Parameter Investigation with Line-Implicit Lower-Upper Symmetric Gauss-Seidel on 3D stretched grids

Evelyn Otero*

KTH Royal Institute of Technology, Stockholm, SE-100 44, Sweden

Peter Eliasson†

FOI Swedish Defence Research Agency, Stockholm, SE-16490, Sweden

An implicit Lower-Upper Symmetric Gauss-Seidel (LU-SGS) solver has been implemented as a multigrid smoother combined with a line-implicit method as an acceleration technique for Reynolds-Averaged Navier-Stokes (RANS) simulation on stretched meshes. The Computational Fluid Dynamics code concerned is Edge, an edge-based finite volume Navier-Stokes flow solver for structured and unstructured grids. The paper focuses on the investigation of the parameters related to our novel line-implicit LU-SGS solver for convergence acceleration on 3D RANS meshes. The LU-SGS parameters are defined as the Courant-Friedrichs-Lewy number, the Left Hand Side dissipation, and the convergence of iterative solution of the linear problem arising from the linearisation of the implicit scheme. The influence of these parameters on the overall convergence is presented and default values are defined for maximum convergence acceleration. The optimized settings are applied to 3D RANS computations for comparison with explicit and line-implicit Runge-Kutta smoothing. For most of the cases, a computing time acceleration of the order of 2 is found depending on the mesh type, namely the boundary layer and the magnitude of residual reduction.

Nomenclature

CFL	Courant-Friedrichs-Lewy number
$\mathbf{F_I}$	Inviscid flux matrix
$\mathbf{F_V}$	Viscous flux matrix
$\mathbf{M}$	Linear system matrix
$N(i)$	Set of neighboring nodes of node i
$\mathbf{n}_{ij}$	Normal vector to the surface between node i and node j
$\mathbf{q}$	Source term vector
$\mathbf{r}$	Residual vector
S_{ij}	Surface area between node i and node j
$\mathbf{u}$	Conservative variables vector
V_i	Control volume centered at node i
Δt	Time step, s
ϵ	Left Hand Side dissipation coefficient
λ_{max}	Spectral radius
<i>Subscript</i>	
i	Grid node number
ij	Edge between node i and j

*PhD student, Department of Aeronautical and Vehicle Engineering, AIAA Student Member.

†Deputy Research Director, Department of Computational Physics, AIAA Member.

I. Introduction

Lower-Upper Symmetric Gauss-Seidel (LU-SGS) is a well known method for convergence acceleration of inviscid and viscous flow simulations¹ with minimum memory requirement. Its efficiency has been improved through different methods over the last decades. First developed for structured grids by Jameson and Yoon,² LU-SGS has been successfully extended to hybrid structured-unstructured grids.^{3,4} Techniques for optimization have been developed such as the definition of a proper implicit operator⁵⁻⁷ and the combination with the generalized minimal residual (GMRES) solver as a preconditioning method.^{8,9} The line-implicit LU-SGS solver¹⁰ has been implemented in the Edge¹¹ CFD solver as smoother for the multigrid technique. Edge is an edge-based flow solver for unstructured grids based on a node-centered finite-volume discretization and a dual grid. Several acceleration techniques exist in Edge such as the agglomeration multigrid (MG) iteration based on a Full Approximation Storage (FAS) scheme.¹² A line-implicit method¹³⁻¹⁵ accelerates the computation by integrating implicitly in time along lines where the grid is highly stretched for Reynolds-Averaged Navier-Stokes (RANS) models.

The originality of the implementation discussed in this paper comes from our most novel approach line-implicit LU-SGS solver¹⁰ and the integration and analysis of a set of “external” parameters controlling the impact of the solver on the overall convergence. The “external” parameters are accessible in a flexible machinery which can be adapted to the problem solved, giving the user control over the solver behavior and its convergence. These parameters have shown to have a relevant role to get the best convergence for each specific case. The improved results since the first computations carried out,¹⁰ reveals in fact the successful impact of the present work.

