Comparison Study of Implicit Gauss-Seidel Line Iteration Method for Transonic Flows

Yiqing Shen*, Baoyuan Wang†, Gecheng Zha‡
Dept. of Mechanical and Aerospace Engineering
University of Miami
Coral Gables, Florida 33124
E-mail: yqshen@miami.edu, gzha@miami.edu

Abstract

This paper studies the sweep direction effect on the convergence rate and CPU time of the implicit unfactored Gauss-Seidel line relaxation (GSLR) method for compressible flows. The line Gauss-Seidel iteration is also compared with the LU-SGS (lower-upper symmetric Gauss-Seidel) method. A modified LU-SGS method, namely LU-GSLR, is studied. The LU-GSLR use the unfactored GSLR method with the simple matrix of the LU-SGS. The numerical experiments indicate that for the external flows, the line Gauss-Seidel relaxation methods with sweeping in all directions achieves the optimum convergence rate and CPU efficiency. For an inviscid transonic internal flow, the best convergence rate is obtained with sweeping in streamwise direction only. Within each time step, one sweep (a forward sweep plus a backward sweep) per time step is sufficient. For the three implicit methods, GSLR, LU-SGS, and LU-GSLR, the GSLR is the most efficient method when the Roe scheme is used. The LU-GSLR method is a feasible method and can achieve better efficiency than the LU-SGS for some cases.

1 Introduction

The implicit methods for compressible flow calculation have been widely employed due to their less stiffness and faster convergence rate than the explicit schemes. In general, implicit methods require the inversion of a linearized system of equations. The direct inversion of the linear equations is usually preventively expensive. The implicit linear equations are therefore commonly inverted by iterative methods.

It is known that the approximately factored (AF) implicit schemes such as the Beam-Warming scheme [1] will introduce the factorization errors, which limits the size of the time steps. For 3D linear wave equation, the AF scheme is even not unconditionally stable. The unfactored schemes with no factorization errors such as the line Gauss-Seidel iterations can have larger time steps with faster convergence rate than the AF methods[2, 3, 4, 5, 6, 7]. However, the unfactored schemes typically require more CPU time per iteration since the matrices are usually the full Jacobian matrices and can not be diagonalized.

The lower-upper symmetric Gauss-Seidel (LU-SGS) method suggested by Jameson and Yoon [8, 9] has been widely used due to their relatively easier implicit implementation[10, 11, 12]. The attractive feature of the LU-SGS is that the evaluation and storage of the Jacobian matrices can be eliminated by making some approximations to the implicit operator. Although the LU-SGS method could be more efficient

* Research Scientist, AIAA Member
† Ph.D. Candidate
‡ Associate Professor, AIAA Member

than its explicit counterpart and is unconditionally stable for linear wave equation, the factorization is approximated and will necessarily introduce the factorization errors.

For the unfactored implicit Gauss-Seidel relaxation scheme used to solve the 2D incompressible Navier-Stokes equations, Rogers[13] compared the efficiency of point-Jacobi relaxation (PR), Gauss-Seidel line relaxation (GSLR), incomplete lower-upper decomposition, and the generalized minimum residual method preconditioned with each of the three other schemes. If a forward sweep plus a backward sweep counts as one sweep, Rogers found that these methods can obtain different efficiency when the different number of the sweeps are used. For three-dimensional incompressible flows, Yuan[14] compared the efficiency of the point-Jacobi relaxation, line Gauss-Seidel relaxation, and diagonalized ADI schemes. Yuan[14] observed that the PR(2) (PR with two sweeps) is optimum in all PR(n), and GSLR(1) is optimum in all GSLR(n). For the line Gauss-Seidel relaxation methods, one can choose one or more of the coordinate directions as the sweep direction[3, 15]. For compressible flows, there is few study on how the sweep directions will affect the convergence rate and CPU time.

This paper is to study the implicit methods for compressible flows with following objectives: 1) investigate the sweep direction effect on the convergence rate and CPU time of the implicit unfactored line Gauss-Seidel iteration method; 2) compare the convergence rate and CPU efficiency of the popularly used LU-SGS method and the unfactored line Gauss-Seidel iteration method. 3) study the convergence rate of a new implicit method that combines the matrix of the LU-SGS with the unfactored Gauss-Seidel line relaxation method.

The study concludes that the unfactored Gauss-Seidel line relaxation has the fastest convergence rate and the most efficient CPU time. The computational cases studied in this paper include: 1) a supersonic laminar flow on a flat plate; 2) a subsonic turbulent flow on a flat plate; 3) an inviscid transonic flow in a converging-diverging nozzle. 4) the transonic flow over RAE2822 airfoil.

