscid (Euler) and viscous fluxes through the cell-interface $S_{i,j}$ (which separates the cell i and its neighbor cell j), respectively (see Fig. 1). Details are explained below.

2.2 Numerical methods

The computational code employed here is "LS-FLOW": JAXA's in-house, unstructured, compressible Navier-Stokes solver for arbitrary polygons. 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 [18].

The Euler fluxes and the implicit schemes are summarized in Table 2. Viscous fluxes are computed by using Wang's second-order method [17]. Formulation of each Euler flux is briefly introduced below, followed by that of time evolution methods (for details, see the original literature).

Table 1: Numerical methods.

Governing Equations		Compressible Euler/Navier-Stokes Equations
Spatial Discretization		Cell-centered FVM
Spatial Reconstruction	Gradients	Green-Gauss Method [15,16] (without slope limiter)
	Inviscid Term	see "Euler Fluxes" in Table 2
	Viscous Term	Wang [17]
Temporal Evolution		see "Implicit Schemes" in Table 2

Table 2: Euler fluxes and implicit schemes.

	Baseline	Low-Dissipation/ Preconditioned (Specified reference values)
Euler Fluxes	Roe [19]	P-Roe [4] (M_{co}), A-Roe [9] ($\rho_* u_*$)
	AUSM$^+$ [20]	AUSM$^+$-up [6] (M_∞)
	SHUS [21]	SLAU [8] (No reference values required)
Implicit Schemes	LU-SGS [10]	pLU-SGS [4] (M_{co})

2.2.1 Euler fluxes

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

1) Roe [19], Preconditioned Roe (P-Roe) [4], and All-Speed-Roe (A-Roe) [9]: Using the difference of variables $\Delta() = ()_R - ()_L$, and the Roe-averaged [19] values $\hat{()}$, the Roe flux is expressed in the following form of Liu and Vinokur [22]:

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

$$\hat{\mathbf{R}}|\hat{\mathbf{\Lambda}}|\hat{\mathbf{L}}\Delta\mathbf{Q} = |\hat{\lambda_1}|\Delta\mathbf{Q} + \delta_1\hat{\mathbf{Q}}^* + \delta_2\mathbf{N}, \tag{2.3b}$$

where

$$\hat{\mathbf{Q}}^* = (1\; u\; v\; w\; H)^T, \quad \mathbf{N} = (0\; n_x\; n_y\; n_z\; V_n)^T, \quad \mathbf{\Gamma} = diag(\lambda_1, \lambda_2, \cdots, \lambda_5), \tag{2.4a}$$

$$\delta_1=\frac{\lambda^+\cdot(\Delta p/\hat{c})+\lambda^-\cdot\hat{\rho}\Delta V_n}{\hat{c}}, \qquad \delta_2=\lambda^+\cdot\hat{\rho}\Delta V_n+\lambda^-\cdot\Big(\frac{\Delta p}{\hat{c}}\Big), \tag{2.4b}$$

$$\lambda^+=-|\hat{\lambda}_1|+\frac{|\hat{\lambda}_2|+|\hat{\lambda}_3|}{2}, \qquad \lambda^-=\frac{|\hat{\lambda}_2|-|\hat{\lambda}_3|}{2}, \tag{2.4c}$$

$$\lambda_{1,4,5}=V_n, \qquad \lambda_2=V_n+c, \qquad \lambda_3=V_n-c, \tag{2.4d}$$

$$V_n=un_x+vn_y+wn_z. \tag{2.4e}$$

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

$$\begin{aligned}\hat{m}_{\frac{1}{2}}=&\frac{1}{2}\Big(\rho_L V_{nL}+\rho_R V_{nR}-|\hat{V}_n|\Delta\rho-\frac{|\hat{M}+1|+|\hat{M}-1|-2|\hat{M}|}{2}\cdot\frac{\Delta p}{\hat{c}}\\ &-\frac{|\hat{M}+1|+|\hat{M}-1|}{2}\cdot\hat{\rho}\Delta V_n\Big),\end{aligned} \tag{2.5}$$

The Roe's Riemann solver is one of the most widely-used numerical fluxes, but this flux is known to suffer from the carbuncle phenomenon [23,24] or an expansion shock at high speeds, and as demonstrated later, unphysical oscillations at low speeds.

To overcome the defect in low speed flow computations, Weiss and Smith derived a version of Roe flux for a preconditioned system, called Preconditioned Roe (P-Roe) [4], by multiplying inverse preconditioning matrix $\mathbf{\Gamma}^{-1}$ to the numerical dissipation term as follows:

$$\mathbf{F}_{\frac{1}{2}}=\frac{1}{2}\big(\mathbf{F}_L+\mathbf{F}_R-\mathbf{\Gamma}^{-1}\hat{\mathbf{R}}|\hat{\mathbf{\Lambda}}|\hat{\mathbf{L}}\Delta\mathbf{Q}\big), \tag{2.6a}$$

$$\mathbf{\Gamma}^{-1}\hat{\mathbf{R}}|\hat{\mathbf{\Lambda}}|\hat{\mathbf{L}}\Delta\mathbf{Q}=|\hat{\lambda}_1|\Delta\mathbf{Q}+\delta_1'\hat{\mathbf{Q}}^*+\delta_2'\mathbf{N}, \tag{2.6b}$$

where

$$\hat{\mathbf{Q}}^*=(1\;u\;v\;w\;H)^H, \quad \mathbf{N}=(0\;n_x\;n_y\;n_z\;V_n)^T, \quad \mathbf{\Gamma}^{-1}\mathbf{\Lambda}=diag(\lambda_1,\,\lambda_2',\,\lambda_3',\,\lambda_4,\,\lambda_5), \tag{2.7a}$$

$$\delta_1'=\Big[\frac{|\hat{\lambda}_2'|+|\hat{\lambda}_3'|}{2}-\varepsilon|V_n|-\frac{1-\varepsilon}{2}\cdot\frac{|\hat{\lambda}_2'|-|\hat{\lambda}_3'|}{2}\cdot\frac{V_n}{\hat{c}'}\Big]\cdot\frac{\Delta p}{\varepsilon\hat{c}}+\frac{|\hat{\lambda}_2'|-|\hat{\lambda}_3'|}{2}\cdot\frac{\hat{\rho}\Delta V_n}{\hat{c}'}, \tag{2.7b}$$

$$\delta_2'=\Big[\frac{|\hat{\lambda}_2'|+|\hat{\lambda}_3'|}{2}-|V_n|+\frac{1-\varepsilon}{2}\cdot\frac{|\hat{\lambda}_2'|-|\hat{\lambda}_3'|}{2}\cdot\frac{V_n}{\hat{c}'}\Big]\cdot\hat{\rho}\Delta V_n+\frac{|\hat{\lambda}_2'|-|\hat{\lambda}_3'|}{2}\cdot\frac{\Delta p}{\hat{c}'}, \tag{2.7c}$$

$$\lambda_{1,4,5}=V_n, \qquad V_{2,3}'=V_n'\pm c'=\frac{1}{2}\Big\{(1+\varepsilon)|V_n|\pm\sqrt{(\varepsilon-1)^2V_n^2+4\varepsilon c^2}\Big\}, \tag{2.7d}$$

with

$$\varepsilon=\min\big(1,\,\max(KM^2,M_{co}^2)\big), \tag{2.8}$$

where K is taken as 1.0, and M_{co} is cutoff Mach number which is set as freestream Mach number M_∞ in the present study, leading to $\varepsilon=\min(1,\max(M^2,M_\infty^2))$. The above expression is borrowed from a preconditioning matrix $\mathbf{\Gamma}$ (see Eq. (A.2) in Appendix) so that the resultant flux formula has all the eigenvalues scaled from the order of the speed of sound to that of the local fluid velocity at low speeds. Note that at supersonic speeds ($\max(M^2, M_\infty^2) > 1.0$), ε is unity and original Roe flux is recovered; otherwise, even at a moderate speed (e.g., $M = 0.5$), the eigenvalues are no longer the same as those in Eq. (2.4d) and the resultant flux is expected to behave in a different manner as unpreconditioned Roe.

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:

$$\mathbf{F}_{\frac{1}{2}} = \frac{1}{2}\Big(f(M)(\mathbf{F}_L + \mathbf{F}_R) + [1 - f(M)]\mathbf{F}_c^{press} - \hat{\mathbf{R}}|\hat{\mathbf{\Lambda}}^{A-Roe}|\hat{\mathbf{L}}\Delta\mathbf{Q}\Big), \tag{2.9a}$$

$$f(M) = \min\Big(M\frac{\sqrt{4 + (1 - M^2)^2}}{1 + M^2}, 1\Big), \tag{2.9b}$$

$$\mathbf{F}_c^{press} = \frac{U_c(\mathbf{\Phi}_L + \mathbf{\Phi}_R)}{2} + \frac{\mathbf{P}_L + \mathbf{P}_R}{2}, \tag{2.9c}$$

$$U_c = \frac{(V_{nL} + V_{nR})}{2} - \frac{c_2}{(\rho^* u^*)\cdot(p_R - p_L)}, \tag{2.9d}$$

$$\mathbf{\Phi} = (\rho,\ \rho u,\ \rho v,\ \rho w,\ \rho H)^T, \qquad \mathbf{P} = (0,\ p n_x,\ p n_y,\ p n_z,\ 0)^T, \tag{2.9e}$$

$$\lambda_{1,4,5}^{A-Roe} = V_n, \qquad \lambda_2^{A-Roe} = V_n + f(M)c, \qquad \lambda_3^{A-Roe} = V_n - f(M)c, \tag{2.9f}$$

where $\mathbf{F}_c^{press}$ is a pressure stabilization term with $c_2 = 0.05$ and $\rho_* u_* = \rho_\infty u_\infty$ (according to the original paper [9], these should be the maximum values in the whole computational domain; however, they are simply set to be freestream values here, since continuity equation justifies $\rho_* u_* = \rho_\infty u_\infty = const$ in this work). This scheme does not rely on "cut-off Mach number M_{co}", which is typically borrowed from preconditioning matrix $\mathbf{\Gamma}$ (see Eq. (A.2) in Appendix) and is included in some other all speed schemes (such as Eq. (2.8) in P-Roe [4], as explained above), though reference values $\rho_* u_*$ should be specified. Furthermore, as P-Roe, this flux is supposed to behave differently compared with original Roe even at a moderate speed.

2) AUSM$^+$ [20] 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) [23, 24]. Among AUSM-family, we introduce two representative methods, i.e., AUSM$^+$ and its all-speed extension, AUSM$^+$-up.