The paper starts with the mathematical formulation introducing the governing equations and the LU-SGS solver. A detailed description of the parameters is then presented showing their influence on the convergence, their interdependencies, and the optimized combinations. Finally, some results illustrate the investigation carried out based on convergence comparisons of the implicit and line-implicit LU-SGS with the explicit and line-implicit Runge-Kutta (RK) solver.

II. RANS equations

The compressible Reynolds-Averaged Navier-Stokes equations solved in Edge¹¹ are,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{F}_I - \mathbf{F}_V) = \mathbf{q} \quad (1)$$

$\mathbf{u}$ is the vector of the conserved variables $\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}$ with ρ the density, u, v, w the velocities and E the total

energy. $\mathbf{F}_I$ and $\mathbf{F}_V$ are the matrices of the inviscid and viscous fluxes¹¹ respectively and $\mathbf{q}$ the source term vector.¹¹

The RANS equations, Eq. (1), are discretized in space with the finite volume method by integrating over a volume Ω contained in the surface S , Eq. (2).

$$\int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} d\Omega + \int_S (\mathbf{F}_I - \mathbf{F}_V) \mathbf{n} dS = \int_{\Omega} \mathbf{q} d\Omega \quad (2)$$

Equation (2) is numerically integrated at node v_i surrounded by the control volume V_i by summing up the fluxes computed on each connected edge ij . The spatially discretized system becomes Eq. (3),

$$\frac{d}{dt}(\mathbf{u}_i V_i) + \sum_{j \in N(i)} \mathbf{F}_{\mathbf{I}ij} \mathbf{n}_{ij} S_{ij} - \sum_{j \in N(i)} \mathbf{F}_{\mathbf{V}ij} \mathbf{n}_{ij} S_{ij} = \mathbf{q}_i V_i \quad (3)$$

with $N(i)$ the set of neighboring nodes to node v_i . Figure 1 illustrates the finite volume discretization of Eq. (3).

Figure 1. Dual grid, control volume and flux vector definitions.

III. LU-SGS

III.A. Implicit time discretization

In the present paper we solve the nonlinear steady state problem of Eq. (3) in the following form,

$$\mathbf{r}(\mathbf{u}) = 0 \quad (4)$$

with $\mathbf{r}$ the nonlinear residual vector resulting from the spatial discretization of the RANS equations, Eq. (3). The nonlinear problem of Eq. (4) is solved by a time marching method, Eq. (5).

$$V \cdot \frac{d\mathbf{u}}{dt} + \mathbf{r}(\mathbf{u}) = 0 \quad (5)$$

A Backward Euler (“BE”) temporal discretization is applied to Eq. (5) obtaining the final discretized system, Eq. (6), allowing for different time steps Δt_i in different cells V_i ,

$$\text{diag} \left(\frac{V_i}{\Delta t_i} \right) \cdot (\mathbf{u}^{n+1} - \mathbf{u}^n) + \mathbf{r}(\mathbf{u}^{n+1}) = 0 \quad (6)$$

The implicit method leads to a nonlinear system of equations to be solved in each time step for $\mathbf{u}^{n+1}$. The residual $\mathbf{r}(\mathbf{u}^{n+1})$ is linearized as $\mathbf{r}(\mathbf{u}^{n+1}) \approx \mathbf{r}(\mathbf{u}^n) + \frac{\partial \mathbf{r}(\mathbf{u}^n)}{\partial \mathbf{u}} \Delta \mathbf{u}^n$ and we obtain Eq. (7) in the form of $\mathbf{M}\mathbf{x} = \mathbf{b}$,

$$\underbrace{\left(\text{diag} \left(\frac{V_i}{\Delta t_i} \right) + \frac{\partial \mathbf{r}(\mathbf{u}^n)}{\partial \mathbf{u}} \right)}_{\mathbf{M}} \underbrace{\Delta \mathbf{u}^n}_{\mathbf{x}^n} = \underbrace{-\mathbf{r}(\mathbf{u}^n)}_{\mathbf{b}} \quad (7)$$

with

$$\Delta \mathbf{u}^n = \mathbf{u}^{n+1} - \mathbf{u}^n \quad (8)$$

and $\mathbf{M}$ and $\mathbf{b}$ defined as the Left Hand Side (LHS) and Right Hand Side (RHS) respectively.

