Seventh International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, Hawaii, July 9-13, 2012

ICCFD7-3106

New Approaches for Efficient Computation of Low Mach Number Unsteady Flows with Sound Propagation

E. Shima* and K. Kitamura*
Corresponding author: shima.eiji@jaxa.jp

* Japan Aerospace Exploration Agency, JAPAN

Abstract: By using all speed numerical flux schemes, such as SLAU [Simple Low Dissipation AUSM (Advection Upstream Splitting Method)], in MUSCL (Monotone Upwind Scheme for Conservation Laws) approach for compressible CFD, low Mach number flows can be computed without loss of accuracy nor parameter tuning. For an efficient computation, this paper deals with a new approach of implicit time integration method. In this approach, the large sparse matrix system, which consists of flux Jacobian of numerical flux function, has to be solved in each time step. In this study, we tried to use FGMRES(k) (Flexible Generalized Minimum Residual Method) to solve the non-diagonal dominant linear system arising from Jacobian of flux function SLAU.

Keywords: Numerical Algorithms, Computational Fluid Dynamics, Aeroacoustics, Preconditioning, Implicit Time Integration, SLAU

Nomenclature

c	= sound speed
e	= total energy per unit volume
$\hat{\mathbf{E}}, \hat{\mathbf{R}}$	=Inviscid and viscous flux outward normal to face
$\tilde{\mathbf{E}}, \tilde{\mathbf{R}}$	=Numerical inviscid and viscous flux outward normal to cell face
h	= total enthalpy
L	= reference length scale
l	= wave length
M	= Mach number
$\mathbf{M}$	= Transforming matrix from conservative to entropy variable
p	= pressure
$\mathbf{P}$	= preconditioning matrix for linear system
$\mathbf{Q}$	= vector of conservative variables, $(\rho, \rho u, \rho v, \rho w, e)^T$
Re	= Reynolds number
s	= area of cell-interface
$\mathbf{S}$	= vector of stored field variable
T	= period of wave
t	= physical time
u, v, w	= Cartesian velocity components
V_n	= velocity component normal to cell-interface
V	= volume of cell
$\mathbf{W}$	= vector of working variables
x_n, y_n, z_n	= outward normal of cell-interface
x, y, z	= Cartesian coordinates

τ = pseudo time
ρ = density
μ = molecular viscosity
μ_T = turbulent viscosity
σ = spectral radius

Subscript
∞ =freestream value

1 Introduction

When compressible CFD methods are applied to low Mach number flows, cares must be taken to excessive numerical dissipation and stiffness due to the large condition number, which is the ratio of maximum and minimum characteristic speeds. Round off error due to very small changes of scalar variables may be another problem, but this can be rather easily avoided by separately storing variables of their reference and variation values as $p=p_\infty+p'$.

The authors proposed all speed numerical flux schemes of AUSM (Advection Upstream Splitting Method) family named SLAU [1] (Simple Low-dissipation AUSM) and showed this scheme can compute from very low to very high Mach number without tuning of parameters, such as the cutoff Mach number. It was also shown that the combination with Weiss-Smith[2] time derivative pre-conditioning is effective and the stiffness can be avoided at least for steady state flows.

SLAU is a compressible flow CFD algorithm that can compute very low Mach number flows, thus, they can be a good choice for the direct solver of aero-acoustic problems in low speed flow, i.e., for solving flow and sound at the same time.

Incompressible flow is thought as low Mach number limit. In incompressible CFD methods pressure wave is neglected, thus acoustics must be treated separately from flow dynamics. If potential methods are used for acoustics, for example, effects of non-uniform velocity field cannot be included easily. As another option, the effect of flow can be reflected by the use of LEE (linearized Euler equation), but still it is difficult when the average flow is hard to be set up. And also the computational cost for LEE is roughly the same as for Euler or laminar Navier-Stokes equation, so the benefit of LEE is not so significant. These are the reasons why we chose the direct solver approach here.

By using explicit time integration, it has been proven in previous research that sound propagation can be computed. However, the usage of explicit schemes is impractical for low Mach number flows since time step determined by sound speed is too restrictive for convection.

A Lager time step can be used with implicit time integration. In implicit method, a large sparse matrix system, which consists of flux Jacobian of numerical flux function, has to be solved in each time step [1, 2]. In our previous work, we introduced Time-Consistent Preconditioned Gauss-Seidel (TC-PGS) [3], a version of preconditioned Gauss-Seidel (GS)[1] implicit time integrations using entropy variables, and demonstrated its accuracy and efficiency over a conventional GS method in solving flow dynamics and aero-acoustics both in low speed flows (see Fig. 1).

Through the derivation of TC-PGS, we realized that the essential point of the preconditioned implicit algorithm is to design the implicit numerical dissipation which is matched to the R.H.S with keeping the positive definiteness of the numerical flux Jacobian, which is a necessary and sufficient condition of the diagonal dominance of the linear system. In this study, SLAU is applied as the R.H.S. numerical flux function, thus the apparent choice is to use the implicit dissipation close to that of SLAU. However, the linear system for approximate Jacobian of SLAU turned out to be non-diagonal dominant, as will be explained later in this paper. Traditional iterative linear solvers such as a Gauss-Seidel method are

[1] The term GS (Gauss-Seidel) is referred to as a GS scheme or a single sweep within a pair of symmetric sweeps in SGS (Symmetric Gauss-Seidel) in this paper.

unstable without the diagonal dominance. Since only an approximate linear solution is required for this purpose, one choice is nevertheless to use diagonal dominant approximation on linear system. This leads to a simpler method named TC-PGS1, which has less dependence on a user-specified parameter, and this will be introduced in this paper first. Then, as another approach, we will solve the non-diagonal dominant system directly by a more sophisticated method. In this study, FGMRES(k) by Saad [4] is employed in which we can use different matrix preconditioners at each GMRES step: The SGS matrix borrowed from TC-PGS1 is used as a matrix preconditioner, along with different numbers of iterations at each step.

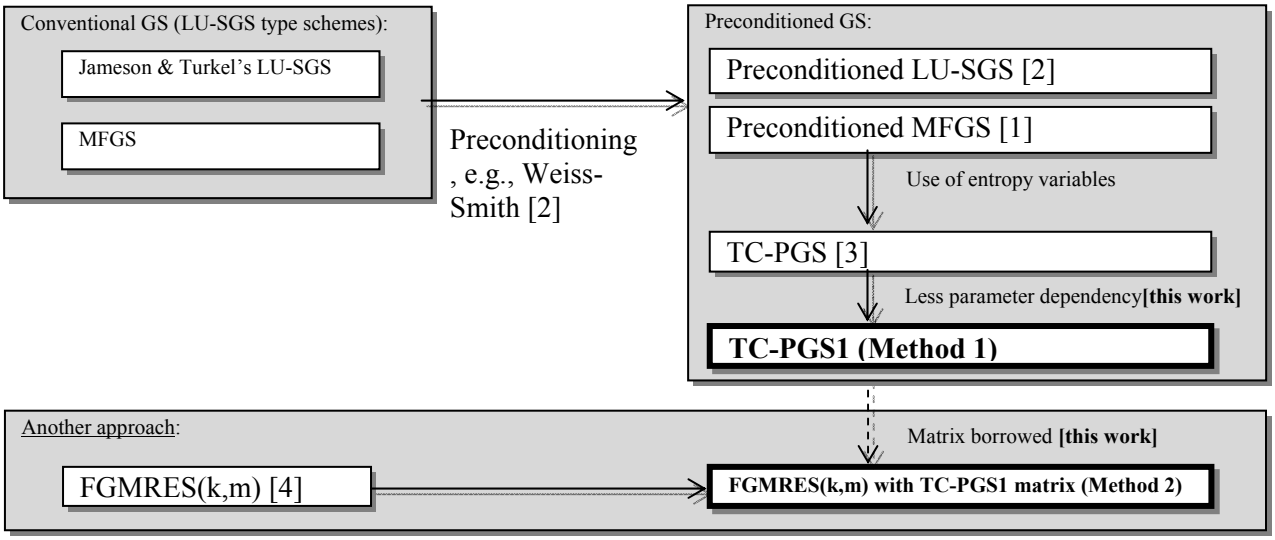


Figure 1. Time integration methods.

2 Governing equation and basic numerical scheme

Compressible Navier-Stokes equation is written in integral form as;

$$\iiint \mathbf{Q}_t dv + \oiint (\hat{\mathbf{E}} - \hat{\mathbf{R}}) ds = 0 \tag{2.1}$$

By using polyhedrons (polygons in 2 dimensions) as control volumes, the basic equation for FVM (Finite Volume Method) is written as;

$$\frac{1}{\Delta t}\Delta \mathbf{Q}_i + \frac{1}{V_i}\sum_j (\tilde{\mathbf{E}}_{i,j} - \tilde{\mathbf{R}}_{i,j}) s_{i,j} = 0 \tag{2.2}$$

$$\Delta \mathbf{Q}_i = \mathbf{Q}^{n+1}{}_i - \mathbf{Q}^{n}{}_i \tag{2.3}$$

Here subscript *i,j* means '*j*'th face of the cell '*i*' , *n* and *n*+1 are physical time steps. Our expression here is based on an unstructured grid formulation, but structured grids can be treated as only a special case.