2 The Implicit Discretization

The normalized Navier-Stokes equations governing compressible viscous flows can be written in the Cartesian coordinate as:

$$\frac{\partial Q}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}+\frac{\partial G}{\partial z}=\frac{1}{Re}\left(\frac{\partial R}{\partial x}+\frac{\partial S}{\partial y}+\frac{\partial T}{\partial z}\right) \tag{1}$$

$$Q=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho e\end{bmatrix},\ E=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ (\rho e+p)u\end{bmatrix},\ F=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vw\\ (\rho e+p)v\end{bmatrix},\ G=\begin{bmatrix}\rho w\\ \rho uw\\ \rho vw\\ \rho w^2+p\\ (\rho e+p)w\end{bmatrix},$$

$$R=\begin{bmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ \tau_{xz}\\ u_k\tau_{xk}-q_x\end{bmatrix},\ S=\begin{bmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ \tau_{yz}\\ u_k\tau_{yk}-q_y\end{bmatrix},\ T=\begin{bmatrix}0\\ \tau_{xz}\\ \tau_{yz}\\ \tau_{zz}\\ u_k\tau_{zk}-q_z\end{bmatrix},$$

The repeated index k stands for the Einstein summation over x,y and z. The stress τ and heat flux q are,

$$\tau_{ik}=\mu\left[(\frac{\partial u_i}{\partial x_k}+\frac{\partial u_k}{\partial x_i})-\frac{2}{3}\delta_{ik}\frac{\partial u_j}{\partial x_j}\right]$$

$$q_j = \frac{-\mu}{(\gamma-1)M_\infty^2 Pr}\frac{\partial T}{\partial x_j}$$

The equation of state is

$$\rho e = \frac{p}{\gamma-1} + \frac{1}{2}\rho(u^2+v^2+w^2)$$

In the generalized coordinates, Eq.(1) can be written as:

$$\frac{\partial Q'}{\partial t} + \frac{\partial E'}{\partial \xi} + \frac{\partial F'}{\partial \eta} + \frac{\partial G'}{\partial \zeta} = \frac{1}{Re}\left(\frac{\partial R'}{\partial \xi} + \frac{\partial S'}{\partial \eta} + \frac{\partial T'}{\partial \zeta}\right) \tag{2}$$

where,

$$Q' = \frac{1}{J}Q,$$
$$E' = \frac{1}{J}(\xi_t U + \xi_x E + \xi_y F + \xi_z G),$$
$$F' = \frac{1}{J}(\eta_t U + \eta_x E + \eta_y F + \eta_z G),$$
$$G' = \frac{1}{J}(\zeta_t U + \zeta_x E + \zeta_y F + \zeta_z G),$$
$$R' = \frac{1}{J}(\xi_x R + \xi_y S + \xi_z T),$$
$$S' = \frac{1}{J}(\eta_x R + \eta_y S + \eta_z T),$$
$$T' = \frac{1}{J}(\zeta_x R + \zeta_y S + \zeta_z T).$$

For simplicity, the prime $'$ in Eq.(2) will be omitted.

In the above equations, ρ is the density, u, v, and w are the Cartesian velocity components in x, y and z directions, p is the static pressure, and e is the total energy per unit mass, μ is the molecular viscosity, J is the transformation Jacobian, γ, Re, M_∞ *and* Pr are the ratio of specific heat, Reynolds number, freestream Mach number and Prandtl number, respectively. Eqs.(2) are discretized into an implicit form as

$$\begin{aligned}\frac{\Delta V}{\Delta t}\Delta Q^{n+1} + (E^{n+1}_{i+\frac{1}{2}} - E^{n+1}_{i-\frac{1}{2}}) + (F^{n+1}_{j+\frac{1}{2}} - F^{n+1}_{j-\frac{1}{2}}) + (G^{n+1}_{k+\frac{1}{2}} - G^{n+1}_{k-\frac{1}{2}}) = \\ \frac{1}{Re}[(R^{n+1}_{i+\frac{1}{2}} - R^{n+1}_{i-\frac{1}{2}}) + (S^{n+1}_{j+\frac{1}{2}} - S^{n+1}_{j-\frac{1}{2}}) + (T^{n+1}_{k+\frac{1}{2}} - T^{n+1}_{k-\frac{1}{2}})]\end{aligned} \tag{3}$$

where Δt is the time interval, ΔV is the volume of control cell.

2.1 Gauss-Seidel Line Relaxation(GSLR)

For different solver, the linearized matrices are different, for example, when the Roe scheme,

$$E_{i+\frac{1}{2}} = \frac{1}{2}[E_L + E_R - \tilde{A}(U_R - U_L)]_{i+\frac{1}{2}} \tag{4}$$

is used, and the surface fluxes E_L (the others are similar as E_L) on time n and $n+1$ can be expressed as,

$$E^{n+1}_L = E^n_L + \frac{\partial E}{\partial U}|^n_L \Delta U^{n+1}_L = E_L + A^L_{i+\frac{1}{2}}\Delta U^{n+1}_L \tag{5}$$

and,

$$\tilde{A}(U_R - U_L)^{n+1} = \tilde{A}(U_R - U_L)^n + \tilde{A}^n(\Delta U_R^{n+1} - \Delta U_L^{n+1}) \tag{6}$$

$$\Delta U_L^{n+1} = \Delta U_i^{n+1}, \Delta U_R^{n+1} = \Delta U_{i+1}^{n+1} \tag{7}$$

then,

$$E_{i+\frac{1}{2}}^{n+1} - E_{i-\frac{1}{2}}^{n+1} = (E_{i+\frac{1}{2}}^n - E_{i-\frac{1}{2}}^n) + \hat{A}_{i+\frac{1}{2}}^R \Delta U_{i+1}^{n+1} + \hat{A}_{i+\frac{1}{2}}^L \Delta U_i^{n+1} - \hat{A}_{i-\frac{1}{2}}^R \Delta U_i^{n+1} - \hat{A}_{i-\frac{1}{2}} \Delta U_{i-1}^{n+1} \tag{8}$$

where,

$$\hat{A}_{i\pm\frac{1}{2}}^R = \frac{1}{2}(A_{i\pm\frac{1}{2}}^R - \tilde{A}_{i\pm\frac{1}{2}}), \hat{A}_{i\pm\frac{1}{2}}^L = \frac{1}{2}(A_{i\pm\frac{1}{2}}^L + \tilde{A}_{i\pm\frac{1}{2}}) \tag{9}$$