The next step to be defined is the construction of the implicit operator $\mathbf{M}$ and the method of resolution. Newton’s method results when $\frac{1}{\Delta t} \rightarrow 0$ and requires the exact Jacobian of $\mathbf{r}$ in $\mathbf{M}$, Eq. (7) and the exact

solution of the linear system of Eq. (7). These two last characteristics make the method highly memory demanding and computationally expensive. Thus the idea is to compute approximately the Jacobian of $\mathbf{r}$ and solve the system approximately with an inexact linear system solver. The construction of the implicit operator $\mathbf{M}$ has been previously investigated¹⁶ favoring the choice of a so-called *exact first-order Jacobian*⁷ for the Jacobian approximation. More precisely, the Jacobian is the exact derivative of a first-order upwind scheme whereas the RHS is a second-order accurate spatial discretization $\mathbf{b}$ in Eq. (7). The iterative solver selected for solving the linear system of Eq. (7) is LU-SGS which requires LU-factorization of only block-diagonal parts of the matrix. For the line-implicit method the LU-factorization applies also to a block tridiagonal part containing the block diagonal and blocksub- and blocksuper diagonals of the “implicit lines”.¹⁰

III.B. Implicit LU-SGS

The LU-SGS algorithm consists in splitting the block system matrix $\mathbf{M}$ into a block lower triangular part $\mathbf{L}$, a block diagonal $\mathbf{D}$, and a block upper triangle $\mathbf{U}$ as in Eq. (9).

$$\mathbf{M}\mathbf{x} = (\mathbf{L} + \mathbf{D} + \mathbf{U})\mathbf{x} = \mathbf{b} \quad (9)$$

A block is the set of unknowns per node. LU-SGS solves through a forward and backward sweep defined by the following two equations,

$$\begin{aligned} (\mathbf{D} + \mathbf{L})\mathbf{x}^* &= \mathbf{b} - \mathbf{U}\mathbf{x}^n \\ (\mathbf{D} + \mathbf{U})\mathbf{x}^{n+1} &= \mathbf{b} - \mathbf{L}\mathbf{x}^* \end{aligned} \quad (10)$$

If several sweep iterations are carried out, cf. IV.C.2, it becomes the so-called Symmetric Gauss-Seidel (SGS) iteration. However, we use the LU-SGS denomination independently of the number of sweeps. For viscous computations on stretched meshes a line-implicit is easily implemented¹⁰ in the LU-SGS framework and used in the investigation.

III.C. Line-implicit LU-SGS

In our new implementation of LU-SGS with line-implicit method¹⁰ the block diagonal matrix $\mathbf{D}$ in $\mathbf{M}$ of Eq. (9), is replaced by a matrix $\mathbf{B}$ which is either a block tridiagonal matrix along the lines or a block diagonal matrix outside the lines. Based on the new definition, $\mathbf{M} = \mathbf{L} + \mathbf{B} + \mathbf{U}$, the modified algorithm becomes,

For $iter = 1 : N_{swp}$

Forward sweep

For $l = 1, 2, \dots, N_l$

$$\mathbf{B}_l \mathbf{x}_l^* = \mathbf{b}_l - \mathbf{L}_l \mathbf{x}^* - \mathbf{U}_l \mathbf{x}^n$$

Backward sweep

For $l = N_l, N_l - 1, \dots, 1$

$$\mathbf{B}_l \mathbf{x}_l^{n+1} = \mathbf{b}_l - \mathbf{U}_l \mathbf{x}^{n+1} - \mathbf{L}_l \mathbf{x}^*$$

$n = n + 1;$

End $iter;$

with l , the line number, N_l the total number of lines and the matrices computed as,

$$\mathbf{B}_{ii} = \frac{V_i}{\Delta t_i} + \sum_{j \in N(i)} \frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{u}_i} \quad (11)$$

$$\{\mathbf{L}_{ij}, \mathbf{U}_{ij}\}_{j \in N(i)} = \frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{u}_j} \quad (12)$$

with $\mathbf{F} = \mathbf{F}_I - \mathbf{F}_V$.