For unsteady computations, the dual time stepping and 3-point backward Euler scheme are introduced;

$$\frac{\mathbf{Q}_i^{k+1} - \mathbf{Q}_i^{k}}{\Delta \tau} + \{\frac{\theta_1 \mathbf{Q}_i^{k+1} - \theta_2 \mathbf{Q}_i^{n} - (\theta_1 - \theta_2)\mathbf{Q}_i^{n-1}}{\Delta t} + \frac{1}{V_i}\sum_j (\tilde{\mathbf{E}}_{i,j}{}^{k+1} - \tilde{\mathbf{R}}_{i,j}{}^{k+1}) s_{i,j}\} = 0 \tag{2.4}$$

Here, *k* and *k*+1 denote pseudo time steps. For the second order temporal accuracy with time step variation, coefficients θ_k are given b;

$$(\theta_1, \theta_2) = \left((r+2)/(r+1), (r+1)/r\right)$$
$$\Delta t^{n-1} = r\Delta t^{n} \tag{2.5}$$

where *r*=1 in this study as usual; for the first order method, they are given by;

$$(\theta_1, \theta_2) = (1,1) \tag{2.6}$$

The pseudo time τ can be chosen independently from physical time t, and correct time evolution is recovered for any choice of τ when the equation is converged about τ. If non-factored implicit schemes, such as Gauss-Seidel iteration or FGMRES in this paper, are used, faster convergence is obtained by bigger $\Delta\tau$. Thus we take $\Delta\tau$ to be infinitely large, and then, obtained the following equation;

$$\frac{\theta_1 \mathbf{Q}_i^{\,k+1} - \theta_2 \mathbf{Q}_i^{\,n} - (\theta_1 - \theta_2)\mathbf{Q}_i^{\,n-1}}{\Delta t} + \frac{1}{V_i}\sum_j (\tilde{\mathbf{E}}_{i,j}^{\,k+1} - \tilde{\mathbf{R}}_{i,j}^{\,k+1}) s_{i,j} = 0 \tag{2.7}$$

3 Implicit time integration algorithm in delta form

Using first order upwind difference and approximate linearization for L.H.S., implicit time integration scheme of Eq.(2.7) is written as;

$$\left[\frac{\theta_1}{\Delta t} + \frac{1}{V_i}\sum_j s_{i,j}\tilde{\mathbf{A}}_{i,j}^{\,+}\right]\Delta\mathbf{Q}_i - \frac{1}{V_i}\sum_j s_{i,j}\tilde{\mathbf{A}}_{j,i}^{\,+}\Delta\mathbf{Q}_j = -\mathbf{H}_i^{\,k} \tag{3.1}$$

$$\mathbf{H}_i^{\,k} = \frac{\theta_1 \mathbf{Q}_i^{\,k} - \theta_2 \mathbf{Q}_i^{\,n} - (\theta_1 - \theta_2)\mathbf{Q}_i^{\,n-1}}{\Delta t} + \frac{1}{V_i}\sum_j (\tilde{\mathbf{E}}_{i,j}^{\,k} - \tilde{\mathbf{R}}_{i,j}^{\,k}) s_{i,j} \tag{3.2}$$

$$\tilde{\mathbf{A}} = \partial\tilde{\mathbf{E}}/\partial\mathbf{Q} - \partial\tilde{\mathbf{R}}/\partial\mathbf{Q} \tag{3.3}$$

$$\Delta\mathbf{Q}_i = \mathbf{Q}_i^{\,k+1} - \mathbf{Q}_i^{\,k} \tag{3.4}$$

$\tilde{\mathbf{A}}_{i,j}^{\,+}$ is the Jacobian matrix of flux at cell interface. The flux Jacobian and dissipation matrix can be simplified by introducing the entropy variable vector which is defined as follows;

$$\partial\mathbf{W} = \left(\partial p, \partial u, \partial v, \partial w, \partial p - c^2\partial\rho\right)^T \tag{3.5}$$

By mapping this variable vector on Eq.(3.1), an implicit method using the entropy variable is written as;

$$\left[\frac{\theta_1\mathbf{I}}{\Delta t} + \frac{1}{V_i}\sum_j s_{i,j}\left(\frac{\mathbf{\Delta}'}{2} + \tilde{\mu}\mathbf{I}\right)_{i,j}\right]\Delta\mathbf{W}_i - \frac{1}{V_i}\sum_j s_{i,j}\left(\frac{\hat{\mathbf{B}}_{j,i} + \mathbf{\Delta}'}{2} + \tilde{\mu}\mathbf{I}\right)\Delta\mathbf{W}_j = -\mathbf{M}_i\mathbf{H}_i^{\,k} \tag{3.6}$$

Here $\tilde{\mu}$ is a viscous component of the spectral radius defined by;

$$\tilde{\mu} \equiv \frac{(\mu + \mu_T)s_{i,j}}{\rho_i V_i} \tag{3.7}$$

where $\mathbf{\Delta}'/2$ is the implicit dissipation matrix stated later. $\hat{\mathbf{B}}$ is the unmodified , i.e. not upwind, flux Jacobian of Euler equation in entropy variable defined as;

$$\hat{\mathbf{B}} = \begin{pmatrix} V_n & \rho c^2 x_n & \rho c^2 y_n & \rho c^2 z_n & 0 \\ \dfrac{x_n}{\rho} & V_n & 0 & 0 & 0 \\ \dfrac{y_n}{\rho} & 0 & V_n & 0 & 0 \\ \dfrac{z_n}{\rho} & 0 & 0 & V_n & 0 \\ 0 & 0 & 0 & 0 & V_n \end{pmatrix} \tag{3.8}$$

$$V_n = ux_n + vy_n + wz_n \tag{3.9}$$

M is the transformation matrix between conservative and entropy variable defined by;

$$\Delta\mathbf{W} = \mathbf{M}\Delta\mathbf{Q} \tag{3.10}$$

Finally, the solution procedure for Eq.(3.7) is summarized as follows. Note that conservation is recovered if non-linear Newton iteration (2.) is converged. It also indicates that the exact solution of a large linear system Eq.(3.6) is not necessary, because Newton iteration approximates the exact solution even if it is obtained. Therefore, some compromise between accuracy and computational efficiency is possible. In the following sections, we will give detailed explanations of solution procedure of Eq.(3.6) in step 2.2.

1. At n–th physical time step, set k:=0, $\mathbf{Q}^k$:=$\mathbf{Q}^n$
2. Repeat the following Newton iteration until prescribed convergence level or iteration count is achieved;
   2.1. Compute RHS=$\mathbf{M}_i^{-1}\mathbf{H}_i^k$ using $\mathbf{Q}^k$
   2.2. Solve Eq.(3.6) approximately to obtain $\Delta\mathbf{W}$
   2.3. Compute $\mathbf{Q}^{k+1}$:=$\mathbf{Q}^k$+$\mathbf{M}^{-1}\Delta\mathbf{W}$
   2.4. Set k=k+1
3. Next Time Step, $\mathbf{Q}^{n+1}$:=$\mathbf{Q}^{k+1}$, n:=n+1

4 Definition of the dissipation matrix

Considering computational efficiency, the implicit dissipation matrix is assumed to be diagonal which is defined by;

$$\mathbf{\Delta}' = diag\left(\Delta_p, \Delta_u, \Delta_u, \Delta_u, \Delta_s\right) \tag{4.1}$$

The numerical dissipation corresponding to SLAU (see Appendix. A) is approximately given by;

$$\Delta_p = |V_n| + c \tag{4.2}$$

$$\Delta_u = \Delta_s = |V_n| + (1-\chi)c \tag{4.3}$$

$$\chi = (1-\widehat{M})^2 \tag{4.4}$$

$$\widehat{M} = \min\left(1.0, \sqrt{\frac{u^2+v^2+w^2}{c^2}}\right) \tag{4.5}$$

On the other hand, for the dissipation matrix to satisfy semi-positivedefiniteness of flux Jacobian $\left(\pm\widetilde{\mathbf{B}}+\mathbf{\Delta}'\right)$ (which is a necessary and sufficient condition for diagonal dominance), the following formulation is proposed for the dissipation coefficients;

$$\Delta_p = |V_n| + \frac{c}{\alpha} \tag{4.8}$$

$$\Delta_u = \Delta_S = |V_n| + \alpha c \tag{4.9}$$

$$\alpha > 0$$

Apparently the spectral radius of the Euler flux defined as follows satisfies the diagonal dominant condition;

$$\Delta_p = \Delta_u = \Delta_S = |V_n| + c \tag{4.10}$$

It is also indicated that the matrix system consisting of the dissipation matrix given by SLAU [Eqs.(4.2-5)] is not diagonal dominant unless χ=0. Therefore, the matrix system with dissipation matrix of Eqs.(4.2-5) cannot be solved by a traditional iterative method such as SGS (Symmetric Gauss-Seidel) method. On the other hand, the implicit dissipation coefficients should be close to that of R.H.S. for faster convergence of the Newton iteration(2.). Note that the dissipation of Eq.(4.10) is much larger than that of Eq.(4.3), when low Mach number flows (M<<1) is of interest. Also let us remind the readers that, only approximate solution of the linear system is required, thus the linear system with the dissipation Eqs.(4.8-9) is acceptable as far as the Newton iteration (2.) is converged. Therefore, we have the following two choices for the linear system solver as;

(Method 1): Use approximate dissipation defined by Eqs.(4.8-4.9) and simple iterative method such as G-S.