Therefore, the final implicit form can be written as,

$$[I + A + B + C]\Delta U_{i,j,k}^{n+1} + A^+ \Delta U_{i+1,j,k}^{n+1} + A^- \Delta U_{i-1,j,k}^{n+1} + B^+ \Delta U_{i,j+1,k}^{n+1} + B^- \Delta U_{i,j-1,k}^{n+1} +$$

$$C^+ \Delta U_{i,j,k+1}^{n+1} + C^- \Delta U_{i,j,k-1}^{n+1} = RHS^n \tag{10}$$

where the coefficient matrices A, $A^\pm$, B, $B^\pm$ and C, $C^\pm$ are defined as

$$A^+ = \frac{\Delta t}{\Delta V}(\hat{A}_{i+\frac{1}{2}}^R - L_{i+\frac{1}{2}}^R) \tag{11}$$

$$A = \frac{\Delta t}{\Delta V}[(\hat{A}_{i+\frac{1}{2}}^L - L_{i+\frac{1}{2}}^L) - (\hat{A}_{i-\frac{1}{2}}^R - L_{i-\frac{1}{2}}^R)] \tag{12}$$

$$A^- = -\frac{\Delta t}{\Delta V}(\hat{A}_{i-\frac{1}{2}}^L - L_{i-\frac{1}{2}}^L) \tag{13}$$

$$B^+ = \frac{\Delta t}{\Delta V}(\hat{B}_{j+\frac{1}{2}}^R - M_{j+\frac{1}{2}}^R) \tag{14}$$

$$B = \frac{\Delta t}{\Delta V}[(\hat{B}_{j+\frac{1}{2}}^L - M_{j+\frac{1}{2}}^L) - (\hat{B}_{j-\frac{1}{2}}^R - M_{j-\frac{1}{2}}^R)] \tag{15}$$

$$B^- = -\frac{\Delta t}{\Delta V}(\hat{B}_{j-\frac{1}{2}}^L - M_{j-\frac{1}{2}}^L) \tag{16}$$

$$C^+ = \frac{\Delta t}{\Delta V}(\hat{C}_{k+\frac{1}{2}}^R - N_{k+\frac{1}{2}}^R) \tag{17}$$

$$C = \frac{\Delta t}{\Delta V}[(\hat{C}_{k+\frac{1}{2}}^L - N_{k+\frac{1}{2}}^L) - (\hat{C}_{k-\frac{1}{2}}^R - N_{k-\frac{1}{2}}^R)] \tag{18}$$

$$C^- = -\frac{\Delta t}{\Delta V}(\hat{C}_{k-\frac{1}{2}}^L - N_{k-\frac{1}{2}}^L) \tag{19}$$

where, as the definition of $A_{i\pm1/2}$, $B_{j\pm1/2}$ and $C_{k\pm1/2}$, $L_{i\pm1/2}$, $M_{j\pm1/2}$ and $N_{k\pm1/2}$ are the Jacobian matrices on cell interface for fluxes F, G, R, S and T, respectively. Superscript R and L denote the right and left of the interface.

$$RHS^n = \frac{\Delta t}{\Delta V}\{[(R_{i+\frac{1}{2}}^n - R_{i-\frac{1}{2}}^n) + (S_{j+\frac{1}{2}}^n - S_{j-\frac{1}{2}}^n) + (T_{k+\frac{1}{2}}^n - T_{k-\frac{1}{2}}^n)] -$$
$$[(E_{i+\frac{1}{2}}^n - E_{i-\frac{1}{2}}^n) + (F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n) + (G_{k+\frac{1}{2}}^n - G_{k-\frac{1}{2}}^n)]\} \tag{20}$$

The Gauss-Seidel line iteration with a certain sweep direction (in this paper, one sweep is defined as a forward sweep plus a backward sweep), for example, ξ direction with the index from small to large, can be written as

$$B^-\Delta U^{n+1}_{i,j-1,k} + \bar{B}\Delta U^{n+1}_{i,j,k} + B^+\Delta U^{n+1}_{i,j+1,k} = RHS' \tag{21}$$

where,

$$\bar{B} = I + A + B + C \tag{22}$$

$$RHS' = RHS^n - A^+\Delta U^n_{i+1,j,k} - A^-\Delta U^{n+1}_{i-1,j,k} - C^+\Delta U^n_{i,j,k+1} - C^-\Delta U^{n+1}_{i,j,k-1} \tag{23}$$