IV. Parameter study

IV.A. Convergence terms and definitions

The parameter investigation is based on the analysis of the overall convergence per MG cycle or time-step. It is henceforth called nonlinear convergence per iteration. The convergence of the linear problem of Eq. (7) is not investigated in detail. The convergence rate is measured in log-residual decrease per time-step, e.g. decades per kilo steps. Also log-residual decrease vs. wall clock time is reported.

IV.B. LU-SGS parameters

The parameters can be divided into two families, the parameters proper to the LU-SGS solver and the ones belonging to the multigrid acceleration. The LU-SGS parameters are: the number of iterations of the linear system (sweeps), the magnitude of the LHS “entropy fix”, and the CFL number. The MG process is controlled by the number of levels and the smoother type per grid level.

The number of sweeps N_{sweep} is the number of block Gauss-Seidel iterations in a time-step, also called linear iterations,

For $n = 1 : N_{sweep}$

Forward sweep: solve $\mathbf{x}^*$ from

$$(\mathbf{D} + \mathbf{L}) \mathbf{x}^* = \mathbf{b} - \mathbf{U}\mathbf{x}^n$$

Backward sweep: solve $\mathbf{x}^{n+1}$ from

$$(\mathbf{D} + \mathbf{U}) \mathbf{x}^{n+1} = \mathbf{b} - \mathbf{L}\mathbf{x}^*$$

End *iter*;

The LHS “entropy fix” parameter ϵ controls the level of artificial dissipation in the LHS dissipation, Eq. (13).

$$\left(\frac{V_i}{\Delta t_i} + \sum_{j \in N(i)} \mathbf{A}_{ij}^+ \right) \Delta \mathbf{u}_i^n + \sum_{j \in N(i)} \mathbf{A}_{ij}^- \Delta \mathbf{u}_j^n = -\mathbf{r}_i(\mathbf{u}^n) \quad (13)$$

$$\begin{aligned} \mathbf{A}_{ij}^- &= \frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{u}_j} = \frac{1}{2} \left(\left(\frac{\partial \mathbf{F}}{\partial \mathbf{u}} \right)_j - |\mathbf{A}''|_{ij} \right) \\ \mathbf{A}_{ij}^+ &= \frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{u}_i} = \frac{1}{2} \left(\left(\frac{\partial \mathbf{F}}{\partial \mathbf{u}} \right)_i + |\mathbf{A}''|_{ij} \right) \end{aligned} \quad (14)$$

and $|\mathbf{A}''|$ the matrix

$$|\mathbf{A}''| = \mathbf{R}|\mathbf{\Lambda}''|\mathbf{R}^{-1} \quad (15)$$

The matrix $\mathbf{R}$ contains the right eigenvectors as columns of the average flux Jacobian $\mathbf{A}''$. The diagonal matrix $\mathbf{\Lambda}''$ holds the eigenvalues λ_k increased by a fraction ϵ of the spectral radius $\lambda_{max_{ij}}$,

$$|\lambda_k''|_{ij} = |\lambda_k|_{ij} + \epsilon \lambda_{max_{ij}} \quad (16)$$

where $\lambda_{max_{ij}} = \max_k |\lambda_k|_{ij}$.

The CFL number defines the local time step of the linear system of Eq. (13), as

$$\Delta t_i = CFL \frac{V_i}{\max_j \lambda_{max_{ij}}} \quad (17)$$

Well-chosen implicit schemes aim to remove the restriction on the time step required for explicit schemes, and the CFL number can be increased to accelerate the convergence to steady state. Note that only if the system of Eq. (13) were solved exactly, and the original equations were linear, the CFL number could be increased “unconditionally”.