Formulation of AUSM$^+$ is given as:

$$\mathbf{F}_{\frac{1}{2}} = \frac{\dot{m} + |\dot{m}|}{2}\mathbf{\Psi}^+ + \frac{\dot{m} - |\dot{m}|}{2}\mathbf{\Psi}^- + \tilde{p}\mathbf{N}, \tag{2.10a}$$

$$\mathbf{\Psi} = (1,\ u,\ v,\ w,\ H)^T, \qquad \mathbf{N} = (0,\ n_x,\ n_y,\ n_z,\ 0)^T, \tag{2.10b}$$

$$\dot{m}_{\frac{1}{2}} = M_{\frac{1}{2}} c_{\frac{1}{2}} \begin{cases} \rho_L, & \text{if } M_{\frac{1}{2}} > 0, \\ \rho_R, & \text{otherwise}, \end{cases} \tag{2.10c}$$

$$M_{\frac{1}{2}} = f_{M,L}^+\big|_{\beta=\frac{1}{8}} + f_{M,R}^-\big|_{\beta=\frac{1}{8}}, \qquad \tilde{p} = f_{p,L}^+\big|_{\alpha=\frac{3}{16}} p_L + f_{p,R}^-\big|_{\alpha=\frac{3}{16}} p_R, \tag{2.10d}$$

$$f_M^{\pm}\big|_{\beta}=\begin{cases}\frac{1}{2}(M\pm|M|), & \text{if } |M|\geq 1,\\ \pm\frac{1}{4}(M\pm 1)^2\pm\beta(M^2-1)^2, & \text{otherwise,}\end{cases} \tag{2.10e}$$

$$f_p^{\pm}\big|_{\alpha}=\begin{cases}\frac{1}{2}(1\pm\text{sign}(M)), & \text{if } |M|\geq 1,\\ \frac{1}{4}(M\pm 1)^2(2\mp M)\pm\alpha M(M^2-1)^2, & \text{otherwise,}\end{cases} \tag{2.10f}$$

where

$$c_{\frac{1}{2}}=\min(\tilde{c}_L,\tilde{c}_R),\quad \tilde{c}=\frac{c^{*2}}{\max(c^*,|u|)},\quad c^{*2}=\frac{2(\gamma-1)}{(\gamma+1)}H. \tag{2.11}$$

This scheme was extended for all speeds as AUSM$^+$-up, with introduction of additional user-specified parameters:

$$\mathbf{F}_{\frac{1}{2}}=\frac{\dot{m}+|\dot{m}|}{2}\mathbf{\Psi}^+ +\frac{\dot{m}-|\dot{m}|}{2}\mathbf{\Psi}^- +\tilde{p}\mathbf{N}, \tag{2.12a}$$

$$\mathbf{\Psi}=(1,\,u,\,v,\,w,\,H)^T,\qquad \mathbf{N}=(0,\,n_x,\,n_y,\,n_z,\,0)^T, \tag{2.12b}$$

$$\dot{m}_{\frac{1}{2}}=M_{\frac{1}{2}}c_{\frac{1}{2}}\begin{cases}\rho_L, & \text{if } M_{\frac{1}{2}}>0,\\ \rho_R, & \text{otherwise,}\end{cases} \tag{2.12c}$$

$$M_{\frac{1}{2}}=f_{M,L}^{+}\big|_{\beta=\frac{1}{8}}+f_{M,R}^{-}\big|_{\beta=\frac{1}{8}}+M_p,\qquad \tilde{p}=f_{p,L}^{+}\big|_{\alpha}p_L+f_{p,R}^{-}\big|_{\alpha}p_R+p_u, \tag{2.12d}$$

where

$$c_{\frac{1}{2}}=\min(\tilde{c}_L,\tilde{c}_R),\quad \tilde{c}_L=\frac{c^{*2}}{\max(c^*,V_{nL})},\quad \tilde{c}_R=\frac{c^{*2}}{\max(c^*,-V_{nR})}, \tag{2.13a}$$

$$M_p=-\frac{K_p}{f_\alpha}\max(1-\sigma\bar{M}^2,0)\frac{p_R-p_L}{\rho_{\frac{1}{2}}c_{\frac{1}{2}}^2},\qquad \rho_{\frac{1}{2}}=\frac{\rho_L+\rho_R}{2}, \tag{2.13b}$$

$$p_u=-K_u f_{pL}^{+}f_{pR}^{-}(\rho_L+\rho_R)(f_\alpha c_{\frac{1}{2}})(V_{nR}-V_{nL}), \tag{2.13c}$$

with

$$K_p=0.25,\qquad K_u=0.75,\qquad \sigma=1.0, \tag{2.14a}$$

$$\bar{M}^2=\frac{V_{nL}^2+V_{nR}^2}{2c_{\frac{1}{2}}^2},\qquad \alpha=\frac{3}{16}(-4+5f_\alpha^2), \tag{2.14b}$$

$$f_\alpha(M_o)=M_o(2-M_0),\qquad M_o^2=\min\big(1,\max(\bar{M}^2,M_\infty^2)\big). \tag{2.14c}$$

This scheme also excludes "cutoff Mach number M_{co}", though freestream Mach number M_∞ is required[†].

[†] In this paper, as in [6, 9], "cutoff Mach number M_{co}" and "freestream Mach number M_∞" are used in different meanings, because M_{co} is not necessarily equal to M_∞, but arbitrary chosen from the values of the order of M_∞.

3) SHUS [21] and SLAU [8]: SHUS (Simple High-resolution Upwind Scheme) is one of AUSM-family schemes, which replaced mass flux of AUSM$^+$ with that of Roe (Eq. (2.5) 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.

$$\mathbf{F}_{\frac{1}{2}}=\frac{\dot{m}+|\dot{m}|}{2}\mathbf{\Psi}^{+}+\frac{\dot{m}-|\dot{m}|}{2}\mathbf{\Psi}^{-}+\tilde{p}\mathbf{N}, \tag{2.15a}$$

$$\mathbf{\Psi}=(1,\, u,\, v,\, w,\, H)^T, \qquad \mathbf{N}=(0,\, n_x,\, n_y,\, n_z,\, 0)^T. \tag{2.15b}$$

The mass flux function of SHUS is given as:

$$\begin{aligned}\dot{m}=&\frac{1}{2}\Big\{(\rho V_n)_L+(\rho V_n)_R-\Big|\frac{V_{nL}+V_{nR}}{2}\Big|\Delta\rho-\frac{|\bar{M}+1|-|\bar{M}-1|}{2}\bar{\rho}\Delta V_n\\&-\frac{|\bar{M}+1|+|\bar{M}-1|-2|\bar{M}|}{2\bar{c}}\Delta p\Big\},\end{aligned} \tag{2.16}$$

where

$$\Delta()=()_R-()_L, \qquad \bar{()}=\frac{()_L+()_R}{2}. \tag{2.17}$$

The pressure term is:

$$\tilde{p}=f^{+}_{pL}|_{\alpha=0}p_L+f^{-}_{pR}|_{\alpha=0}p_R, \tag{2.18}$$

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):

$$\mathbf{F}_{\frac{1}{2}}=\frac{\dot{m}+|\dot{m}|}{2}\mathbf{\Psi}^{+}+\frac{\dot{m}-|\dot{m}|}{2}\mathbf{\Psi}^{-}+\tilde{p}\mathbf{N}, \tag{2.19a}$$

$$\mathbf{\Psi}=(1,\, u,\, v,\, w,\, H)^T, \qquad \mathbf{N}=(0,\, n_x,\, n_y,\, n_z,\, 0)^T. \tag{2.19b}$$

The mass flux function of SLAU is:

$$\dot{m}=\frac{1}{2}\Big\{\rho_L(V_{nL}+|\bar{V}_n|^{+})+\rho_L(V_{nL}-|\bar{V}_n|^{-})-\frac{\chi}{\bar{c}}\Delta p\Big\}, \quad |\bar{V}_n|=\frac{\rho_L|V_{nL}|+\rho_R|V_{nR}|}{\rho_L+\rho_R}, \tag{2.20a}$$

$$|\bar{V}_n|^{+}=(1-g)|\bar{V}_n|+g|V_{nL}|, \qquad |\bar{V}_n|^{-}=(1-g)|\bar{V}_n|+g|V_{nR}|, \tag{2.20b}$$

$$g=-\max\big[\min(M_L,0),-1\big]\cdot\min\big[\max(M_R,0),1\big]\in[0,1], \tag{2.20c}$$

and the pressure flux is

$$\tilde{p}=\frac{p_L+p_R}{2}+\frac{f^{+}_{pL}|_{\alpha=0}-f^{-}_{pR}|_{\alpha=0}}{2}(p_L-p_R)+(1-\chi)\big(f^{+}_{pL}|_{\alpha=0}+f^{-}_{pR}|_{\alpha=0}-1\big)\frac{p_L+p_R}{2}, \tag{2.21a}$$

$$\chi=(1-M)^2, \tag{2.21b}$$

$$M=\min\left(1.0,\frac{1}{\bar{c}}\sqrt{\frac{u_L^2+v_L^2+w_L^2+u_R^2+v_R^2+w_R^2}{2}}\right), \tag{2.21c}$$

$$M=\frac{V_n}{\bar{c}}=\frac{un_x+vn_y+wn_z}{\bar{c}}. \tag{2.21d}$$

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].