(Method 2): Use more accurate dissipation defined by Eqs.(4.2-5) with a better linear solver. Both options will be considered in the next section. Relations of those methods and other prior developments are summarized in Fig. 1.

5 Solution Method 1: TC-PGS1 - Approximate dissipation and use of SGS

5.1 Formulation

With keeping low dissipation for velocity equation of Eq.(4.3), the diagonal dominant dissipation is derived as;

$$\Delta_p = |V_n| + c/\alpha \tag{5.1}$$

$$\Delta_u = \Delta_s = |V_n| + \alpha c \tag{5.2}$$

$$\alpha = 1-(1-\hat{M}')^2 \tag{5.3}$$

$$\hat{M}' = \min\left(1.0, \sqrt{\frac{u^2+v^2+w^2}{\bar{c}^2} + Mc^2}\right) \tag{5.4}$$

Here Mc (>0) is the cutoff Mach number to avoid zero division and zero dissipation at the stagnation, i.e. u=v=w=0. Note that this dissipation is closely related to the low Mach number preconditioning method of Weiss & Smith. (Also Note that "low Mach number preconditioning" is different idea from "matrix preconditioning" stated later.) The dissipation of Eq(5.2) is close to Eq.(4.3) as far as scale of Mc is the convective Mach number. It has been found from the authors' experience that Mc=M_∞ gives best convergence for wide range of flows. Also note that, the dissipation of Eq.(4.10) is recovered if Mc=1 is chosen. One G-S sweep can be written as;

$$\Delta\mathbf{W}_i = \left[\frac{\theta_1\mathbf{I}}{\Delta t} + \frac{1}{V_i}\sum_j s_{i,j}\left(\frac{\mathbf{\Delta}'}{2} + \tilde{\mu}\mathbf{I}\right)_{i,j}\right]^{-1}\left[\frac{1}{V_i}\sum_j s_{i,j}\left(\frac{\hat{\mathbf{B}}_{j,i}+\mathbf{\Delta}'}{2} + \tilde{\mu}\mathbf{I}\right)\Delta\mathbf{W}_j - \mathbf{M}_i\mathbf{H}_i^{\,k}\right] \tag{5.5}$$

As ***Δ'*** is a diagonal matrix, no block matrix operation is necessary. This procedure is summarized as follows;

2.2 Linear solver using SGS
  2.2.1 Set$\Delta\mathbf{W_i}$:=0
  2.22 Repeat until prescribed convergence level or iteration count;
    2.2.2.1 Forward sweep of Eq.(5.5)
    2.2.2.1 Backward sweep of Eq.(5.5)

It is also found that approximately 10 symmetric sweep is best to reduce overall computational time. This method is named as TC-PGS1 (Time Consistent Pre-conditioned Gauss-Seidel 1), since this has close relation to the time derivative pre-conditioning matrix method as will be mentioned in the next subsection. TC-PGS1 can be written in a LU-SGS type form (See Appendix.D), if only one pair of symmetric sweep is applied, although it is not the best in overall efficiency.

5.2 Correspondence of TC-PGS1 to time derivative preconditioning matrix method (TC-PGS)

Implicit time integration scheme with time derivative preconditioning matrix leads following system of linear equation. (see Appendix B)

$$\left[\frac{\theta_1\mathbf{\Gamma}_i}{\Delta t} + \frac{1}{V_i}\sum_j s_{i,j}\left(\mathbf{\Gamma}_i\tilde{\mathbf{A}}_{i,j}\right)^+\right]\Delta\mathbf{Q}_i - \frac{1}{V_i}\sum_j s_{i,j}\left(\mathbf{\Gamma}_i\tilde{\mathbf{A}}_{j,i}\right)^+\Delta\mathbf{Q}_j = -\mathbf{\Gamma}_i\mathbf{H}_i^{\,k} \tag{5.6}$$

Hear $\mathbf{\Gamma}$ is the time derivative preconditioning matrix. And $\left(\mathbf{\Gamma}_i\tilde{\mathbf{A}}_{i,j}\right)^+$ is the Jacobian matrix of preconditioned flux at cell interface that has only positive eigenvalues in direction from "i" to "j". The preconditioned upwind flux Jacobian including viscous term is formulated as follows, using the spectral radius σ following the LU-SGS of Jameson & Turkel;

$$\left(\mathbf{\Gamma}_i\tilde{\mathbf{A}}_{i,j}\right)^+ = \frac{\mathbf{\Gamma}_i\hat{\mathbf{A}}_{i,j} + \tilde{\sigma}_{i,j}\mathbf{I}}{2} \tag{5.7}$$

$$\sigma_{i,j} = \sigma\left(\mathbf{\Gamma}_i\tilde{\mathbf{A}}_{i,j}\right) \tag{5.8}$$

$$\tilde{\sigma}_{i,j} \equiv \sigma_{i,j} + 2\tilde{\mu} \tag{5.9}$$

By using these definitions, Eq.(3.1) can be rewritten as;

$$\left[\frac{\theta_1}{\Delta t}\mathbf{I} + \frac{\mathbf{\Gamma}_i^{-1}}{V_i}\sum_j s_{i,j}\frac{\tilde{\sigma}_{i,j}}{2}\right]\Delta\mathbf{Q}_i - \frac{1}{V_i}\sum_j s_{i,j}\frac{\hat{\mathbf{A}}_{j,i} + \tilde{\sigma}_{j,i}\mathbf{\Gamma}_i^{-1}}{2}\Delta\mathbf{Q}_j = -\mathbf{H}_i^{\,k} \tag{5.10}$$

This G-S iteration can be rewritten using the working variable as;

$$\Delta\mathbf{W}_i = \left[\frac{\theta_1}{\Delta t}\mathbf{I} + \frac{1}{V_i}\sum_j s_{i,j}\frac{\tilde{\sigma}_{i,j}\hat{\mathbf{\Gamma}}_i^{-1}}{2}\right]^{-1}\left[\frac{1}{V_i}\sum_j s_{i,j}\frac{\hat{\mathbf{B}}_{j,i} + \tilde{\sigma}_{j,i}\hat{\mathbf{\Gamma}}_i^{-1}}{2}\Delta\mathbf{W}_j - \mathbf{M}_i\mathbf{H}_i^{\,k}\right] \tag{5.11}$$

This is what we called original "TC-PGS" in our previous work. Comparing Eq.(5.5) and Eq.(5.11), we can see difference between both schemes just in choice of implicit dissipation matrix. If we use the preconditioning matrix of Weiss&Smith[2] which is a variation of Turkel's matrix, the preconditioning matrix and the spectral radius σ are given by;

$$\widehat{\mathbf{\Gamma}} = diag(\varepsilon,1,1,1,1) \tag{5.12}$$

$$\sigma_{i,j} = \frac{1}{2}\left\{(1+\varepsilon)\,|V_{ni,j}| + \sqrt{(\varepsilon-1)^2 V_{ni,j}{}^2 + 4\varepsilon c_i^2}\right\} \tag{5.13}$$

$$\varepsilon = \min(1, \max(M^2, M_{cutoff}{}^2)) \tag{5.14}$$

With this formulation, the dissipation matrix is written as;

$$\frac{\widetilde{\sigma}\widehat{\mathbf{\Gamma}}^{-1}}{2} = diag\left(\frac{\sigma+2\widetilde{\mu}}{2\varepsilon}, \frac{\sigma+2\widetilde{\mu}}{2}, \frac{\sigma+2\widetilde{\mu}}{2}, \frac{\sigma+2\widetilde{\mu}}{2}, \frac{\sigma+2\widetilde{\mu}}{2}\right) \tag{5.15}$$

On the other hand, the numerical dissipation of TC-PGS1 is written as;

$$\frac{\mathbf{\Delta}'}{2} + \widetilde{\mu}\mathbf{I} = diag\left(\frac{\Delta_p}{2}+\widetilde{\mu}, \frac{\Delta_u}{2}+\widetilde{\mu}, \frac{\Delta_u}{2}+\widetilde{\mu}, \frac{\Delta_u}{2}+\widetilde{\mu}, \frac{\Delta_u}{2}+\widetilde{\mu}\right) \tag{5.16}$$

$$\Delta_p = |V_n| + c/\alpha \tag{5.17}$$

$$\Delta_u = |V_n| + \alpha c \tag{5.18}$$

Comparing Eq.(5.15) with Eq.(5.16), two major differences are found;

(1) Viscous component in the pressure equation was magnified by 1/ε in the preconditioning approach (TC-PGS). It is natural consequence of the time derivative preconditioning which gave effects on all the components. As a consequence, the viscous term became too large in low Reynolds number flows, and it led to slow convergence, as shown in Appendix C.