2.2 LU-SGS method [8, 9]

When the LU-SGS method is used, the discretization of eq.(1) is divided into implicit and explicit parts, and can be written as

$$[I + \delta^-_\xi A^+ + \delta^+_\xi A^- + \delta^-_\eta B^+ + \delta^+_\eta B^- + \delta^-_\zeta C^+ + \delta^+_\zeta C^-]\Delta U^{n+1} = RHS^n \tag{24}$$

and,

$$\begin{aligned} A^\pm &= \tfrac{1}{2}\tfrac{\Delta t}{\Delta V}[A \pm \tilde{\rho}(A)I] \\ B^\pm &= \tfrac{1}{2}\tfrac{\Delta t}{\Delta V}[B \pm \tilde{\rho}(B)I] \\ C^\pm &= \tfrac{1}{2}\tfrac{\Delta t}{\Delta V}[C \pm \tilde{\rho}(C)I] \end{aligned} \tag{25}$$

where $\tilde{\rho}(A) = max[|\lambda(A)|]$ and represent a spectral radius of the Jacobian Matrix A with the eigenvalues $\lambda(A)$. If first-order one side differences are used, Eq.(24) can be factored into

$$LD^{-1}U\Delta U^{n+1} = RHS^n \tag{26}$$

where

$$\begin{aligned} L &= \tilde{\rho}I - A^+_{i-1,j,k} - B^+_{i,j-1,k} - C^+_{i,j,k-1} \\ D &= \tilde{\rho}I \\ U &= \tilde{\rho}I + A^-_{i+1,j,k} + B^-_{i,j+1,k} + C^-_{i,j,k+1} \end{aligned}$$

where

$$\tilde{\rho}I = \frac{\Delta t}{\Delta V}[\tilde{\rho}(A) + \tilde{\rho}(B) + \tilde{\rho}(C)]$$

Further rearrangement leads to

$$\begin{aligned} L\Delta U^* &= RHS^n \\ U\Delta U^{n+1} &= D\Delta U^* \end{aligned}$$

or

$$\begin{aligned} \Delta U^* &= D^{-1}[RHS^n + A_{i-1}\Delta U^*_{i-1} + B_{j-1}\Delta U^*_{j-1} + C_{k-1}\Delta U^*_{k-1}] \\ \Delta U^{n+1} &= \Delta U^* - D^{-1}[A_{i+1}\Delta U^{n+1}_{i+1} + B_{j+1}\Delta U^{n+1}_{j+1} + C_{k+1}\Delta U^{n+1}_{k+1}] \end{aligned} \tag{27}$$

In the present study, for the computation of RHS^n, the Roe scheme with the MUSCL(monotone upstream-centered schemes for conservation laws) differencing approach [16] is used for computing inviscid terms, and the second-order central difference scheme is used for viscous terms.

2.3 Modified LU-SGS Method with GSLR

A modified LU-SGS method, namely LU-GSLR, is proposed to replace the matrices of the unfactored line Gauss-Seidel iteration by the LU-SGS matrices. The purpose is to make use of the advantages of the high convergence rate of the unfactored line Gauss-Seidel iteration and the simplicity of the LU-SGS matrices and hopes it may achieve high CPU efficiency.

For Eqs.(21)-(23), the matrices A^+, $\cdots$, C^- and $\bar{B}$ are replaced by the following matrices based on the LU-SGS[8],

$$A^- = \frac{1}{2}\frac{\Delta t}{\Delta V}[A + \tilde{\rho}(A)I]_{i-1}$$

$$A^+ = \frac{1}{2}\frac{\Delta t}{\Delta V}[A - \tilde{\rho}(A)I]_{i+1}$$

$$B^- = \frac{1}{2}\frac{\Delta t}{\Delta V}[B + \tilde{\rho}(B)I]_{j-1}$$

$$B^+ = \frac{1}{2}\frac{\Delta t}{\Delta V}[B - \tilde{\rho}(B)I]_{j+1}$$

$$C^- = \frac{1}{2}\frac{\Delta t}{\Delta V}[C + \tilde{\rho}(C)I]_{k-1}$$

$$C^+ = \frac{1}{2}\frac{\Delta t}{\Delta V}[C - \tilde{\rho}(C)I]_{k+1}$$

and

$$\bar{B} = I + \frac{\Delta t}{\Delta V}[\tilde{\rho}(A) + \tilde{\rho}(B) + \tilde{\rho}(C)]I$$

then, the LU-GSLR method is obtained by using the process of the Gauss-Seidel line iteration Eqs.(21)-(23).

3 Results and Discussion

3.1 Supersonic Laminar Boundary Layer

The first case is a laminar supersonic boundary layer flow on an adiabatic flat plate. The incoming Mach number is 2.0. The Reynolds number based on the length of the flat plate is 4.0×10^4. The Prandtl number of 1.0 is used in order to compare the numerical solution with the analytical solution. The mesh size is 180×60.

The convergence histories for the supersonic flat plate boundary layer are shown in Fig.1. It can be seen that the fastest convergence was obtained by sweeping in both coordinate directions within each time step. The convergence rate of sweeping only in the streamwise direction alternatively (ξ) is slightly less than that of sweeping in both coordinate directions, but is much faster than that of sweeping in the spanwise (η) direction only. This means that even though the sweeping in spanwise (η) direction only is the least efficient, but sweeping in the streamwise and the spanwise direction within each times step is the most efficient, better than sweeping in either direction only.