2.2.2 Time evolution methods

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:

$$\frac{V_i}{\Delta t_i}\Delta\mathbf{Q}_i^n+\mathbf{\Gamma}_i^{-1}\sum_j(\Delta\mathbf{F}_{i,j}^n-\Delta\mathbf{Fv}_{i,j}^n)S_{i,j}=\mathbf{\Gamma}_i^{-1}\cdot\mathbf{Res}_i^n, \tag{2.22}$$

where

$$\Delta()^n=()^{n+1}-()^n, \tag{2.23}$$

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

$$\mathbf{Res}_i^n=\sum_j(\mathbf{F}_{i,j}^n-\mathbf{Fv}_{i,j}^n)S_{i,j}. \tag{2.24}$$

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

Then, Eq. (2.22) is rewritten in the form of Gauss-Seidel (GS) iterative method by decomposition of new (updated) and old (non-updated) values

$$\begin{aligned}\Delta\mathbf{Q}_i^{n+1}=\mathbf{D}_i^{n-1}\cdot\Big[\mathbf{\Gamma}_i^{-1}\sum_{j\in Lower}S_{i,j}\mathbf{A}_{j,i}^{+\,new/old}\Delta\mathbf{Q}_j^{new/old}\\+\mathbf{\Gamma}_i^{-1}\sum_{j\in Upper}S_{i,j}\mathbf{A}_{j,i}^{+\,new/old}\Delta\mathbf{Q}_j^{new/old}-\mathbf{\Gamma}_i^{-1}\cdot\mathbf{Res}_i^n\Big],\end{aligned} \tag{2.25}$$

where

$$\mathbf{A}_{i,j}=\frac{\partial\mathbf{F}_{i,j}}{\partial\mathbf{Q}_i}-\frac{\partial\mathbf{Fv}_{i,j}}{\partial\mathbf{Q}_i}, \tag{2.26}$$

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

$$\mathbf{D}_i=\frac{V_i}{\Delta t_i}\mathbf{I}+\mathbf{\Gamma}_i^{-1}\sum_j S_{i,j}\mathbf{A}_{i,j}^+. \tag{2.27}$$

Specifically, in LU-SGS (Lower Upper Symmetric Gauss-Seidel), Eq. (2.25) is further rewritten as

$$\Delta\mathbf{Q}_i^* = \mathbf{D}_i^{-1}\cdot\left[\mathbf{\Gamma}_i^{-1}\sum_{j\in Lower} S_{i,j}\mathbf{A}_{j,i}^{+\,*}\Delta\mathbf{Q}_j^* - \mathbf{\Gamma}_i^{-1}\cdot\mathbf{Res}_i^n\right], \tag{2.28a}$$

$$\begin{aligned}\Delta\mathbf{Q}_i^{n+1} &= \mathbf{D}_i^{-1}\cdot\left[\mathbf{\Gamma}_i^{-1}\sum_{j\in Lower} S_{i,j}\mathbf{A}_{j,i}^{+\,*}\Delta\mathbf{Q}_j^* + \mathbf{\Gamma}_i^{-1}\sum_{j\in Upper} S_{i,j}\mathbf{A}_{j,i}^{+\,n+1}\Delta\mathbf{Q}_j^{n+1} - \mathbf{\Gamma}_i^{-1}\cdot\mathbf{Res}_i^n\right]\\ &= \Delta\mathbf{Q}_i^* + \mathbf{D}_i^{-1}\cdot\mathbf{\Gamma}_i^{-1}\sum_{j\in Upper} S_{i,j}\mathbf{A}_{j,i}^{+\,n+1}\Delta\mathbf{Q}_j^{n+1}.\end{aligned} \tag{2.28b}$$

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").

$$\mathbf{A}_{i,j}^+ \approx \frac{\mathbf{A}_{i,j}+\sigma_{i,j}\mathbf{I}}{2}, \tag{2.29}$$

where

$$\sigma_{i,j} = \sigma_{i,j}(\mathbf{A}_{i,j}) = |V_{ni,j}| + c_i + \frac{2(\mu_i+\mu_j)}{(\rho_i+\rho_j)\cdot\Delta h_{i,j}}, \tag{2.30}$$

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,

$$\mathbf{\Gamma}_i^{-1}\mathbf{A}_{i,j}^+ \approx \frac{\mathbf{\Gamma}_i^{-1}\mathbf{A}_{i,j}+\sigma'_{i,j}\mathbf{I}}{2}, \tag{2.31}$$

where

$$\sigma'_{i,j} = \sigma'(\mathbf{\Gamma}_i^{-1}\mathbf{A}_{i,j}) = \frac{1}{2}\left\{(1+\varepsilon)|V_{ni,j}| + \sqrt{(\varepsilon-1)^2V_{ni,j}^2+4\varepsilon c_i^2}\right\} + \frac{2(\mu_i+\mu_j)}{(\rho_i+\rho_j)\cdot\Delta h_{i,j}}. \tag{2.32}$$

The preconditioning coefficient ε, which should be of the order of M_∞^2, appears in the preconditioning matrix as given by $\varepsilon = \min(1,\max(KM^2,M_{co}^2))$ (Eq. (A.3) in Appendix). Variables in this equation can be arbitrary chosen, and here, K is taken as unity and $M_{co} = M_\infty$, leading to

$$\varepsilon = \min\left(1,\max(M^2,M_\infty^2)\right). \tag{2.33}$$

With the above approximation, the diagonal block $\mathbf{D}_i$ (in Eq. (2.27))) is transformed into merely a scalar

$$\mathbf{D}_i = \frac{V_i}{\Delta t_i}\mathbf{I} + \sum_j S_{i,j}\mathbf{\Gamma}_i^{-1}\mathbf{A}_{i,j}^+ = \left(\frac{V_i}{\Delta t_i} + \sum_j S_{i,j}\frac{\sigma'_{i,j}}{2}\right)\mathbf{I}, \tag{2.34a}$$

$$\because \quad \sum_j S_{i,j}\mathbf{A}_{i,j} = 0. \tag{2.34b}$$

Substituting Eqs. (2.26), (2.31), and (2.34a), Eqs. (2.28a) and (2.28b) becomes

$$\begin{aligned}\Delta\mathbf{Q}_i^*&=\Big(\frac{V_i}{\Delta t_i}+\sum_j S_{i,j}\frac{\sigma'_{i,j}}{2}\Big)^{-1}\cdot\Big[\boldsymbol{\Gamma}_i^{-1}\sum_{j\in Lower}S_{i,j}\mathbf{A}_{j,i}^{+*}\Delta\mathbf{Q}_j^*-\boldsymbol{\Gamma}_i^{-1}\cdot\mathbf{Res}_i^n\Big]\\&=\Big(\frac{V_i}{\Delta t_i}+\sum_j S_{i,j}\frac{\sigma'_{i,j}}{2}\Big)^{-1}\cdot\Big[\sum_{j\in Lower}S_{i,j}\frac{\boldsymbol{\Gamma}_i^{-1}\mathbf{A}_{j,i}+\sigma'_{j,i}\mathbf{I}}{2}\Delta\mathbf{Q}_j^*-\boldsymbol{\Gamma}_i^{-1}\cdot\mathbf{Res}_i^n\Big]\\&=\Big(\frac{V_i}{\Delta t_i}+\sum_j S_{i,j}\frac{\sigma'_{i,j}}{2}\Big)^{-1}\cdot\Big[\frac{1}{2}\sum_{j\in Lower}S_{i,j}\big(\boldsymbol{\Gamma}_i^{-1}(\Delta\mathbf{F}_{j,i}^*-\Delta\mathbf{Fv}_{j,i}^*)+\sigma'_{j,i}\Delta\mathbf{Q}_j^*\big)-\boldsymbol{\Gamma}_i^{-1}\cdot\mathbf{Res}_i^n\Big],\qquad(2.35a)\\\Delta\mathbf{Q}_i^{n+1}&=\Delta\mathbf{Q}_i^*+\Big(\frac{V_i}{\Delta t_i}+\sum_j S_{i,j}\frac{\sigma'_{i,j}}{2}\Big)^{-1}\cdot\boldsymbol{\Gamma}_i^{-1}\sum_{j\in Upper}S_{i,j}\mathbf{A}_{j,i}^{+n+1}\Delta\mathbf{Q}_j^{n+1}\\&=\Delta\mathbf{Q}_i^*+\Big(\frac{V_i}{\Delta t_i}+\sum_j S_{i,j}\frac{\sigma'_{i,j}}{2}\Big)^{-1}\cdot\frac{1}{2}\sum_{j\in Upper}S_{i,j}\big(\boldsymbol{\Gamma}_i^{-1}(\Delta\mathbf{F}_{j,i}^{n+1}-\Delta\mathbf{Fv}_{j,i}^{n+1})+\sigma'_{j,i}\Delta\mathbf{Q}_j^*\big).\qquad(2.35b)\end{aligned}$$

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.

$$\boldsymbol{\Gamma}^{-1}\Delta\mathbf{F}=\Delta\mathbf{F}-\frac{(1-\varepsilon)dp}{c^2}(1\ u\ v\ w\ H)^T,\qquad(2.36a)$$

$$dp=(\gamma-1)\Big(\frac{u^2+v^2+w^2}{2}\Delta\mathbf{F}_1-u\Delta\mathbf{F}_2-v\Delta\mathbf{F}_3-w\Delta\mathbf{F}_4+\Delta\mathbf{F}_5\Big),\qquad(2.36b)$$

where $\mathbf{F}_l$ stands for the l-th row of $\Delta\mathbf{F}$ (e.g., $\Delta\mathbf{F}_1=\Delta(\rho V_n)$).

With the above expression one can easily obtain preconditioned variables by avoiding actual matrix operation of $\boldsymbol{\Gamma}^{-1}\Delta\mathbf{F}$.

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

$$\Delta t_i=\frac{CFL*V_i}{\big[\sum_j\sigma_{i,j}S_{i,j}\big]},\qquad(2.37)$$

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

The use of Eq. (2.37)) is called local time stepping (used in 3.5), whereas the global time stepping (used in 3.1-3.4) takes the form of

$$\Delta t=\frac{CFL}{\max_{i,j}(\sigma_{i,j}/\Delta h_{i,j})}.\qquad(2.38)$$

3) Sub-iteration procedure: The sub-iteration (sometimes called Newton-iteration) is used to enhance the convergence rate outside the LU-SGS loop. Eq. (2.22),

$$\frac{V_i}{\Delta t_i}\Delta\mathbf{Q}_i^n+\boldsymbol{\Gamma}_i^{-1}\sum_j(\Delta\mathbf{F}_{i,j}^n-\Delta\mathbf{Fv}_{i,j}^n)S_{i,j}=\boldsymbol{\Gamma}_i^{-1}\cdot\mathbf{Res}_i^n,\qquad(2.39)$$

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

$$\frac{3\mathbf{Q}^{m+1}-4\mathbf{Q}^n+\mathbf{Q}^{n-1}}{2}=\frac{3\Delta\mathbf{Q}^m+3\mathbf{Q}^m-4\mathbf{Q}^n+\mathbf{Q}^{n-1}}{2} \tag{2.40}$$

$$=-\frac{\Delta t}{V}\mathbf{\Gamma}^{-1}\cdot\left\{\mathbf{Res}^m-(\Delta\mathbf{F}^n-\Delta\mathbf{Fv}^n)S\right\}. \tag{2.41}$$

Thus,

$$\Delta\mathbf{Q}^m=-\frac{2}{3}\left[\frac{3\mathbf{Q}^m-4\mathbf{Q}^n+\mathbf{Q}^{n-1}}{2}+\frac{\Delta t}{V}\mathbf{\Gamma}^{-1}\cdot\left\{\mathbf{Res}^m-(\Delta\mathbf{F}^n-\Delta\mathbf{Fv}^n)S\right\}\right], \tag{2.42}$$

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 but used sub-iterations only to accelerate and stabilize computations of steady flows[‡]. The same idea is shared by developers of OVERFLOW code (see [25] and Private Communication with Dr. Chris Nelson, Mar. 2010).