(2) Inviscid dissipations also look different, however, it can be shown that these components in Eq.(5.15) and Eq.(5.16) become almost equivalent, if the following condition is satisfied;

$$\varepsilon = \frac{\alpha^2 + \alpha|V_n|/c}{1+\alpha|V_n|/c} \tag{5.19}$$

Since $\alpha \approx M$ under definition of Eqs.(5.3,4) at low Mach number and ε have a scale of M^2, it is concluded that both approaches are very similar. However, TC-PGS1 cannot be expressed in the framework of the preconditioning approach perfectly, since the ε given by Eq.(5.19) differs face by face. The benefits of the definition of inviscid component of Eq.(5.16) over Eq.(5.15) are i) simplicity of definition and ii) less dependency on cutoff Mach number for optimum convergence efficiency: Our preliminary numerical tests showed that the recommended Mc would be $Mc=M_\infty$ for general use of TC-PGS1.

5.3 Variations on working variables

Entropy variables are used as working variables to lead TC-PGS1 since the flux Jacobian of Eq.(3.8) becomes simplest in this variable space, however, other variables can be used as well. The essential point is to keep pressure and velocities in working variables. By keeping this, exactly same solution algorithm and a set of implicit numerical dissipation defined by Eq.(5.1-4) can be used. Two examples are shown bellow for variable sets and flux Jacobian in that variable space.

(Example.1) Viscous variables

$$\mathbf{W} = (p,u,v,w,CpT)^T \tag{5.20}$$

$$CpT = \frac{e+p}{\rho} - \frac{u^2+v^2+w^2}{2} \tag{5.21}$$

$$\hat{\mathbf{B}}=\begin{pmatrix} V_n & \rho c^2 x_n & \rho c^2 y_n & \rho c^2 z_n & 0 \\ \dfrac{x_n}{\rho} & V_n & 0 & 0 & 0 \\ \dfrac{y_n}{\rho} & 0 & V_n & 0 & 0 \\ \dfrac{z_n}{\rho} & 0 & 0 & V_n & 0 \\ 0 & c^2 x_n & c^2 y_n & c^2 z_n & V_n \end{pmatrix} \tag{5.22}$$

where *Cp* and *T* denote the specific heat at constant pressure and temperature.

(Example.2) Primitive variables

$$\mathbf{W}=(p,u,v,w,\rho)^T \tag{5.23}$$

$$\hat{\mathbf{B}}=\begin{pmatrix} V_n & \rho c^2 x_n & \rho c^2 y_n & \rho c^2 z_n & 0 \\ \dfrac{x_n}{\rho} & V_n & 0 & 0 & 0 \\ \dfrac{y_n}{\rho} & 0 & V_n & 0 & 0 \\ \dfrac{z_n}{\rho} & 0 & 0 & V_n & 0 \\ 0 & \rho x_n & \rho y_n & \rho z_n & V_n \end{pmatrix} \tag{5.24}$$

If these variables are used instead of the entropy variables and exact conservation in unsteady computation is omitted, the following simplified form, where variable transformation between the conservative and the working variables in every time step is not needed, can be used.

$$\left[\frac{\theta_1\mathbf{I}}{\Delta t}+\frac{1}{V_i}\sum_j s_{i,j}\left(\frac{\mathbf{\Delta}'}{2}+\tilde{\mu}\mathbf{I}\right)_{i,j}\right]\Delta\mathbf{W}_i-\frac{1}{V_i}\sum_j s_{i,j}\left(\frac{\widehat{\mathbf{B}}_{j,i}+\mathbf{\Delta}'}{2}+\tilde{\mu}\mathbf{I}\right)\Delta\mathbf{W}_j=-\mathbf{H}'_i{}^k \tag{5.25}$$

$$\mathbf{H}'_i{}^k=\frac{\theta_1\mathbf{W}_i^k-\theta_2\mathbf{W}_i^n-(\theta_1-\theta_2)\mathbf{W}_i^{n-1}}{\Delta t}+\frac{\mathbf{M}_i}{V_i}\sum_j(\tilde{\mathbf{E}}_{i,j}{}^k-\tilde{\mathbf{R}}_{i,j}{}^k)s_{i,j} \tag{5.26}$$

$$\mathbf{M}_i=\left(\frac{\partial\mathbf{W}}{\partial\mathbf{Q}}\right)_i \tag{5.27}$$

The conservative variables are only used in evaluation of flux balance and rest of whole computation is done in the working variables in this form. It corresponds to variable transformation often used in the time derivative preconditioning methods. It indicate that TC-PGS1 can be easily applied in computer code based the primitive or viscous variables.

6 Solution Method2: More accurate dissipation and use of FGMRES

We apply FGMRES(k) (Flexible Generalized Minimum Residual method) of Saad [4] to solve non-diagonal dominant matrix system. In order to avoid zero dissipation at the stagnation, the following dissipation is used;

$$\Delta_p=|V_n|+c \tag{6.1}$$

$$\Delta_u=\Delta_s=|V_n|+(1-\chi')c \tag{6.2}$$

As mentioned before, merely approximate linear solution is required. Thus only several times of restarts are used for FGMRES(k). Here, FGMRES(k) with m times restarts is denoted as FGMRES(k,m).This procedure is summarized as follows;

2.2 Linear solver using FGMRES(k,m)

2.2.1 Define $b \equiv [-\mathbf{M}_i \mathbf{H}_i]$, $x \equiv [\Delta \mathbf{W}_i]$

$$Ax \equiv \left[\left\{ \frac{\theta_l \mathbf{I}}{\Delta t} + \frac{1}{V_i} \sum_j s_{i,j} \left(\frac{\mathbf{\Lambda}'}{2} + \tilde{\mu} \right)_{i,j} \right\} \Delta \mathbf{W}_i - \frac{1}{V_i} \sum_j s_{i,j} \left(\hat{\mathbf{B}}_{j,i} + \frac{\mathbf{\Lambda}'}{2} + \tilde{\mu} \mathbf{I} \right) \Delta \mathbf{W}_j \right]$$

2.2.2 $\boldsymbol{x_0}:=0$

2.2.3 For l=1,…,m do

2.2.3.1 $r_0=b-Ax_0$, $\beta:=\| r_0 \|_2$ $v_1 := r_0 / \beta$

2.2.3.2 For j=1,…,k do;

- $z_j := \boldsymbol{P}_j^{-1} v_j$
- $w:=Az_j$
- For i=1,…j do $\{h_{i,j}:=(w,v_i)$, $w:=w-h_{i,j}v_i\}$
- $h_{j+1,j}=\| w \|_2$, $v_{j+1}=w/h_{j+1,j}$

2.2.3.3 Define $Z_m:=[z_1 \ldots, z_k]$

2.2.3.4 $x=x_0+Z_m y_m$ where $y_m=\mathrm{argmin}_y \| \beta e_1 - H_m y \|_2$, $e_1=[1,0,\ldots,0]^T$ and H_m is the Hessenberg matrix consisting of h

2.2.3.5 $x_0:=x$

The use of matrix pre-conditioner $\boldsymbol{P}_j^{-1}$ is effective for faster convergence. Note that matrix pre-conditioner and time derivative, and low Mach number in this study, pre-conditioning is a quite different idea. Here we use TC-PGS1 defined in the previous section as matrix pre-conditioners in the step 2.2.1. The flexibility of FGMRES allows us to use different number of SGS sweeps in each step. FGMRES(k) with small k sometimes stalls, meaning that solution drops to the local minimum within the specific search space. This can be avoided in this study by simply changing coefficients of each search vector and switches to SGS if stall is detected. Since SGS is used already in FGMRES, this procedure is very simple and needs negligible CPU increase.

7 Numerical Examples

7.1 Computational efficiency for steady problem and optimum parameters for FGMRES(k,m)

For an example of the application to steady flow, inviscid flow around NACA0015 airfoil at M=0.01, that means uniform convective velocity is 1% of sound speed, is computed. FGMRES parameters tested are (1,1), (2,1), (4,1), (8,1), (16,1) and (4,4).

Convergence histories in the physical time step for the residual of momentum are shown in Fig.2. As shown in this figure, convergence becomes faster as FGMRES parameter or restarts increase, however, it almost is saturated at (8,1).