Figs.2 and 3 show the comparisons of the computed velocity and temperature profiles with the Blasius solutions. They agree excellently.

3.2 Subsonic Flat Plate Turbulent Boundary Layer

The subsonic flat plate turbulent boundary layer is used as the second test example. In this case, the Baldwin-Lomax turbulence model is used. The mesh size is 80×60. The non-dimensional distance y^+ of the first cell center to the wall is kept under 0.2. The inlet Mach number is 0.5, and the Reynolds number is 4×10^6 based on the plate length. The flow is subsonic at inlet and outlet.

For this subsonic turbulent boundary layer flow, the same conclusion as the supersonic flat plate boundary layer is obtained for the convergence histories, which is shown in Fig.4. That is, the Gauss-Seidel line relaxation sweeping in both coordinate directions within each time step is the most efficient, faster than sweeping in either direction only within each time step. Fig.5 shows that the computed velocity profile agrees well with the law of the wall.

3.3 Transonic Converging-Diverging Nozzle

The transonic converging-diverging nozzle is calculated to study the behavior of the Gauss-Seidel iteration for internal flows. The nozzle was designed and tested at NASA and was named as Nozzle A1[17] and is symmetric about the centerline. Hence only the upper half of nozzle is computed. The mesh size is 175×80. The grid is clustered near the wall. The inlet Mach number is 0.22.

Fig.6 compares the convergence histories for the nozzle flow with different sweeping directions. Different from the two flat plate boundary layer flow aforementioned which are external flows, the Gauss-Seidel iteration with two sweeps in both the coordinates direction within each time step is the least efficient one, slower than alternative sweeping two times in either of the coordinate direction, that is the streamwise or spanwise direction.

Fig.7 is the computed pressure contour of this flow.

3.4 Transonic RAE2822 Airfoil

The steady state solution of the transonic RAE2822 airfoil is calculated using the Reynolds averaged NS equation with the Baldwin-Lomax turbulent model. The mesh size is 128×50, $M_\infty = 0.729$, $Re = 6.5 \times 10^6$, the angle of attack is $\theta = 2.31^o$.

For this case, the computational zone is divided into two blocks, and in each block, the mesh in i-direction is defined along the airfoil surface. So the streamwise and the spanwise can not be distinguished by using i or j-direction.

From the convergence histories shown in Fig.8, it can be seen that the fastest convergence rate is obtained by alternative sweeping in j-direction two times within each time step. The sweeping in i-direction has about the same convergence rate as that sweeping in both direction. Fig.9 shows the computed pressure coefficient agree well with the experiment. Fig.10 is the pressure contours, which shows the shock wave captured.

3.5 Comparisons of GSLR, LU-SGS and LU-GSLR

Fig.11 gives the comparison of the convergence histories for the supersonic boundary layer flow using the GSLR, LU-SGS and LU-GSLR methods. It shows that the GSLR is far more efficient than the LU-SGS and LU-GSLR method. The CPU time used by the GSLR is less than one-eighth of LU-SGS. For this case, the LU-GSLR is more efficient than the LU-SGS and its CPU time is about two-third of LU-SGS.

Fig.12 shows the comparison of convergence histories for the subsonic flat plate turbulent boundary layer. In this case, the CPU advantage of the GSLR over the LU-SGS and LU-GSLR is even more greater

than the supersonic flat plate boundary layer flow. GSLR can obtain the converged solution at about 1/16 of the CPU time required by LU-GSLR, and 1/26 of the CPU time required by the standard LU-SGS method.

Fig.13 is the comparison of convergence histories of the transonic converging-diverging nozzle. GSLR behaves the same as for the previous two cases with the highest CPU efficiency and convergence rate. However, for this case, the LU-SGS outperforms the LU-GSLR with less CPU time to converge to machine zero.

Fig.14 shows the comparison of convergence histories for the transonic RAE2822 airfoil. Again, GSLR is the most efficient method. Before the residual reaches the level of 10^{-10}, LU-GSLR is faster than LU-SGS. For the last 3 order of magnitude to machine zero, the convergence rate of the LU-GSLR is decreased and the LU-SGS maintain a linear convergence rate.

It needs to point out that the comparison of the convergence behavior of the GSLR, LU-SGS and LU-GSLR in this paper is for the Roe scheme only. The conclusion that the GSLR is the most efficient method may not be general. For the GSLR, the Jacobian matrix based on Roe scheme is used. For the LU-SGS or LU-GSLR, the general Jacobian matrix suggested by Jameson [8] is used. It is almost certain that there is a matching or compatibility issue. That is when the implicit matrix based on the Roe scheme on the LHS matches the Roe scheme on the RHS such as the GSLR method, high convergence rate can be obtained. When the implicit matrix on the LHS does not match the Roe scheme on the RHS such as the LU-SGS or LU-GSLR method, the convergence rate will be significantly slowed down. However, this does not rule out that a high CPU efficiency may be obtained when the LU-SGS or LU-GSLR is used with other schemes on the RHS such as the scheme used by Jameson[8].