In the subsequent sections, CFL 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.

Table 3: Test cases and conditions.

Cases	Conditions	Comments
1) Viscous	$M_\infty=0.5$, $Re_\infty=5{,}000$ [26,27]	Moderate Speed (Validation)
2) Inviscid	$M_\infty=0.001-0.1$	Low Speeds
3) Viscous	$M_\infty=0.01$, $Re_\infty=2{,}000$ [3,11]	Low Speed

2.3 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. In order to highlight differences among methods, we generated the following two grids that are not too fine or too coarse (Fig. 2):

[‡] We found that three inner-iterations were enough to accelerate convergence and/or stabilize computations for steady state computations with significant reduction (2 orders drop) of RHS residual at the third inner loop. Further iterations reduced residual slightly more, but did not improve accuracy or convergence dramatically.

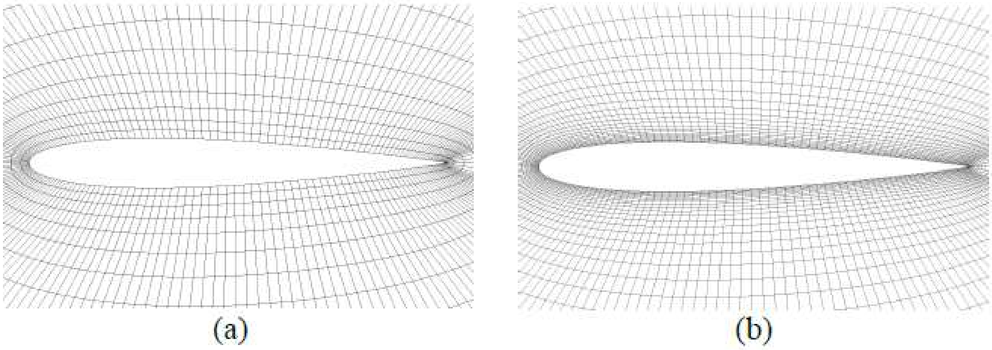


Figure 2: Computational grids for (a) inviscid (201×31 points) and (b) viscous (201×51 points) simulations.

- 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 $\delta = 1.0e-3$, 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.

3 Results and discussions

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

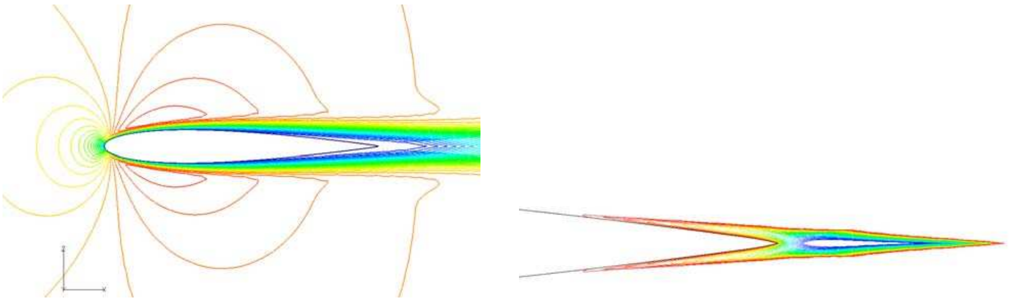
- S (Successful): The L_2-norm of density residual dropped at least four orders with physically correct solution.
- U (Unsatisfactory): The solution reached an unphysical one with poor quality, and/or the residual remained significant.
- F (Failure): The calculation diverged.

In the Case 1, separations points (designated as % chord length from the leading edge) are also included in the table. The result of each case will be discussed in the subsequent sub-sections.

3.1 Case 1: $M_\infty = 0.5$ viscous (laminar) flow for code validation

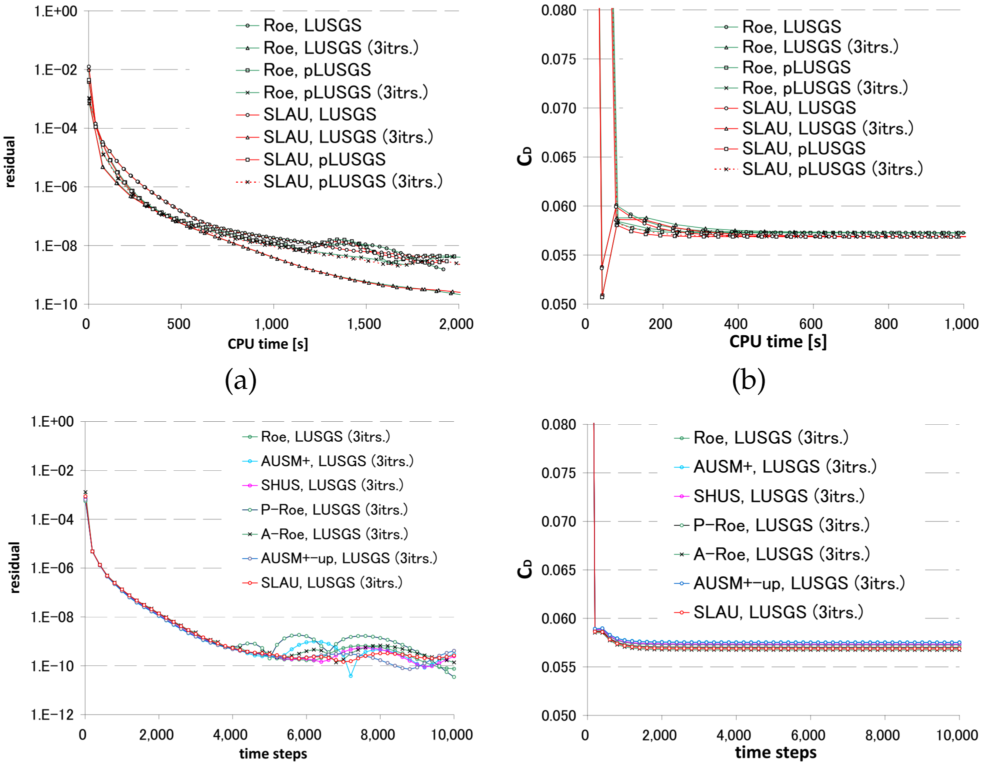
This test case has been widely used as a benchmark [26, 27]. The computations were conducted for 10,000 timesteps. Typical computed flow field is displayed in Fig. 3. Fig. 4 shows histories of drag coefficient C_D and L_2-norm of density residuals for the successful cases.

For all the successful cases, the computed flows were almost identical to each other, with slightly different separation points [26, 27] near the trailing edge. These locations,



(a) (b)

Figure 3: Computed flow field by LU-SGS/SLAU, no sub-iterations, Case 1 ($M_\infty = 0.5$, 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 ($M_\infty = 0.5$, viscous): (a) Residual vs. CPU time, (b) C_D vs. CPU time, (c) Residual vs. time steps, and (d) C_D vs. time steps.

included in Tables 4 and 5, are in good agreement with reference separation points of 80%-89% chord length (comparisons to other methods in literature were made in a separate work [18]).

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	$M_\infty=0.5$	A) $M_\infty=0.1$	B) $M_\infty=0.01$	C) $M_\infty=0.001$	$M_\infty=0.01$
		$Re_\infty=5{,}000[Sep.point(\%)]$	$Re_\infty=\infty$ (Inviscid)			$Re_\infty=2{,}000$
LU-SGS	Roe	**S**[84.9]	**U**			**U**
	AUSM$^+$	**S**[85.0]				
	SHUS	**S**[84.9]				
	P-Roe	**S**[83.8]	**F**	**F**	**F**	**F**
	A-Roe	**S**[82.7]	**S**	**F**	**F**	**F**
	AUSM$^+$-up	**S**[84.9]	**F**	**F**	**F**	**F**
	SLAU	**S**[83.8]	**S**	**S**	**S**	**S**
LU-SGS	Roe	**S**[84.9]	**F**			**F**
	AUSM$^+$	**F**				
	SHUS	**U**[89.7]				
	P-Roe	**S**[83.8]	**S**	**S**	**S**	**S**
	A-Roe	**S**[82.7]	**F**	**F**	**F**	**S**
	AUSM$^+$-up	**F**	**F**	**S**	**S**	**S**
	SLAU	**S**[83.8]	**S**	**S**	**S**	**S**

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	$M_\infty=0.5$	A) $M_\infty=0.1$	B) $M_\infty=0.01$	C) $M_\infty=0.001$	$M_\infty=0.01$
		$Re_\infty=5{,}000[Sep.point(\%)]$	$Re_\infty=\infty$ (Inviscid)			$Re_\infty=2{,}000$
LU-SGS	Roe	**S**[84.9]	**U**			**U**
	AUSM$^+$	**S**[84.9]				
	SHUS	**S**[84.9]				
	P-Roe	**S**[84.1]	**F**	**F**	**F**	**F**
	A-Roe	**S**[83.0]	**S**	**F**	**F**	**F**
	AUSM$^+$-up	**S**[84.9]	**F**	**F**	**F**	**F**
	SLAU	**S**[83.8]	**S**	**S**	**S**	**S**
LU-SGS	Roe	**S**[84.9]	**F**			**F**
	AUSM$^+$	**F**				
	SHUS	**U**[85.4]				
	P-Roe	**S**[83.8]	**S**	**S**	**S**	**S**
	A-Roe	**S**[82.6]	**S**	**S**	**S**	**S**
	AUSM$^+$-up	**F**	**F**	**S**	**S**	**S**
	SLAU	**S**[83.8]	**S**	**S**	**S**	**S**

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

- Coupled with LU-SGS, all the fluxes yielded physically correct solutions.
- 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. 4(a) and (b)). Even worse, calculations diverged in some cases (AUSM$^+$, SHUS, and AUSM$^+$-up, see Tables 4 and 5). This would be because i) some combinations, such as pLU-SGS/AUSM$^+$, clashed and produced an insufficient amount of dissipation (explained later), or ii) the scaling function of AUSM$^+$-up did not work well in conjunction with pLU-SGS under the present flow conditions.
- Effect of Euler fluxes seemed to be minor (Figs. 4(c) and (d)), compared with the above mentioned factors.

For further code validation, we also conducted $M=2.0$, $\alpha=0$, inviscid flow computations on a grid in Fig. 2(a). We point out that all the 14 methods produced similar and satisfactory results with the help of Venkatakrishnan's limiter [28] with Wang's correction ($\varepsilon'=0.05$) [29] (not shown).