For practical computations, the reduction of overall computational cost is important, therefore, convergence histories in CPU time are then considered. Convergence histories of momentum and density, i.e. continuum equation, in CPU time are shown in Fig.3(a) and 3(b), respectively. Although the convergence characteristics are different between these variables, FGMRES(4,1) or FGMRES(2,1) is the fastest or close to the fastest in convergence. Therefore FGMRES(4,1) is chosen as a standard method. The need for more storage for Kryrov sub-space is disadvantage of GMRES(k,m) for large scale problem, however, this is greatly reduced by use of smaller GMRES parameter.

The comparisons between FGMRES(4,1) and TC-PGS1, that is SGS with modified numerical dissipation discussed in the section V, are shown in Fig.4. TC-PGS1 is the fastest in convergence for momentum residual and FGMRES(4.1) is slightly faster in the convergence of aerodynamic drag. It is mainly because computational cost for one physical time step of TC-PGS1 is under 1/4 of FGMRES(4,1). Thus it can be said that TC-PGS1 is as good as FGMRES(4,1) for this kind of applications. Also note that the case labeled as "TC-PGS1 – Mc =1," which is TC-PGS1 with Mc set as unity, showed much slower convergence compared with "TC-PGS1," in which $Mc=M_\infty$ as stated earlier.

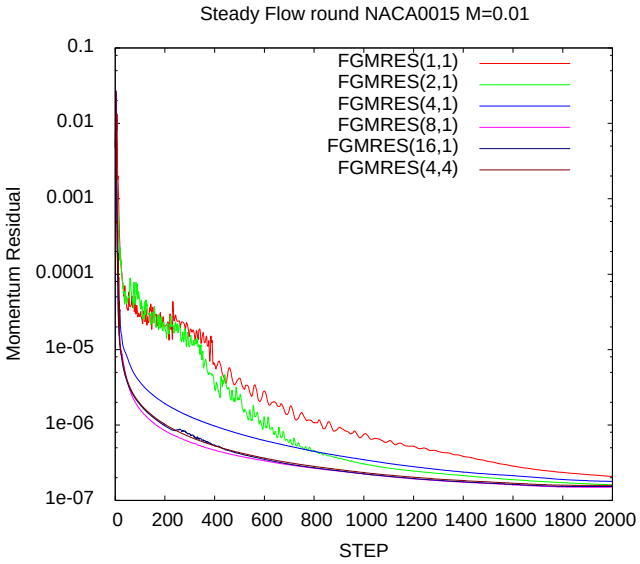
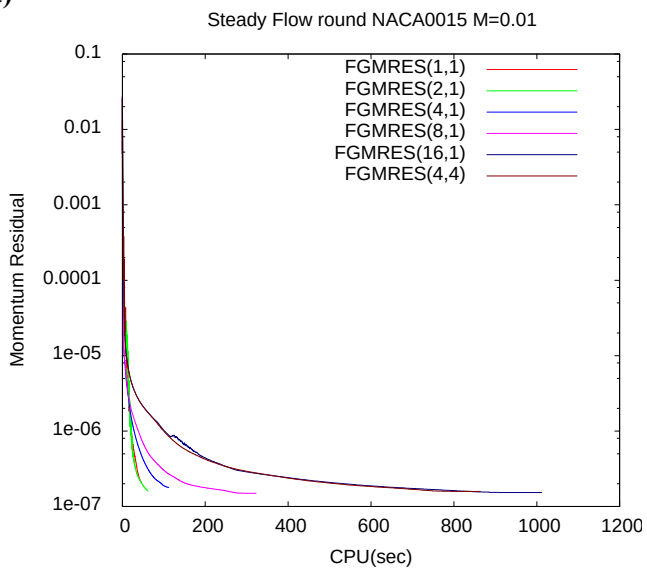


Figure 2. Convergence history of momentum in physical time step for steady inviscid flow around NACA0015 at M=0.01.

a)



b)

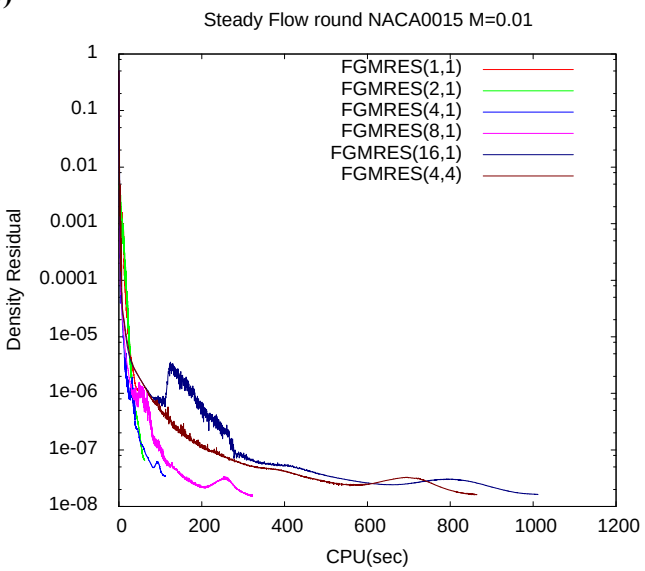


Figure 3. Convergence history of a) momentum and b) density in CPU time for steady inviscid flow around NACA0015 at M=0.01.

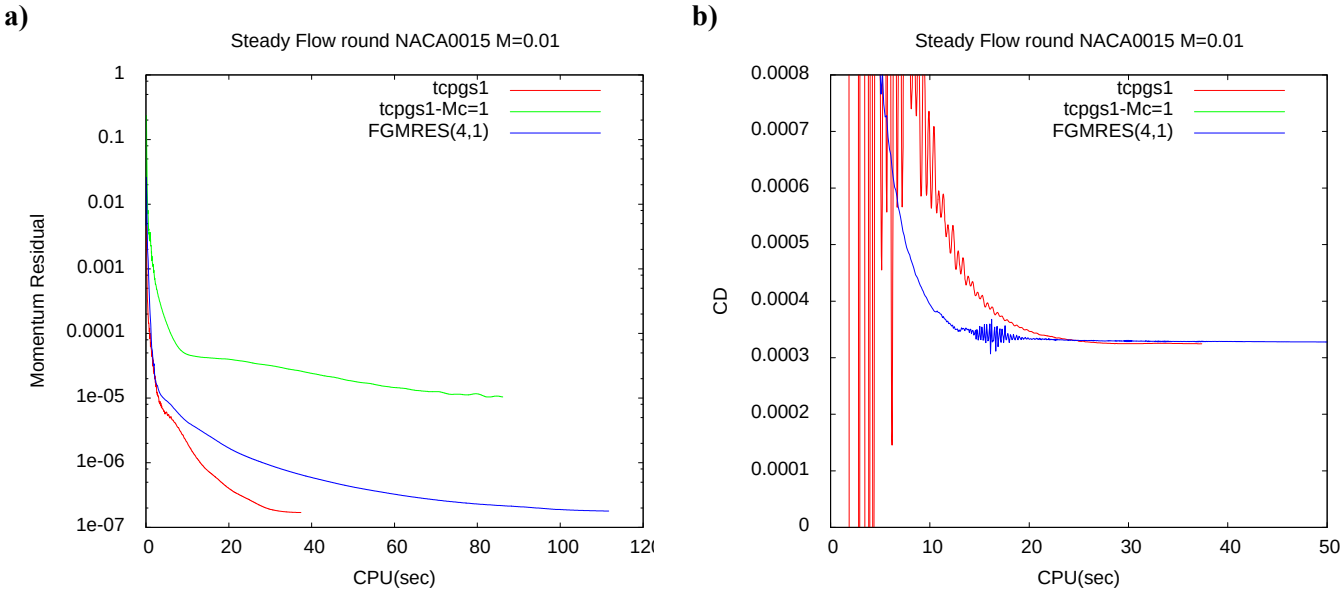


Figure 4. Comparison between FGMRES(4.1) and TC-PGS1: The history of a) momentum and b) aerodynamic drag in CPU time for steady inviscid flow around NACA0015 at M=0.01.

7.2 Accuracy and efficiency for unsteady viscous separated flow around a circular cylinder

For time accurate computations of unsteady flows, non-linear iteration in a physical time step must converge to enough accuracy. The nature of the linear solver gives significant influence to this convergence. The performance of FGMRES(4.1) and TC-PGS1 are compared in low Mach number using 2-dimensional flows around a circular cylinder of Re=100, in which periodic Karman vortex shedding is found, for an example of low Mach number flows. O-type mesh that has 100 cells in azimuthal and 150 cells in radial directions is used. Minimum mesh spacing in radial direction is 10^{-2}R and size of outer radius is 10000R in order to avoid the influence from outer boundary. Free flow Mach numbers is 0.01 and temporal step size is set so that Courant number based on convective velocity is about one. As a result, Courant number based on sound speed become approximately 700. Time integration method is second order, 3-point backward Euler scheme and spatial accuracy is nominally third order using piecewise parabolic reconstruction. In order to give the same initial disturbance, the result after 500 steps is commonly used as initial conditions for all cases.
Cases, number of non-linear iterations and relative CPU time are summarized in Table 1. Time histories of aerodynamic lift coefficients of cases achieving acceptable accuracy are shown in Fig.5(a), and the others are in Fig.5(b). The reference solution is computed by TC-PGS1 with over 30 non-linear iterations. If time accuracy is good enough, the flow should respond very similarly with the reference. It can be said from the results, FGMRES(4.1) over 2 non-linear iterations and TC-PGS1 with 4 non-linear iterations give acceptable accuracy. TC-PGS1 with 2 iteration might be acceptable except for small overshoot before T<100. Although TC-PGS1 is slightly better, it can be said both have comparable performance.