4 Conclusion

Gauss-Seidel line relaxation methods with different sweeping directions have been studied for computing external and internal flows. The numerical experiments indicate that the different sweeping direction has influence on the convergence rate and CPU efficiency. For the computation of external flow, it is best to use Gauss-Seidel line relaxation methods with sweeping in all coordinate directions. However, for the inviscid internal transonic nozzle flow, the better convergence rate is obtained with the sweep in a single direction, not in both the coordinate direction. We also studied the effect of the sweep number on the convergence rate, and found one sweep is optimum as the conclusion in Ref.[14].

In this study, we suggest a modified LU-SGS method, namely LU-GSLR, which uses the unfactored Gauss-Seidel line relaxation with the simple matrix of the standard LU-SGS matrix instead of the full Jacobian matrix of the standard GSLR. Numerical results show that the new method is feasible and can get better convergence rate than LU-SGS in some cases. Among the three implicit methods, GSLR, LU-SGS, and LU-GSLR based on Roe scheme, the GSLR has the fastest convergence rate. It is not certain if the GSLR is still the most efficient method if other scheme instead of the Roe scheme is used for the inviscid fluxes.

5 Acknowledgment

This work is supported by AFOSR Grant FA9550-06-1-0198 monitored by Dr Fariba Fahroo.

References

[1] R. Beam and R. Warming, "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations ," *AIAA Journal*, vol. 16, No. 4, pp. 393–402, 1978.

[2] A.C.Taylor III, W.F.Ng, R.W.Walters, "An improved upwind finite volume relaxation method for high speed viscous flows," *J.Compu.Phys.*, vol. 99, pp. 159–168, 1992.

[3] S.E.Rogers, D.Kwak, "An upwind difference scheme for time accurate incompressible navier-stokes equations," *AIAA J.*, vol. 28, pp. 253–262, 1990.

[4] S.E.Rogers, F.R.Menter, N.N.Mansour, and P.A.Durbin, "A comparison of turbulence models in computing multi-element airfoil flows." AIAA 94-0291, Jan. 1994.

[5] G.-C. Zha and E. Bilgen, "Numerical Study of Three-Dimensional Transonic Flows Using Unfactored Upwind-Relaxation Sweeping Algorithm," *Journal of Computational Physics*, vol. 125, pp. 425–433, 1996.

[6] G.-C. Zha and Z.-J. Hu, "Calculation of Transonic Internal Flows Using an Efficient High Resolution Upwind Scheme," *AIAA Journal*, vol. 42, No. 2, pp. 205–214, 2004.

[7] X.Chen and G.-C.Zha, "Fully coupled fluid-structural interactions using an efficient high solution upwind scheme," *Journal of Fluid and Structure*, vol. 20, pp. 1105–1125, 2005.

[8] A. Jameson and S.Yoon, "Lower-upper implicit schemes with multiple grids for the Euler equations," *AIAA J.*, vol. 7, pp. 929–935, 1987.

[9] S.Yoon and A.Jameson, "Lower-upper symmetric-gauss-seidel method for the euler and navier-stokes equations," *AIAA J.*, vol. 26, pp. 1025–1026, 1988.

[10] G.H.Klopfer and S.Yoon, "Multizonal Navier-Stokes Code with LU-SGS Scheme." AIAA 93-2965, July 1993.

[11] M.Kandula and P.G.Buning, "Implementation of LU-SGS Algorithm and Roe Upwinding Scheme in OVERFLOW Thin-Layer Navier-Stokes Code." AIAA 94-2357, Jun. 1994.

[12] A.Jameson and D.A.Caughey, "How many steps are required to solve the Euler equations of steady, compressible flow: in search of a fast solution algorithm." AIAA-2001-2673, Jun. 2001.

[13] S.E.Rogers, "Comparison of implicit schemes for the incompressible navier-stokes equations," *AIAA J.*, vol. 33, pp. 2066–2072, 1995.

[14] L.Yuan, "Comparison of implicit multigrid schemes for three-dimensional incompressible flows," *J. Compu. Phys.*, vol. 177, pp. 134–155, 2002.

[15] S.E.Rogers, D.Kwak, C.Kiris, "Numerical solution of the incompressible navier-stokes equations for steady-state and time-dependent problems," *AIAA J.*, vol. 29, pp. 603–610, 1991.

[16] B.van Leer, "Towards the ultimate conservative difference scheme, v:a second-order sequel to godunov's method," *J. Compu. Phys.*, vol. 32, pp. 101–136, 1979.

[17] M. L. Mason and L. E. Putnam, "The Effect of Throat Contouring on Two-Dimensional Converging-Diverging Nozzles at Static Conditions ." NASA Technical Paper 1704, 1980.