3.2 Case 2: Low speed ($M_\infty=0.1$, 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: $M_\infty=0.1$ (Case 2A), 0.01 (2B), and 0.001 (2C). Solutions and convergence rates are compared for different methods. Fig. 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 with $M_\infty=0.01$ (Case 2B), the drag coefficient history reached a plateau at a significant value (Fig. 6(b)) with an apparently unphysical solution shown in Fig. 5(a), even though the corresponding density residual showed three orders of reduction as well as other baseline fluxes cases (Fig. 7). The final values of computed drag coefficients in all the runs are summarized in Table 6.

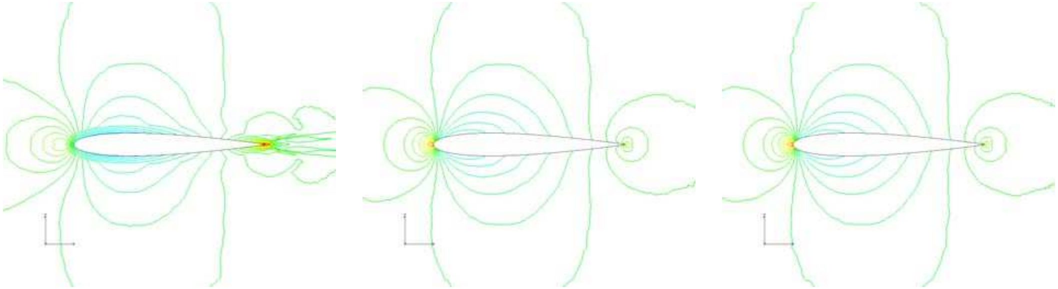


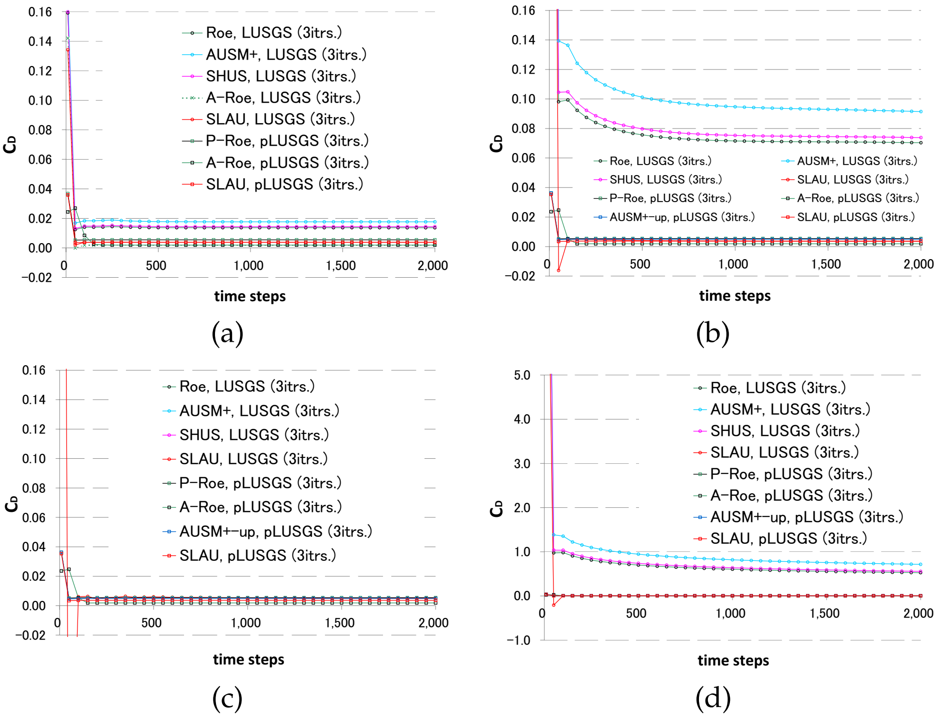
Figure 5: Computed flow fields ($-1<C_p<1$), no sub-iterations, Case 2B ($M_\infty=0.01$, inviscid): (a) LU-SGS/Roe (Unphysical), (b) LU-SGS/SLAU (Stable, slow convergence), and (c) pLU-SGS/SLAU (Stable, fast convergence).

From those Figs. 5 and 6 and Tables 4 and 5, the following general remarks are confirmed:

(a) If no-preconditioned system of Eq. (2.1) without $\mathbf{\Gamma}$ is solved, such as LU-SGS/Roe, calculations do not diverge, but reach unphysical solutions due to excessive numerical dissipation of the method (Fig. 5(a)) [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 [4,8,30].

(c) If only preconditioning B) (numerical flux preconditioning) is used, such as LU-SGS/P-Roe, calculations diverge in most cases; the only exception is LU-SGS/SLAU combination, showing slow convergence but a stable solution (Fig. 5(b)) [3,8].



(a) (b) (c) (d)

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

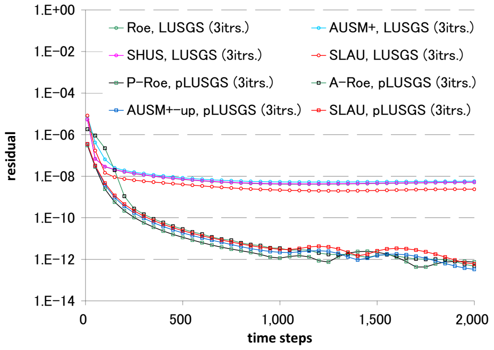


Figure 7: Residual histories, Case 2B ($M_\infty = 0.01$, inviscid), 3 sub-iterations.

(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. 5(c)). The pLU-SGS/A-Roe diverged without sub-iterations, but it was expectedly cured by introduction of sub-iterations (Recall that, as stated above, P-Roe, A-Roe and Roe are supposed to behave in different manners at subsonic speeds). The pLU-SGS/AUSM$^+$-up, however, failed to reach a solution at $M_\infty = 0.1$ as in the Case 1 ($M_\infty = 0.5$, viscous case) even with sub-iterations.

These remarks are summarized in Table 7.

Table 6: Calculated drag coefficient in inviscid flow (= numerical error) over NACA0012 airfoil, Case 2 (M_∞=0.1, 0.01, and 0.001; inviscid).

		Case 2, Sub-iterations: 1			Case 2, Sub-iterations: 3		
Implicit Schemes	Euler Fluxes	A) $M_\infty=0.1$	B) $M_\infty=0.01$	C) M_∞=0.001	A) $M_\infty=0.1$	B) $M_\infty=0.01$	C) M_∞=0.001
		$Re_\infty=\infty$ (Inviscid)			$Re_\infty=\infty$ (Inviscid)		
LU-SGS	Roe	0.0137	0.0720	0.6280	0.0137	0.0704	0.5267
	AUSM$^+$	0.0177	0.0956	0.8442	0.0177	0.0915	0.7157
	SHUS	0.0144	0.0758	0.6621	0.0144	0.0738	0.5565
	P-Roe	(Diverged)	(Diverged)	(Diverged)	(Diverged)	(Diverged)	(Diverged)
	A-Roe	0.0019	(Diverged)	(Diverged)	0.0019	(Diverged)	(Diverged)
	AUSM$^+$-up	(Diverged)	(Diverged)	(Diverged)	(Diverged)	(Diverged)	(Diverged)
	SLAU	0.0038	0.0041	0.0069	0.0038	0.0037	0.0051
pLU-SGS	Roe	(Diverged)			(Diverged)		
	AUSM$^+$						
	SHUS						
	P-Roe	0.0055	0.0055	0.0055	0.0055	0.0055	0.0055
	A-Roe	(Diverged)	(Diverged)	(Diverged)	0.0019	0.0019	0.0019
	AUSM$^+$-up	(Diverged)	0.0049	0.0049	(Diverged)	0.0049	0.0049
	SLAU	0.0038	0.0037	0.0036	0.0038	0.0037	0.0036

Table 7: Summary of typical computed results (expect for a few exceptions), S (Successful), U (Unphysical or oscillatory), and F (Failure, diverged).

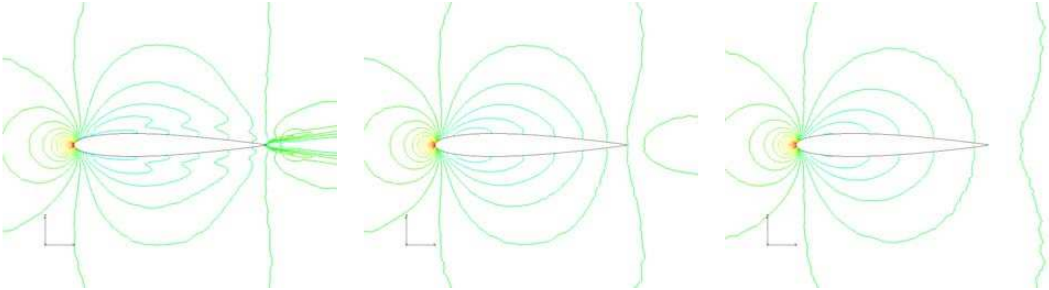
		Euler Fluxes	
		Baseline (Roe, AUSM$^+$, SHUS)	Low-Dissipation (P-Roe, A-Roe, AUSM$^+$-up, SLAU) (Preconditioning B)
Implicit Schemes	LU-SGS	**U**	**S** (slow convergence, SLAU) or **F** (other fluxes)
	pLU-SGS (Preconditioning A)	**F**	**S** (fast convergence)

According to Tables 4 and 5, qualitative performances of most of the methods presented here were independent on the Mach number. Quantitatively, however, the drag coefficients for non-preconditioned cases increased with decreasing Mach number, from the similar order to that of the preconditioned cases ($M_\infty=0.1$) to orders larger ($M_\infty=0.001$), indicating the recommended lower limit of non-preconditioned methods for use be over $M_\infty=0.1$. Meanwhile, those C_D values for the preconditioned cases stayed almost 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), pLU-SGS/AUSM$^+$-up (0.0049), and pLU-SGS/P-Roe (0.0055), in the $M_\infty=0.01$, three sub-iteration cases (the sub-iteration did not seem to affect solutions significantly) (Table 6). Therefore, at least from these data, pLU-SGS/A-Roe appeared to have produced the most accurate solution if it successfully worked.

In addition, it is confirmed that a pLU-SGS/Low-Dissipation-Flux combination can handle even $M_\infty=0.001$ flow. Specifically, pLU-SGS/P-Roe and pLU-SGS/SLAU produced more successful results than other methods in a range of $M_\infty=0.001-0.5$.