Table 1. Table of cases for effect of outer iteration convergence criteria to time accuracy

CASE	METHOD	Non-linear Iteration	Relative CPU Time
TCPGS 1	TCPGS1	1	0.21
TCPGS 2	TCPGS1	2	0.44
TCPGS 3	TCPGS1	4	1.10
FGMRES 1	FGMRES(4,1)	1	1.00
FGMRES 2	FGMRES(4,1)	2	1.71
FGMRES 3	FGMRES(4,1)	4	3.53

a) **b)**

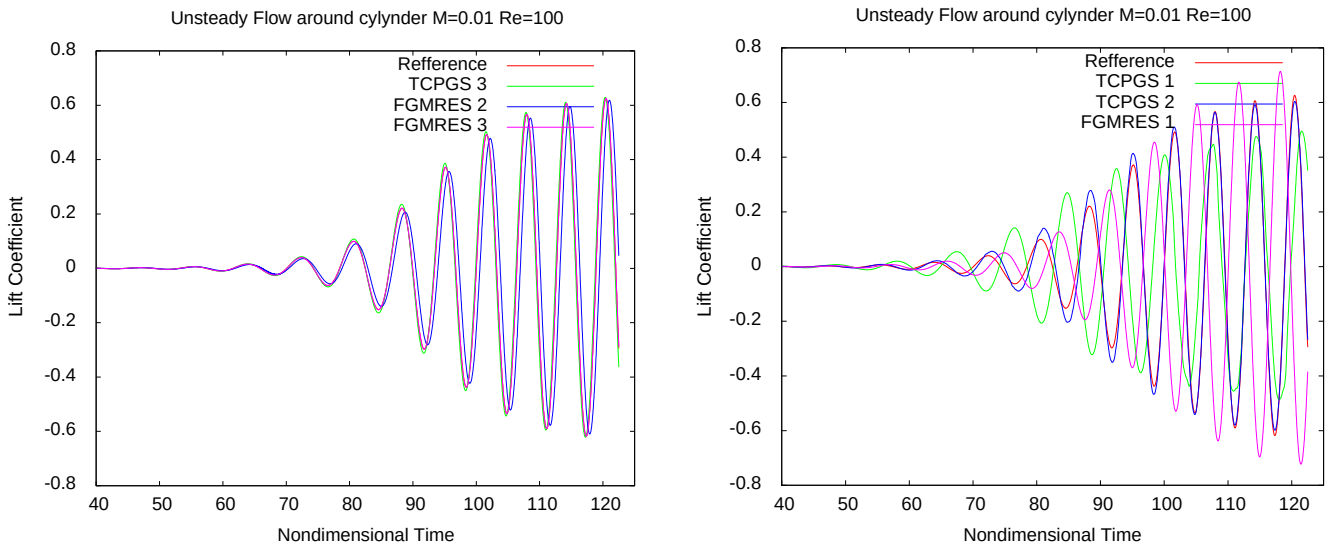


Figure 5. Time histories of aerodynamic lift of a circular cylinder in flow at M=0.01 and Re=100: a) the results with acceptable accuracy, and b) others.

7.3 1D sound propagation by implicit time integration with large Courant number and small cutoff Mach number

It is necessary to use the large Courant number based on sound speed for efficient computation of low Mach number flows. According to simple theoretical analysis, sound propagation can be numerically computed by an ideal implicit time integration method with the large Courant number, when temporal and spatial discretization size (Δt and Δx) are smaller enough than the characteristic scales of sound wave (T and l): It has been shown in the previous study [3] that the condition $T/\Delta t \geq 40$ and $l/\Delta x \geq 40$ gives enough accuracy for second order scheme in both time and space. It indicates that all speed compressible CFD method with implicit time integration method have a potential to compute low Mach number flow and sound propagation at the same time as long as the above condition is satisfied. In order to realize an ideal implicit method, the convergence of the nonlinear iteration is necessary. Thus, as closer the approximation of linear system is to the non-linear system, as faster the convergence becomes.

Accuracy of implicit scheme on sound propagation is numerically tested by one dimensional computation. The computational conditions are as follows;

- TC-PGS1 and FGMRES(4,1) implicit scheme
- Second order 3-point backward Euler method
- Mc=0.01
- Courant number = 10
- 400 PPW (point per wave)
- $T/\Delta t$=40 and $l/\Delta x$=40

Considering the computation with low Mach number flows, small Mc is used. The other parameters are summarized in Table 2 for each case. The pressure distributions in x axis computed by FGMRES are shown in Fig.6(a) and those by TC-PGS1 are shown Fig.6(b). From the results, it can be said 4 iterations give acceptable accuracy and 8 iterations are enough for FGMRES, on the other hand, 200 iterations are acceptable and 300 iterations are enough for TC-PGS1. Therefore, FGMRES(4.1) is over 10 times more efficient for this kind of computations. The poor performance of TC-PGS1 comes from excessive numerical dissipation due to small cutoff number in pressure variable field expressed in Eq.(5.1).

Table 2. Table of cases for effect of outer iteration and time integration methods

CASE	METHOD	Non-linear Iteration	Relative CPU Time
TCPGS 1	TCPGS1	8	1.28
TCPGS 2	TCPGS1	100	16.66
TCPGS 3	TCPGS1	200	33.69
TCPGS 4	TCPGS1	300	50.89
FGMRES 1	FGMRES(4,1)	2	1.00
FGMRES 2	FGMRES(4,1)	4	1.95
FGMRES 3	FGMRES(4,1)	8	3.91

a) **b)**

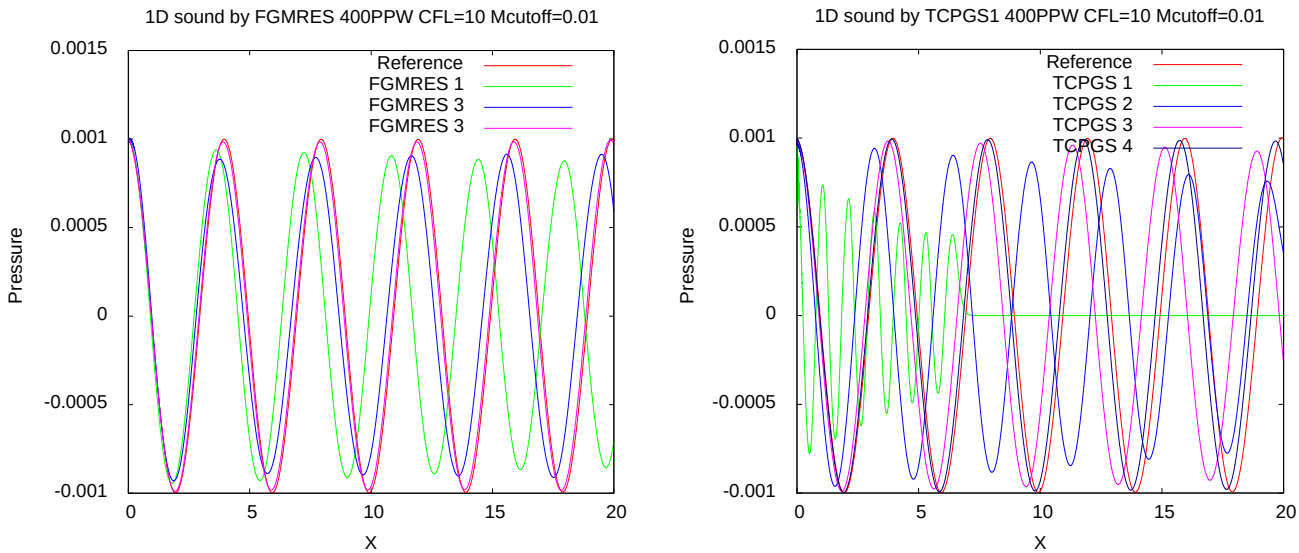


Figure 6. Pressure distribution in one dimensional sound propagation: a) FGMRES(4.1) and b) TC-PGS1.

8. Conclusion and Future Works

The contents of this paper are summarized as follows;

(1) Implicit time integration method for low Mach number compressible flows using all speed numerical flux scheme SLAU (Simple Low-dissipation AUSM) has been investigated to compute sound propagation at the same time in the framework of MUSCL (Monotone Upstream Scheme for Conservation Laws).

(2) The matrix system for implicit time integration derived using the approximate Jacobian of SLAU was found to be non-diagonal dominant, thus, traditional iterative method such as SGS (Symmetirc Gauss-Seidel) could not be used.