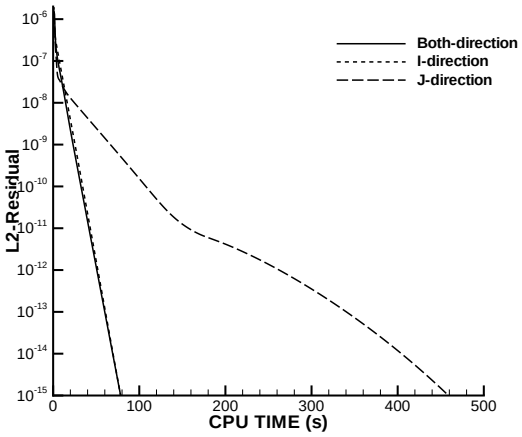

Figure 1: L2-norm residual vs CPU time, the supersonic boundary layer flow

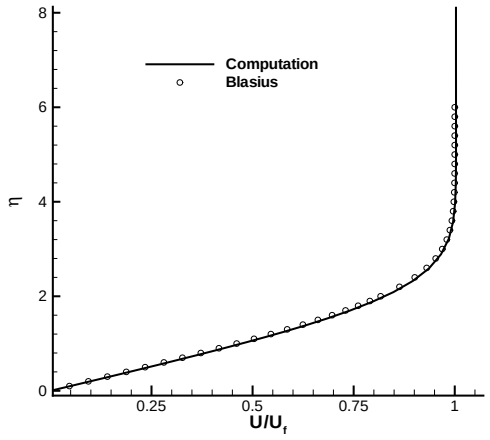

Figure 2: Velocity profile of the supersonic boundary layer flow

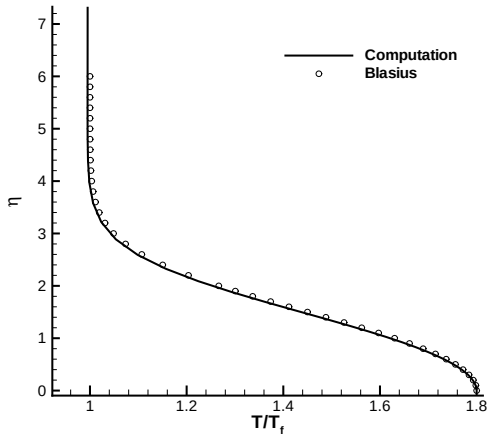

Figure 3: Temperature profile of the supersonic boundary layer flow

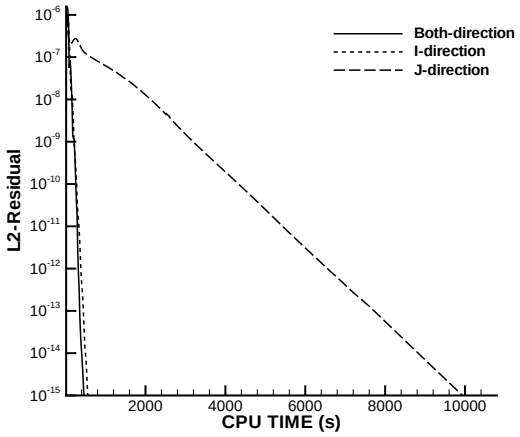

Figure 4: L2-norm residual vs CPU time, the turbulent boundary layer

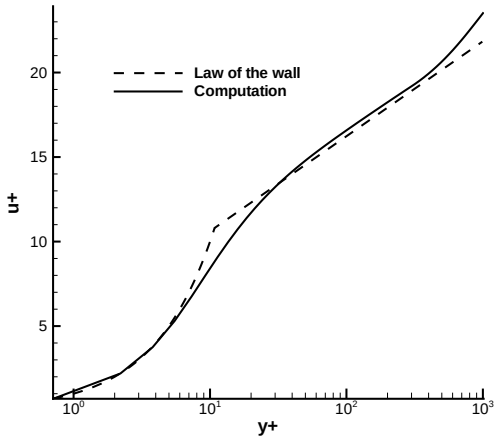

Figure 5: Computed velocity profile compared with the law of the wall

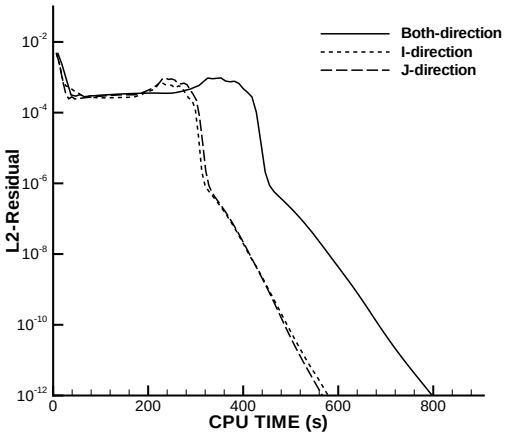

Figure 6: L2-norm residual vs CPU time, the transonic converging-diverging nozzle flow