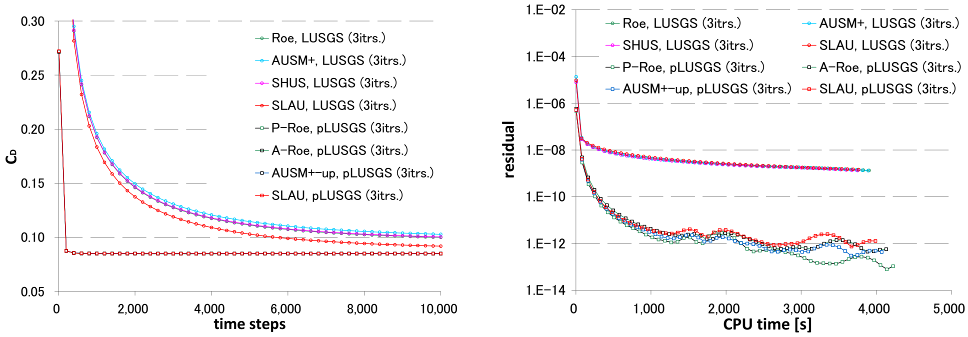
3.3 Case 3: Low speed ($M_\infty=0.01$), viscous flow

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



(a) (b) (c)

Figure 8: Computed flow fields ($-1<C_p<1$), 3 sub-iterations, Case 3 ($M_\infty=0.01$, 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 ($M_\infty=0.01$, viscous), 3 sub-iterations: (a) C_D, (b) Residual.

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, 6(b), and 7):

- Non-preconditioned cases showed excessive drag and/or slow convergence.
- Fully preconditioned computations yielded satisfactory solutions with fast convergence.
- SLAU is again only one flux which showed accepted solutions without time-derivative preconditioning (Preconditioning A), in expense of slow convergence, though.

Therefore, viscous effects played a minor role in the present cases with only one exception: A-Roe flux (without sub-iterations) yielded a satisfactory solution only in the viscous case, probably because its pressure stabilization term (Eqs. (2.9c)-(2.9e)) in combination with the viscous source term (**Fv** in Eq. (2.1b)) had a favorable contribution to the solution.

We also point out that the difference between inviscid and viscous cases will be enhanced with the use of local time stepping as shown later in 3.5.

Table 8: Summary of CFL Effect, S (Successful), U (Unphysical or oscillatory), and F (Failure, diverged).

Sub-iterations: 1		Case 2B) $M_\infty = 0.01$, $Re_\infty = \infty$ (Inviscid) [C_D]				Case 3) $M_\infty = 0.01$, $Re_\infty = 2{,}000$			
Implicit Scheme	Euler Fluxes	$CFL = 20$	200	2,000	20,000	$CFL = 20$	200	2,000	20,000
pLU-SGS	P-Roe	**S** [0.0055]	**S** [0.0055]	**S** [0.0055]	**S** [0.0055]	**S**	**S**	**S**	**S**
	A-Roe	**F**	**F**	**F**	**F**	**S**	**S**	**S**	**S**
	AUSM$^+$-up	**S** [0.0049]	**S** [0.0049]	**S** [0.0049]	**U** [0.0035]	**S**	**S**	**S**	**U**
	SLAU	**S** [0.0037]	**F**	**F**	**F**	**S**	**F**	**F**	**F**

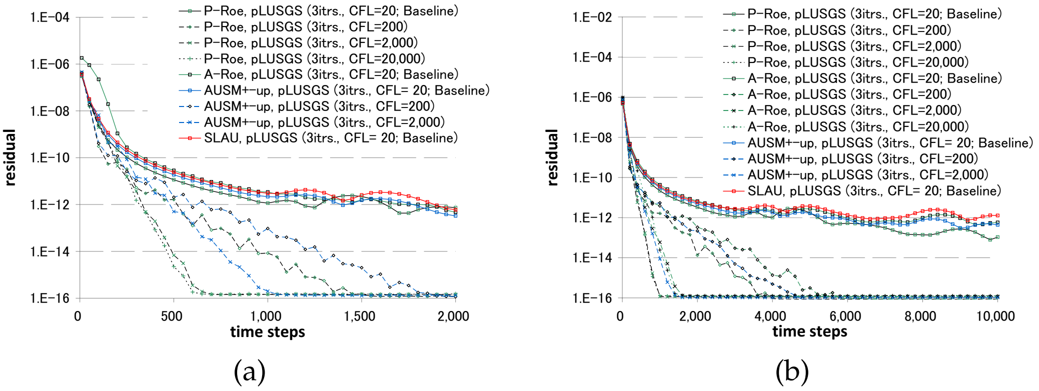
Table 9: Summary of CFL Effect, S (Successful), U (Unphysical or oscillatory), and F (Failure, diverged).

Sub-iterations: 3		Case 2B) $M_\infty = 0.01$, $Re_\infty = \infty$ (Inviscid) [C_D]				Case 3) $M_\infty = 0.01$, $Re_\infty = 2{,}000$			
Implicit Scheme	Euler Fluxes	$CFL = 20$	200	2,000	20,000	$CFL = 20$	200	2,000	20,000
pLU-SGS	P-Roe	**S** [0.0055]	**S** [0.0055]	**S** [0.0055]	**S** [0.0055]	**S**	**S**	**S**	**S**
	A-Roe	**S**[0.0019]	**F**	**F**	**F**	**S**	**S**	**S**	**S**
	AUSM$^+$-up	**S** [0.0049]	**S** [0.0049]	**S** [0.0049]	**F**	**S**	**S**	**S**	**F**
	SLAU	**S** [0.0037]	**F**	**F**	**F**	**S**	**F**	**F**	**F**

3.4 Effect of CFL numbers

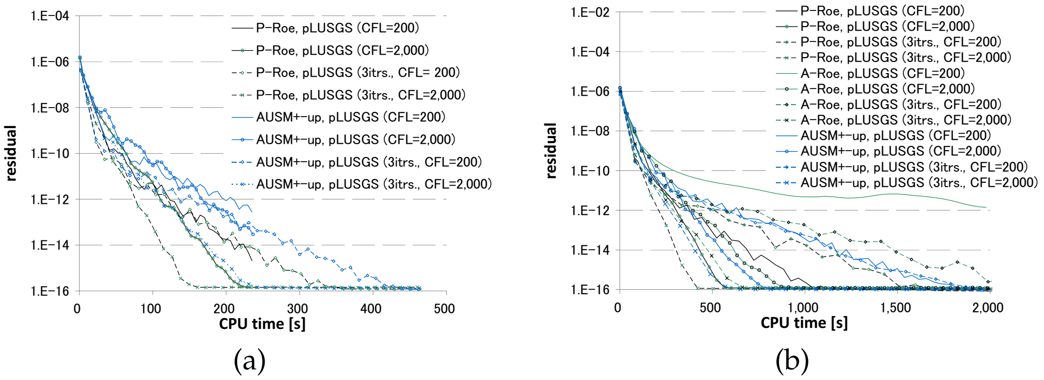
Now that we have confirmed pLU-SGS/Low-dissipation-flux combinations are recommended for low speeds and that their performances are almost irrelevant to Mach number at least in the range of $M_\infty = 0.1 - 0.001$, we compared convergence rates of these combinations with different CFL numbers ranging from 20 to 20,000 for $M_\infty = 0.01$, both in inviscid (Case 2B) and viscous (Case 3) cases. The results are summarized in Tables 8 and 9, respectively. According to the results, the larger the CFL number, the more oscillatory or unstable the computation tends to be. Fig. 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/P-Roe with $CFL = 2{,}000$ and $CFL = 20{,}000$ (with no remarkable difference), followed by 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 these combinations of methods with $CFL = 200$ achieved faster convergence for four-order reduction of residual (about 100-200 time steps; Figs. 10(a), and (b)), which is twice as fast as that of the $CFL = 20$ case. Thus, in general, CFL=200-2,000 appeared to be ″optimum.″ CFL larger than such values led to stagnated acceleration of convergence (e.g., $CFL = 20{,}000$ with P-Roe), or oscillatory or unstable solutions (e.g., $CFL = 20{,}000$ for fluxes other than P-Roe). SLAU is the exception which showed the smallest maximum allowable $CFL = 20$ among four low-dissipation fluxes for stable computations.

In Fig. 11, only successful cases with $CFL = 200$ or 2,000 both with and without sub-iterations are shown together for inviscid (Case 2B) and viscous (Case 3) computations, again (as ″residual histories versus CPU time″). It is confirmed that using sub-iterations generally yielded faster convergence. With the effect of number of sub-iterations taken into account, the choice of pLU-SGS/P-Roe with $CFL = 2{,}000$ (3 sub-iterations), followed by pLU-SGS/P-Roe with $CFL = 2{,}000$ (no sub-iterations) (inviscid, Case 2B) or pLU-



(a) (b)

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



(a) (b)

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

SGS/AUSM$^+$-up with CFL=2,000 (3 sub-iterations) (viscous, Case 3), showed the fastest convergence rate towards machine zero; for 4-order drop of residual, pLU-SGS/P-Roe with $CFL = 200$ (3 sub-iterations) is the fastest in the inviscid cases (Case 2B), whereas in the viscous simulations (Case 3) pLU-SGS/P-Roe with $CFL = 200$ and 2,000 (3 sub-iterations), pLU-SGS/A-Roe with $CFL = 200$ (3 sub-iterations), and pLU-SGS/AUSM$^+$-up with CFL=200 (3 sub-iterations) are in the fastest group. From those results, in terms of efficiency, pLU-SGS/P-Roe appeared to be the best with the maximum allowable CFL numbers, followed by pLU-SGS/AUSM$^+$-up. Based on this limited set of results, numerical dissipation in P-Roe and AUSM$^+$-up seemed to be compatible with those produced by LU-SGS (or pLU-SGS) for large CFL numbers[§], probably due to their use of M_∞.

[§] From Eqs. (2.34) and (2.37), the larger the CFL, the smaller the scalar D becomes, degrading its diagonal dominance and hence, introducing more numerical dissipation into the system of equations.

3.5 Effect of local time-stepping

All the computations above were conducted with global time stepping so that discussions therein could be applied to unsteady flow computations with the use of dual-time stepping, in which temporal convergence is attained in each time step [4] (not actually covered in this work, though). On the other hand, it is commonly known that the local time-stepping technique (Eq. (2.37)) accelerates convergence rate for steady flow computations. Thus, in case one is interested only in steady solutions, we employed local-time-stepping technique for the test cases here.

The computations were conducted for the same cases and methods as in the above discussions, and three sub-iterations were adopted. The results are summarized in Tables 10 ($CFL=20$) and 11 ($CFL=20-20{,}000$ for pLU-SGS/Low-dissipation-flux cases), and residual histories are shown in Figs. 12 and 13.