(3) One remedy was to modify the implicit numerical dissipation in order to recover diagonal dominance. A simple SGS iteration which does not need block matrix inversion has been derived by using numerical viscosity in diagonal form. This scheme, named TC-PGS1 (Time Consistent Pre-conditioned GS1), has a similarity with the time derivative preconditioning method of Weiss & Smith, in spite of simpler formulation and less dependency upon the cut off number in convergence efficiency.

(4) Another method was to use a modern iterative solver with keeping non-diagonal dominant matrix system. FGMRES of Saad was used as a matrix solver in this study. FGMRES with k restart parameter and m inner iteration was expressed as FGMRES(k,m) in this study. TC-PGS1 was used as the matrix pre-conditioner for FGMRES. The flexibility of FGMRES was utilized to use different iteration of TC-PGS1 in each pre-conditioning procedure. Note that “time derivative or low Mach number pre-conditioning” and “matrix pre-conditioning” are quite different ideas.

(5) FGMRES needs additional memory space for storing searching direction in Krylov subspace and this can be too large for large scale CFD computations in general. However, the optimum number of the restart parameter, which is equivalent to the number of Krylov vectors, was found to be

around 4 in this study. Therefore, the increment of memory was reasonable.

(6) FGMRES(4,1) is about 10 times faster than TC-PGS1 in the (unsteady acoustic) computation of 1D sound propagation when cutoff Mach number was set as 0.01, which is free stream Mach number in this low Mach number flow computation.

(7) When only steady or unsteady flow field is of interest (i.e., acoustics is not), FGMRES and simpler TC-PGS1 gave similar efficiency.

(8) Therefore, it can be said that FGMRES(4,1) will be significantly faster than TC-PGS1 when a low Mach number flow and sound propagation both are computed at the same time, whereas TC-PGS1 is good enough for computing the flow field only.

Followings are listed as future works;

(1) Further improvement of efficiency may be possible by using other iterative linear solver than FGMRES.

(2) Although TC-PGS1 worked well as the matrix pre-conditioner, a better choice may exist.

(3) The performance with large scale parallel computing should be investigated.

(4) These time integration methods can be applied to a variety of compressible flow solvers, although only the second order MUSCL along with SLAU was used here to compute R.H.S. The implicit method is applied after mapping conservative variables to entropy variables, therefore, even finite difference methods in another variable space, such as the primitive variables, can be used.

(5) Aero-acoustic problems, especially flow induced by sound should be computed by the proposed methods.

Acknowledgement

The author thanks to Prof. Fujno of Kyusyu univ., Prof. Nonomura at ISAS and Dr.Iizuka for valuable discussion.

References

[1] E.Shima and K.Kitamura, “Parameter-Free Simple Low-Dissipation AUSM-Family Scheme for All Speeds”, AIAA Journal, vol.49 ,no.8 (2011)

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

[3] Y.Saad, “A flexible inner-outer preconditioned GMRES algorithm”, SIAM Journal on Scientific Computing, (1993)

[4] Shima, E. and Kitamura, K., “CFD Method for Aero-Acoustics using All-Speed Numerical Flux and Preconditioned Implicit Time Integration,” AIAA 2011-3045, 2011.

Appendix

Appendix A. Right hand side evaluation

Viscous terms are computed by a central differencing manner. The evaluation of inviscid terms in MUSCL type schemes are described below. First, distribution of physical quantities in cell is reconstructed using the average value at the cell and the surrounding cells. Second, the values at each cell face are given by this reconstruction. Third, numerical inviscid flux function is calculated using face normal vector $\mathbf{N}$ and values at left($\mathbf{Q}^L$) and right($\mathbf{Q}^R$) side of the face; those are generally discontinuous. As a numerical flux function, SLAU, which is an all speed scheme of AUSM family, is used in this study. Numerical flux is given by;

$$\widetilde{\mathbf{E}} = \widetilde{\mathbf{E}}\left(\mathbf{Q}^L, \mathbf{Q}^R, \mathbf{N}\right) \tag{A.1}$$

Numerical flux function of AUSM family can be written as;

$$\widetilde{\mathbf{E}} = \left(\dot{m} + |\dot{m}|\right)\boldsymbol{\Phi}^L / 2 + \left(\dot{m} - |\dot{m}|\right)\boldsymbol{\Phi}^R / 2 + \tilde{p}\mathbf{N} \tag{A.2}$$

$$\boldsymbol{\Phi} = \left(1, u, v, w, h\right)^T \quad \mathbf{N} = \left(0, x_n, y_n, z_n, 0\right)^T \quad h = (e+p)/\rho \tag{A.3}$$

Here u, v are the velocity components, e, p, and ρ are total energy, pressure and density. Then, choice of mass flux function $\dot{m}$ and average pressure $\tilde{p}$ gives unique formulation of AUSM family schemes. Mass flux of SLAU is given by;

$$\dot{m} = \frac{1}{2}\{(\rho V_n)^L + (\rho V_n)^R - |\overline{V}_n|\Delta\rho\}(1-g) - \chi\Delta p/(2\overline{c}) \tag{A.6}$$

$$\Delta q = q^R - q^L \tag{A.7}$$

$$g = g^L \cdot g^R \quad \in [0,1];$$
$$g^L = -\max[\min(M^L, 0), -1] \tag{A.8}$$
$$g^R = \min[\max(M^R, 0), 1]$$

$$\chi = (1-\hat{M})^2 \tag{A.9}$$

$$\hat{M} = \min\left(1.0, \frac{1}{\overline{c}}\sqrt{\frac{u^{+2}+v^{+2}+w^{+2}+u^{-2}+v^{-2}+w^{-2}}{2}}\right) \tag{A.10}$$

The function g above is a remedy for strong expansion, thus it can be simply set to be zero for low Mach number flows. The average pressure of SLAU is given by;

$$\tilde{p} = \frac{p^L + p^R}{2} + \frac{\beta_+ - \beta_-}{2}(p^L - p^R) + (1-\chi)(\beta_+ + \beta_- - 1)\frac{p^L + p^R}{2} \tag{A.11}$$

$$\beta^{\pm} = \begin{cases} \frac{1}{4}(2 \mp M^{\pm})(M^{\pm} \pm 1)^2, & |M^{\pm}| < 1 \\ \frac{1}{2}(1 + sign(\pm M^{\pm})) & , \text{otherwise} \end{cases} \tag{A.12}$$

$$M^+ = \frac{V_n^L}{\overline{c}}, \quad M^- = \frac{V_n^R}{\overline{c}} \tag{A.13}$$

Appendix B. Derivation of time accurate preconditioning implicit method

Time accurate implicit scheme for FVM can be written as follows using dual time stepping method and time derivative preconditioning matrix Γ;

$$\frac{\mathbf{Q}_i^{k+1} - \mathbf{Q}_i^k}{\Delta\tau} + \mathbf{\Gamma}\{\frac{\theta_1\mathbf{Q}_i^{k+1} - \theta_2\mathbf{Q}_i^n - (\theta_1 - \theta_2)\mathbf{Q}_i^{n-1}}{\Delta t} + \frac{1}{V_i}\sum_j(\tilde{\mathbf{E}}_{i,j}^{k+1} - \tilde{\mathbf{R}}_{i,j}^{k+1})s_{i,j}\} = 0 \tag{B.1}$$

By introducing first order upwind difference and approximate linearization, implicit time integration scheme of Eq.(B.1) is written as;

$$\left[\frac{1}{\Delta\tau} + \frac{\theta_1\mathbf{\Gamma}_i}{\Delta t} + \frac{1}{V_i}\sum_j s_{i,j}\left(\mathbf{\Gamma}_i\tilde{\mathbf{A}}_{i,j}\right)^+\right]\Delta\mathbf{Q}_i - \frac{1}{V_i}\sum_j s_{i,j}\left(\mathbf{\Gamma}_i\tilde{\mathbf{A}}_{j,i}\right)^+\Delta\mathbf{Q}_j = -\mathbf{\Gamma}_i\mathbf{H}_i^k \tag{B.2}$$

$$\mathbf{H}_i^k = \frac{\theta_1\mathbf{Q}_i^k - \theta_2\mathbf{Q}_i^n - (\theta_1 - \theta_2)\mathbf{Q}_i^{n-1}}{\Delta t} + \frac{1}{V_i}\sum_j(\tilde{\mathbf{E}}_{i,j}^k - \tilde{\mathbf{R}}_{i,j}^k)s_{i,j} \tag{B.3}$$

$$\tilde{\mathbf{A}} = \frac{\partial\tilde{\mathbf{E}}}{\partial\mathbf{Q}} - \frac{\partial\tilde{\mathbf{R}}}{\partial\mathbf{Q}} \tag{B.4}$$

$$\Delta\mathbf{Q}_i = \mathbf{Q}_i^{k+1} - \mathbf{Q}_i^k \tag{B.5}$$

Here, $\left(\mathbf{\Gamma}_i\tilde{\mathbf{A}}_{i,j}\right)^+$ is the Jacobian matrix of preconditioned flux at cell interface that has only positive eigenvalues in direction of "i" to "j". The pseudo time step τ can be chosen independently from physical time t, and correct time evolution is recovered for any choice of when the equation is converged about τ. If non-factored implicit schemes such as Gauss-Seidel iteration are used, faster convergence is obtained by bigger $\Delta\tau$. Thus, we take $\Delta\tau$ to be infinitely large, and then, obtained the following equation;

$$\left[\frac{\theta_1\mathbf{\Gamma}_i}{\Delta t} + \frac{1}{V_i}\sum_j s_{i,j}\left(\mathbf{\Gamma}_i\tilde{\mathbf{A}}_{i,j}\right)^+\right]\Delta\mathbf{Q}_i - \frac{1}{V_i}\sum_j s_{i,j}\left(\mathbf{\Gamma}_i\tilde{\mathbf{A}}_{j,i}\right)^+\Delta\mathbf{Q}_j = -\mathbf{\Gamma}_i\mathbf{H}_i^k \tag{B.6}$$