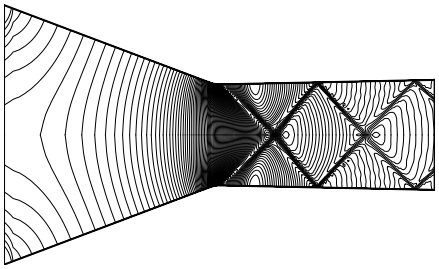

Figure 7: Pressure contours of the transonic converging-diverging nozzle flow

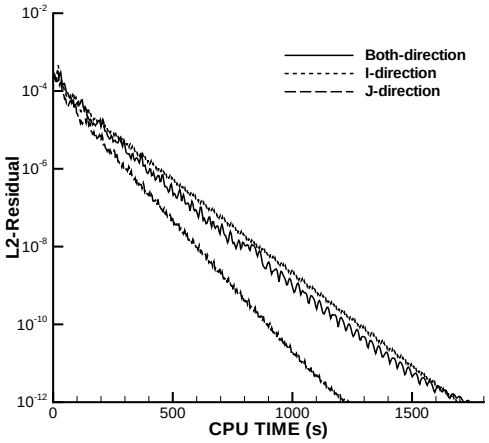

Figure 8: L2-norm residual vs CPU time, the transonic flow over RAE2822 airfoil

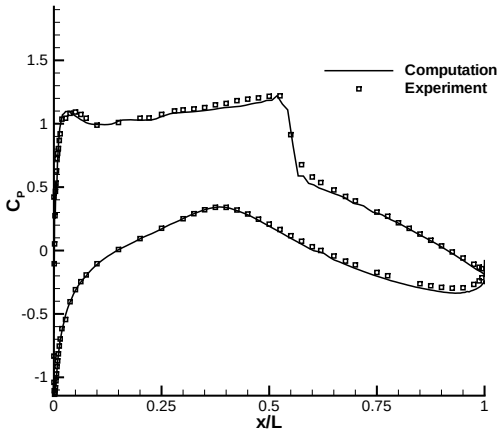

Figure 9: Pressure coefficients at the airfoil surface of the transonic flow over RAE2822 airfoil

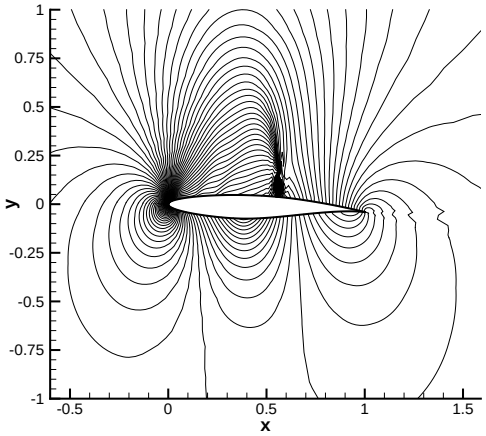

Figure 10: Pressure contours of of the transonic flow over RAE2822 airfoil

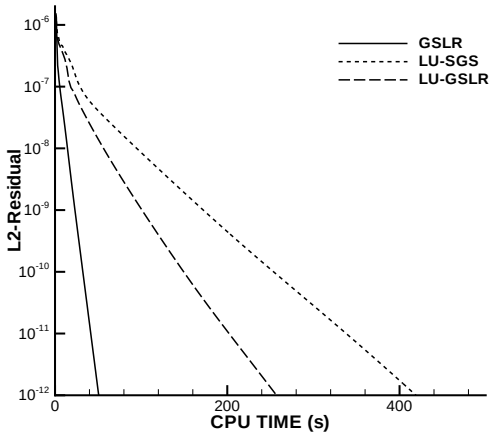

Figure 11: Comparison of L2-norm residual vs CPU time, the supersonic boundary layer flow

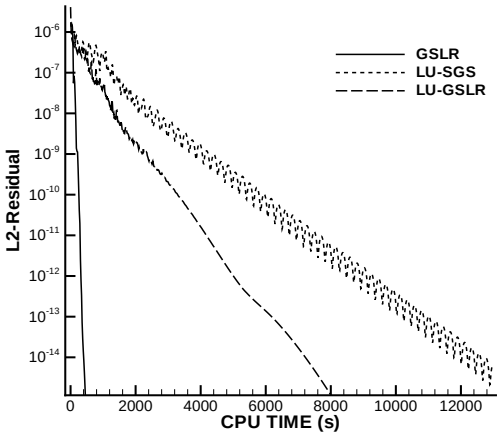

Figure 12: Comparison of L2-norm residual vs CPU time, the turbulent boundary layer

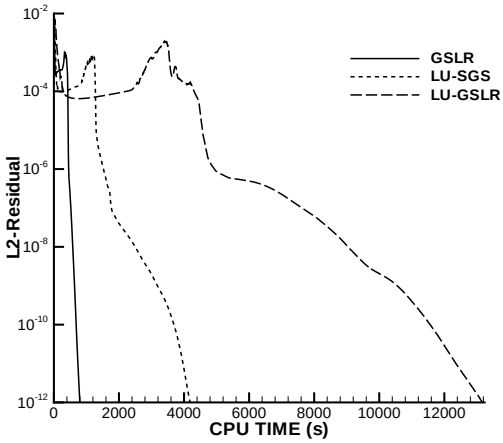

Figure 13: Comparison of L2-norm residual vs CPU time, the transonic converging-diverging nozzle flow

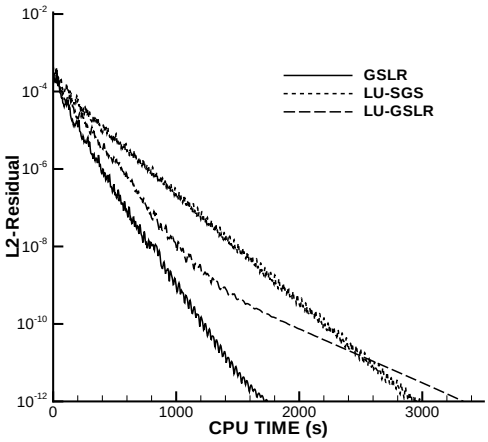

Figure 14: Comparison of L2-norm residual vs CPU time, the transonic flow over RAE2822 airfoil