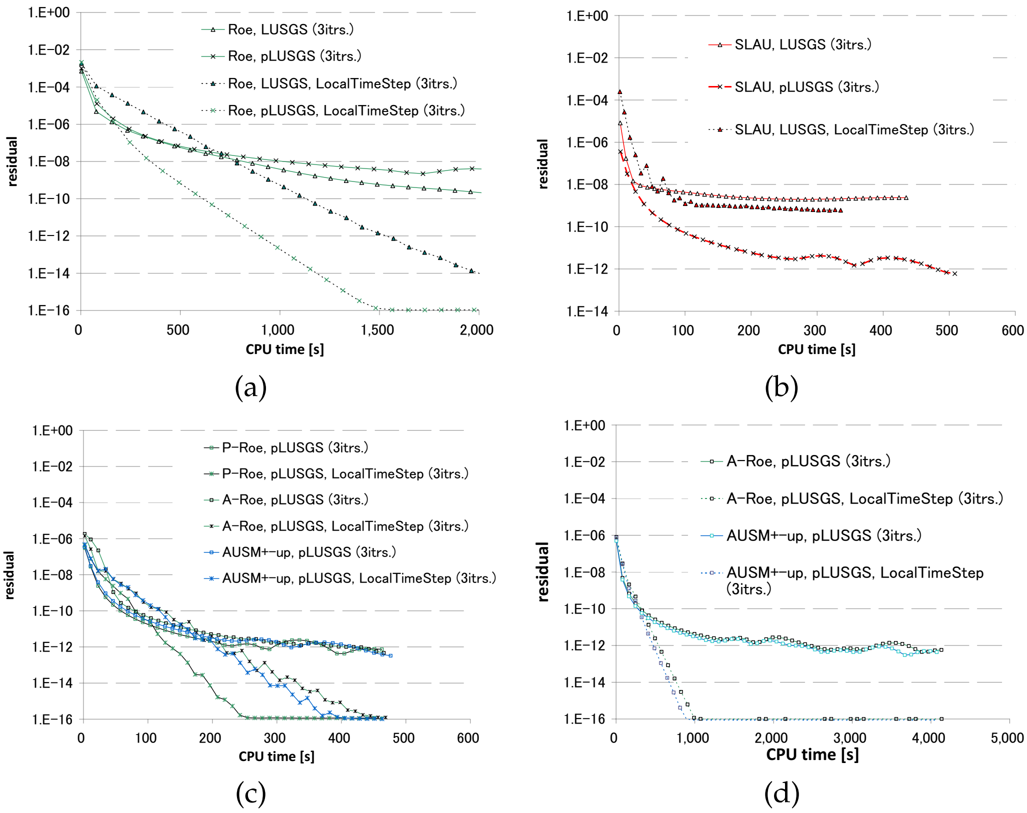
- The portions of successful cases and others are roughly similar to the global time-stepping cases shown in Table 5.
- However, SLAU seemed to be destabilized by employing the local time-stepping and showed no successful cases when coupled with pLU-SGS, although this flux is only one which showed relative robustness at low speeds when used with LU-SGS, again.
- pLU-SGS/P-Roe is the most robust against increasing CFL whether the local time-stepping is used or not.

Table 10: Computed results of three sub-iteration cases ($CFL=20$ with local time stepping), S (Successful), U (Unphysical or oscillatory), and F (Failure, diverged).

Sub-iterations: 3		Case 1	Case 2			Case 3
Implicit Schemes	Euler Fluxes	$M_\infty=0.5$ $Re_\infty=5{,}000[Sep.point(\%)]$	A) $M_\infty=0.1$ $Re_\infty=\infty$ (Inviscid)	B) $M_\infty=0.01$	C) $M_\infty=0.001$	$M_\infty=0.01$ $Re_\infty=2{,}000$
LU-SGS	Roe	**S**[84.9]	**U**			**U**
	AUSM$^+$	**S**[84.9]				
	SHUS	**S**[84.9]				
	P-Roe	**S**[83.8]	**F**	**F**	**F**	**F**
	A-Roe	**S**[82.7]	**S**	**F**	**F**	**F**
	AUSM$^+$-up	**S**[82.7]	**F**	**F**	**F**	**F**
	SLAU	**S**[82.7]	**S**	**S**	**S**	**U**
pLU-SGS	Roe	**S**[82.7]	**F**			**F**
	AUSM$^+$	**F**				
	SHUS	**U**[82.9]				
	P-Roe	**S**[82.7]	**S**	**S**	**S**	**S**
	A-Roe	**S**[82.7]	**S**	**S**	**S**	**S**
	AUSM$^+$-up	**F**	**F**	**S**	**S**	**S**
	SLAU	**U**[83.8]	**F**	**F**	**F**	**F**

Table 11: Summary of CFL Effect, S (Successful), U (Unphysical or oscillatory), and F (Failure, diverged).

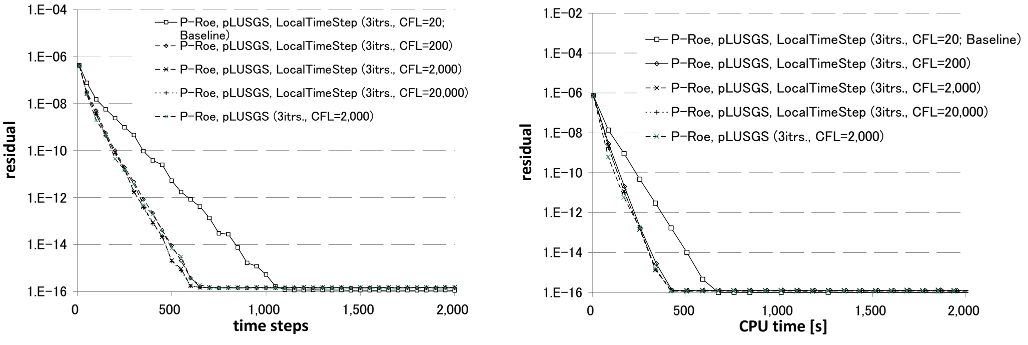
Sub-iterations: 3 *with local time stepping*		Case 2B) $M_\infty=0.01$, $Re_\infty=\infty$ (Inviscid) $[C_D]$				Case 3) $M_\infty=0.01$, $Re_\infty=2{,}000$			
Implicit Scheme	Euler Fluxes	$CFL=20$	200	2,000	20,000	$CFL=20$	200	2,000	20,000
pLU-SGS	P-Roe	**S** [0.0055]	**S** [0.0055]	**S** [0.0055]	**S** [0.0055]	**S**	**S**	**S**	**S**
	A-Roe	**S**[0.0019]	**F**	**F**	**F**	**S**	**S**	**S**	**S**
	AUSM$^+$-up	**S** [0.0049]	**F**	**F**	**F**	**S**	**F**	**F**	**F**
	SLAU	**F**	**F**	**F**	**F**	**F**	**F**	**F**	**F**



(a) (b)

(c) (d)

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



(a) (b)

Figure 13: Residual histories with different CFL numbers, with and without local time stepping, for $M_\infty=0.01$: (a) inviscid (Case 2B) and (b) viscous (Case 3) computations.

- As shown in Fig. 12(a), the local time-stepping clearly accelerated the convergence for viscous, moderate Mach number flow of $M_\infty=0.5$ (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. (2.30), (2.32), and (2.37). The spectral radius, $\sigma = |V_n| + c + (\text{viscous term})$, is dominated by $|V_n|$ and 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. 2(b)), significantly affecting the time step Δt_i, compared with preconditioning $\sigma \rightarrow \sigma'$, which leaves the order of the spectral radius remained.

- At low Mach numbers, on the other hand, the local time-stepping is less effective than time-derivative preconditioning (Fig. 12(b)-(d)). Again, this is clearly explained from Eqs. (2.30), (2.32), and (2.37). If preconditioning technique is used at low speeds, 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; on the contrary, if only the local time-stepping is used, this technique has little effect on the spectral radius $\sigma = |V_n| + c + (\text{viscous term})$, since $|V_n| \ll c$.
- Combination of the time-derivative preconditioning and the local time stepping is considered quite effective, but this set led to unstable or oscillatory solutions under some conditions (e.g., pLU-SGS/AUSM$^+$-up with $CFL = 200$ or more).
- According to Fig. 13, in which residual histories for cases with the time-derivative preconditioning and the local time stepping both are shown, convergence acceleration by increasing CFL is stagnated at $CFL = 200$, showing comparable convergence rate with $CFL = 2{,}000$ without local time-stepping, in either the inviscid (Case 2B) or viscous (Case 3) simulation for $M_\infty = 0.01$.

3.6 The "Best" schemes

According to the error estimation above ($M_\infty = 0.01$, inviscid case of Case 2B), pLU-SGS/A-Roe, followed by pLU-SGS/SLAU, produced the most *accurate* solution if it successfully worked.

In terms of *efficiency*, we counted minimum required CPU time for machine zero convergence for each method with its maximum allowable CFL (usually 2,000, except for pLU-SGS/SLAU with $CFL = 20$) in $M_\infty = 0.01$, viscous case of Case 3 (Fig. 10(b)). According to this criterion, pLU-SGS/P-Roe (357 sec.), followed by pLU-SGS/AUSM$^+$-up (473 sec.), appears to be the best.

To compare *robustness*, we simply counted numbers of successful cases marked in Tables 4 and 5 for global time-stepping cases: pLU-SGS/P-Roe (10), pLU-SGS/A-Roe (7), pLU-SGS/AUSM$^+$-up (6), pLU-SGS/SLAU (10). Thus, pLU-SGS/P-Roe and pLU-SGS/SLAU produced more successful cases than other methods in a range of $M_\infty = 0.001 - 0.5$, and these combinations seem to be the most robust among all the methods. In local time-stepping cases, on the other hand, pLU-SGS/SLAU failed and pLU-SGS/P-Roe and pLU-SGS/A-Roe are the best.

Based on all the discussions above, evaluation of each combination of preconditioned LU-SGS scheme and a low-dissipation flux is presented in Table 12. All in all, in low speed flow computations, each method is suggested for use in the following occasions:

- pLU-SGS/P-Roe: One seeks the fastest convergence, and M_{co} is available.
- pLU-SGS/A-Roe: Obtaining the most accurate solutions is the top priority.
- pLU-SGS/AUSM$^+$-up: One seeks fast convergence, and M_∞ is available.

Table 12: Evaluation of Preconditioned LU-SGS Scheme/Low-Dissipation Euler Fluxes.