Appendix C. Comparison with time derivative pre-conditioning method for steady flows

As stated before, physical time evolution can be consistently computed by TC-PGS. On the other hand, the preconditioned equation shown below, in which proper physical evolution is neglected, is often used to obtain steady states of low Mach number flows rapidly.

$$\frac{1}{\Delta t}\Delta \mathbf{Q}_i + \frac{\mathbf{\Gamma}_i}{V_i}\sum_j (\widetilde{\mathbf{E}}_{i,j} - \widetilde{\mathbf{R}}_{i,j}) s_{i,j} = 0 \qquad \text{(C.1)}$$

Two examples of steady flows in low Mach number are shown. First, flows around cylinder, using the same mesh used in Sec. VII.B at M=0.01, Re=1 and Re=40, are computed. Since only fast convergence is of interest, first order Euler implicit scheme and one Newton iteration is used. Histories of L2-norm of momentum are shown in Fig.C1, where the method based on Eq.(C.1) is denoted as "Preconditioning". It is clearly seen that TC-PGS converges faster than the conventional preconditioning method in both conditions. In the conventional preconditioning method, excessive implicit dissipation of the viscous component mentioned in Section VI seemed to slow down the convergence.

The other example is an inviscid flow around NACA0012 airfoil. Convergence histories of L2-norms of moment and drag coefficient are shown in Fig.C2. It is found that those convergence histories are different each other; however, converged solutions are obtained in approximately the same time steps. Therefore, it is concluded that TC-PGS gives faster convergence for low Reynolds number flows and is at least as good as the preconditioning method even for inviscid steady flow problems.

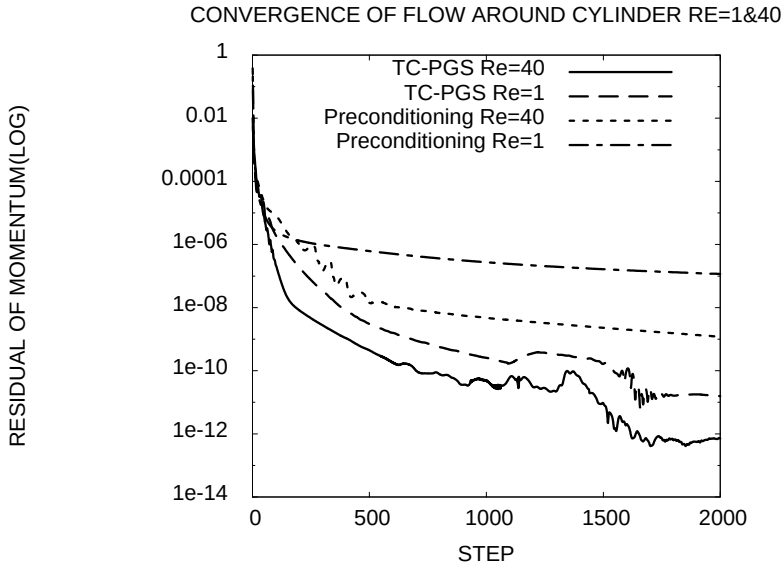
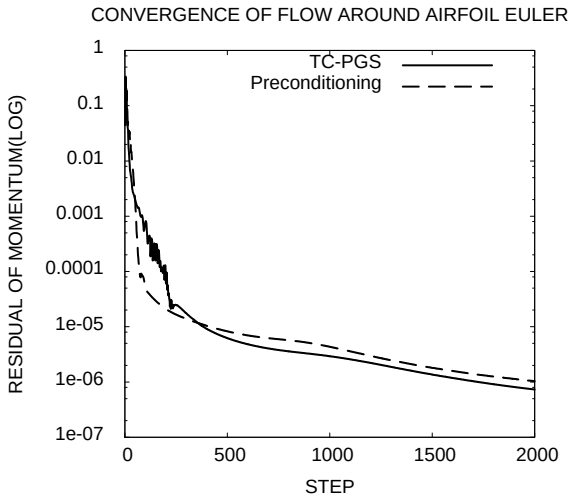


Figure C1. Convergence of residual of flow around a cylinder at M=0.01 and Re=1 & 40.

a)



b)

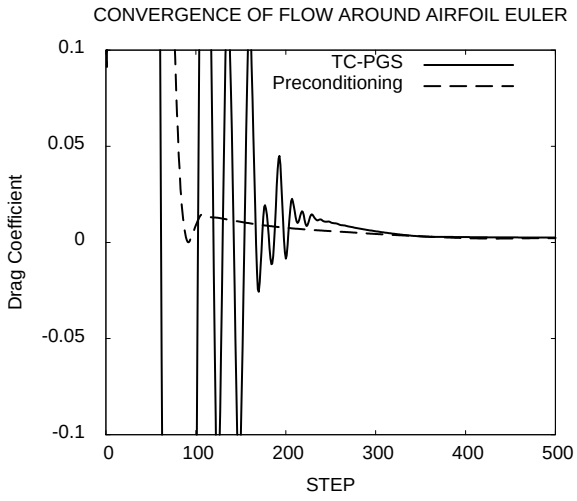


Figure C2. Convergence of a) residual and b) drag coefficient of inviscid flow around a NACA0012 at M=0.01.

Appendix D. Application of TC-PGS1 to LU-SGS

The proposed TC-PGS1 can be easily formulated as a LU-SGS type scheme, since LU-SGS is essensially a pair of symmetric Gauss-Seidel sweep. One step in the Newton iteration using the LU-SGS type scheme can be written as follows.

Step 1: Setting R.H.S and Converting to the working variable

$$\mathbf{H'}_i^{\,k} = \left(\frac{\partial \mathbf{W}}{\partial \mathbf{Q}}\right)_i \left[\frac{\theta_1 \mathbf{Q}_i^{\,k} - \theta_2 \mathbf{Q}_i^{\,n} - (\theta_1 - \theta_2)\mathbf{Q}_i^{\,n-1}}{\Delta t} + \frac{1}{V_i}\sum_j (\tilde{\mathbf{E}}_{i,j}^{\,k} - \tilde{\mathbf{R}}_{i,j}^{\,k}) s_{i,j}\right] \tag{D.1}$$

Step 2: Forward sweep

$$\Delta \mathbf{W}_i^{\,*} = \left[\frac{\theta_1 \mathbf{I}}{\Delta t} + \frac{1}{V_i}\sum_j s_{i,j}\left(\frac{\mathbf{\Delta'}}{2} + \tilde{\mu}\mathbf{I}\right)_{i,j}\right]^{-1} \left[\frac{1}{V_i}\sum_{j\in Lower} s_{i,j}\left(\frac{\hat{\mathbf{B}}_{j,i} + \mathbf{\Delta'}}{2} + \tilde{\mu}\mathbf{I}\right)\Delta \mathbf{W}_j^{\,*} - \mathbf{H'}_i^{\,k}\right] \tag{D.2}$$

Step 3: Backward sweep

$$\Delta \mathbf{W}_i = \Delta \mathbf{W}_i^{\,*} + \left[\frac{\theta_1 \mathbf{I}}{\Delta t} + \frac{1}{V_i}\sum_j s_{i,j}\left(\frac{\mathbf{\Delta'}}{2} + \tilde{\mu}\mathbf{I}\right)_{i,j}\right]^{-1} \left[\frac{1}{V_i}\sum_{j\in Upper} s_{i,j}\left(\frac{\hat{\mathbf{B}}_{j,i} + \mathbf{\Delta'}}{2} + \tilde{\mu}\mathbf{I}\right)\Delta \mathbf{W}_j\right] \tag{D.3}$$

Step 4: Increment to next step

$$\mathbf{Q}_i^{\,k+1} = \mathbf{Q}_i^{\,k} + \left(\frac{\partial \mathbf{Q}}{\partial \mathbf{W}}\right)_i \Delta \mathbf{W}_i \tag{D.4}$$