		Accuracy (indicated by numerical errors, C_D, in $M_\infty = 0.01$, inviscid flow simulation)	Efficiency (indicated by minimum CPU time for machine zero convergence)	Robustness (numbers of successful cases)	
				global time step (10 cases)	local time step (5 cases)
pLU-SGS	P-Roe	0.0055	357 sec.	10	5
	A-Roe	0.0019	546 sec.	7	5
	AUSM$^+$-up	0.0049	473 sec.	6	3
	SLAU	0.0037	More than 4,000 sec.	10	0

- pLU-SGS/SLAU: One is not sure whether the computation reaches stable solutions, or there is no reference (uniform) flow present. In addition, if computational time really does not matter, LU-SGS/SLAU would be an alternative choice.

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 P-Roe, A-Roe or AUSM$^+$-up only when they are available.

4 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), divergence or slow convergence (control of dissipation in numerical flux only), or divergence of calculations (preconditioning of time integration only). It is also confirmed that the recommended lower limit of non-preconditioned methods for use be over $M_\infty = 0.1$. All in all, in low speed flow computations, pLU-SGS/P-Roe, pLU-SGS/A-Roe, pLU-SGS/SLAU or pLU-SGS/AUSM$^+$-up combination is suggested for use in the following occasions:

- pLU-SGS/P-Roe: One seeks the fastest convergence, and the cutoff Mach number M_{co} are available.
- pLU-SGS/A-Roe: Obtaining the most accurate solutions is the top priority.
- pLU-SGS/AUSM$^+$: One seeks fast convergence, and the freestream Mach number M_∞ is available.
- pLU-SGS/SLAU: One is not sure whether the computation reaches stable solutions, or there is no reference (uniform) flow present. Moreover, if computational time really does not matter, LU-SGS/SLAU would be an alternative choice.

In addition, SLAU is the only all-speed scheme which is totally free from restrictions of specifying reference values, such as M_{co} or 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, we would like to point out the following:

1. Viscous effects played a minor role.
2. The 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.

Appendix

The transformation matrix from primitive variables to conservative variables is written as

$$\frac{\partial \mathbf{Q}}{\partial \mathbf{q}} = \begin{pmatrix} \frac{\gamma}{c^2} & 0 & 0 & 0 & -\frac{\rho}{T} \\ \frac{\gamma}{c^2}u & \rho & 0 & 0 & -\frac{\rho}{T}u \\ \frac{\gamma}{c^2}v & 0 & \rho & 0 & -\frac{\rho}{T}v \\ \frac{\gamma}{c^2}w & 0 & 0 & \rho & -\frac{\rho}{T}w \\ \frac{\gamma}{c^2}H-1 & \rho u & \rho v & \rho w & -\frac{\rho}{T}\frac{q^2}{2} \end{pmatrix}, \qquad q^2 = u^2+v^2+w^2, \tag{A.1}$$

where $\mathbf{Q}$ is the conservative state vector $(\rho,\rho u,\rho v,\rho w,\rho E)^T$, and $\mathbf{q}$ is the primitive one employing pressure [14] $(\rho,u,v,w,T)^T$.

Then, the preconditioner of Weiss and Smith for conservative variables is written as follows due to Turkel [5], although this form is not used in the actual implementation.

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

$$\varepsilon = \min\left(1, \max(KM^2, M_{co}^2)\right), \tag{A.3}$$

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_∞.

As stated in the main text, Eq. (2.36) shown below is used instead and the computational cost for the implementation of $\mathbf{\Gamma}^{-1}\Delta\mathbf{F}$ is trivial according to Turkel [5]:

$$\mathbf{\Gamma}^{-1}\Delta\mathbf{F}=\Delta\mathbf{F}-\frac{(1-\varepsilon)dp}{c^2}(1\ u\ v\ w\ H)^T, \tag{A.4}$$

$$dp=(\gamma-1)\Big(\frac{u^2+v^2+w^2}{2}\Delta\mathbf{F}_1-u\Delta\mathbf{F}_2-v\Delta\mathbf{F}_3-w\Delta\mathbf{F}_4+\Delta\mathbf{F}_5\Big), \tag{A.5}$$

where $\Delta\mathbf{F}_l$ stands for the l-th row of $\Delta\mathbf{F}$ (e.g., $\Delta\mathbf{F}_1=\Delta(\rho V_n)$).

With the above expression one can easily obtain preconditioned variables by avoiding actual matrix operation of $\mathbf{\Gamma}^{-1}\Delta\mathbf{F}$.

Nomenclature

C_D	drag coefficient
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, $1.0e-3$ in viscous cases (based on the airfoil chord length of 1)
E	total energy
$\mathbf{F}$,$\mathbf{F}_v$	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,$\kappa=\mu c_p/Pr$,
M	Mach number
$\dot{m}$	mass flux, $\dot{m}=\rho u$,
μ	molecular viscosity
$\mathbf{n}$	normal vector to the cell-interface, $(n_x,n_y,n_z)^T$
P	pressure
Pr	Prandtl number, 0.72
$\mathbf{Q}$	(conservative) state vector

Re	Reynolds number (based on the airfoil chord length of 1)
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
	$V_n=(u,v,w),\quad \mathbf{n}=un_x+vn_y+wn_z$
$(^-)$	arithmetic averaged value
$\hat{()}$	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

Superscripts

m	value at m-th sub-iteration
n	value at n-th timestep

Acknowledgments

This work was conducted as a joint research between JAXA and Iowa State University (ISU). The authors are grateful to Prof. Kozo Fujii and other staff at JAXA and ISU both for the relevant support. The authors also thank Associate Prof. Nobuyuki Tsuboi at Kyushu Institute of Technology and Dr. Chris Nelson at Innovative Technology Applications Company for having constructive discussions with us. We also thank Dr. Kazuto Kuzuu at JEDI Center, JAXA, and Mr. Junya Aono at Research Center of Computational Mechanics, Inc., for improvement of our numerical code.

References

[1] C. C. Kiris, D. Kwak, W. Chan, and J. A. Housman, High-fidelity simulations of unsteady flow through tubopumps and flowliners, Comput. Fluids., 37 (2008), 536–546.

[2] N. Tani, S. Tsuda, and N. Yamanishi, Numerical study of cavitating inducer in cryogenic fluid, 49th Japan Conference on Propulsion and Power, B14 (2009) (in Japanese).

[3] N. Tsuboi, K. Fukiba, and T. Shimada, 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] J. M. Weiss, and W. A. Smith, Preconditioning applied to variable and constant density flows, AIAA J., 33(11) (1995), 2050–2057.

[5] E. Turkel, Preconditioning technique in computational fluid dynamics, Annu. Rev. Fluid. Mech., 31 (1999), 385–416.

[6] M. S. Liou, A sequel to AUSM, part II: AUSM$^+$-up for all speeds, J. Comput. Phys., 214 (2006), 137–170.

[7] J. R. Edwards, Towards unified CFD simulation of real fluid flows, AIAA Paper, 2001–2524, 2001.

[8] E. Shima, and K. Kitamura, On new simple low-dissipation scheme of AUSM-family for all speeds, AIAA Paper, 2009-0136, 2009.

[9] X. S. Li, and C. W. Gu, An all-speed Roe-type scheme and its asymptotic analysis of low mach number behavior, J. Comput. Phys., 227 (2008), 5144–5159.

[10] A. Jameson, and E. Turkel, Implicit schemes and LU decompositions, Math. Comput., 37 (1981), 385–397.

[11] S. Yamamoto, Preconditioning method for condensate fluid and solid coupling problems in general curvilinear coordinates, J. Comput. Phys., 207 (2005), 240–260.

[12] F. Xie, W. Song, and Z. Han, Numerical study of high-resolution scheme based on preconditioning method, J. Aircraft., 46(2) (2009), 520–525.

[13] D. Unrau, and D. W. Zingg, Viscous airfoil computations using local preconditioning, AIAA Paper, 97-2027, 1997.

[14] G. Hauke, and T. J. R. Hughes, A comparative study of different sets of variables for solving compressible and incompressible flows, Comput. Methods. Appl. Mech. Engrg., 153 (1998), 1–44.

[15] E. Shima, K. Kitamura, and K. Fujimoto, 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.

[16] D. J. Mavriplis, Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes, AIAA Paper, 2003-3986, 2003.

[17] Z. J. Wang, A quadtree-based adaptive cartesian/quad grid flow solver for Navier-Stokes equations, Comput. Fluids., 27(4) (1998), 529–549.

[18] K. Kitamura, K. Fujimoto, E. Shima, K. Kuzuu, and Z. J. Wang, Validation of arbitrary unstructured CFD code for aerodynamic analyses, transactions of the Japan society for aeronautical and space sciences, (accepted for publication).

[19] P. L. Roe, Approximate Riemann solvers, parameter vectors and difference schemes, J. Comput. Phys., 43 (1981), 357–372.

[20] M. S. Liou, A sequel to AUSM: AUSM$^+$, J. Comput. Phys., 129 (1996), 364–382.

[21] E. Shima, Role of CFD in aeronautical engineering (No. 14)-AUSM type upwind schemes-, NAL-SP30, Proceedings of 13th NAL symposium on Aircraft Computational Aerodynamics, 41–46, 1996.

[22] Y. Liu, and M. Vinokur, Upwind algorithms for general thermo-chemical nonequilibrium flows, AIAA Paper, 89–0201, 1989.

[23] K. M. Peery, and S. T. Imlay, Blunt-body flow simulations, AIAA Paper, 88-2904, 1988.

[24] K. Kitamura, P. Roe, and F. Ismail, Evaluation of Euler fluxes for hypersonic flow computations, AIAA Journal, 47(1) (2009), 44–53.

[25] R. Nichols, R. Tramel, and P. Buning, Solver and turbulence model upgrades to OVERFLOW 2 for unsteady and high-speed applications, AIAA-2006-2824, 2006.

[26] Y. Sun, Z. J. Wang, and Y. Liu, Efficient implicit non-linear LU-SGS approach for compressible flow computation using high-order spectral difference method, Commun. Comput. Phys., 5 (2009), 760–778.

[27] D. J. Mavriplis, A. Jameson, and L. Martinelli, Multigrid solution of the Navier-Stokes equations on triangular meshes, AIAA Paper, 89-0120, 1989.

[28] V. Venkatakrishnan, Convergence to steady state solutions of the Euler equations on unstructured grids with limiters, J. Comput. Phys., 118 (1995), 120–130.

[29] Z. J. Wang, A fast nested multi-grid viscous flow solver for adaptive cartesian/quad grids, Int. J. Numer. Meth. Fluids., 33 (2000), 657–680.

[30] M. S. Liou, and J. R. Edwards, Numerical speed of sound and its application to schemes of all speeds, NASA TM-1999-09286, 1999; AIAA Paper, 99-3268-CP, 1999.