IV.C. Parameter influence and interaction

IV.C.1. CFL number and LHS dissipation

For highest convergence acceleration the parameter ϵ should be reduced as much as possible and the CFL number increased. The ϵ and CFL number variations are considered in the following ranges, $\epsilon \in [0.1 : 1]$ and $CFL \in [10 : 1000000]$. From the experiments carried out, Figure 2 illustrates the effect of the ϵ variation on the convergence rate. The minimum value ϵ_{min} defines a limit where the maximum convergence rate is reached but below which the overall iterations do not converge. Higher values degrade the convergence rate, and 0.1 has been found in many cases as ϵ_{min} for small CFL numbers.

ϵ_{min} is defined for a specific CFL number that we call CFL_{max} . The interplay between ϵ and the CFL number can be analyzed by noting the approximate expression,

$$\max_j \lambda_{max_{ij}} \left(\frac{1}{CFL} + \epsilon \right) \quad (18)$$

for the diagonal elements of the matrix $\mathbf{M}$, Eqs. (7) and (13).

From the expression of Eq. (18) and figure 2, one understands that increasing the CFL number has the same effect on the matrix diagonal blocks as decreasing ϵ . The main difference is that ϵ also affects the off-diagonal terms. However, a balance should be maintained in Eq. (18) for overall convergence. This means that an $\epsilon = \epsilon_{min}$ value will give convergence for $CFL \leq CFL_{max}$. And the other way around, $CFL = CFL_{max}$ can be combined with $\epsilon \geq \epsilon_{min}$. We define then the optimized combination as $[\epsilon_{min}, CFL_{max}]$. Higher CFL numbers than CFL_{max} will require the increase of ϵ_{min} shifting the curve of figure 2 to the right. For example, a factor of 10 applied to the CFL_{max} will shift the ϵ_{min} value by +0.1. For most of the cases, $\epsilon = 0.2$ and $CFL = 1000$ was close to optimal and is defined as the default combination. When the default does not work well, as small a change as increasing ϵ by adding a fraction of 0.1 or decreasing CFL by a factor of 10 will often suffice. In this case our recommendation is to reduce the CFL number, section V.A.

The reasons for the observed behavior is as follows. Whereas a decrease of the CFL number is beneficial for the convergence of the linear problem by improving the diagonal dominance,⁷ it also increases the number of time steps and decelerates the nonlinear convergence. Therefore, big CFL numbers are more beneficial for overall convergence. When ϵ is increased, the approximation of the RHS in Eq. (7) becomes less accurate which decelerates the nonlinear convergence.⁷

IV.C.2. Number of sweeps

The number of sweeps is another parameter with strong influence on the convergence rate. Figure 3 illustrates the effect of the variation of the number of sweeps in the overall convergence in terms of iterations and time, based on the default combination $\epsilon = 0.2$, $CFL = 1000$, section IV.C.1. Increasing the number of sweeps increases the convergence rate, until a specific number of sweeps from which it becomes almost constant. In the computing time this shows up as a phase of acceleration followed by an increase of the computing time.

The optimized number of sweeps $Nswp_{opt}$ gives the best trade-off between number of time steps and computing time per time step: Too many linear iterations add computing time not compensated by the improvement in residual reduction. The value of $Nswp_{opt}$ can vary depending on the parameters ϵ and CFL but also on the number of grid points. For the default combination and with coarser grids it was found $Nswp_{opt}=3$, section V.B and V.C. For finer meshes the number of sweeps had to be increased to $Nswp_{opt}=10$, section V.C. In fact, an increase of the number of unknowns generates a larger condition number for the system matrix $\mathbf{M}$, Eq. (7), which causes slower convergence of the linear problem.

Figure 2. Effect of the LHS dissipation on the convergence rate for a fixed CFL number.

Figure 3. Effect of the Number of sweeps on the convergence wrt. iterations and time.

Table 1 gives two examples of residual reduction for 1 to 10 sweeps with the line-implicit LU-SGS solver and MG technique. They correspond to a RANS turbulence model computation on coarse meshes for M6 wing and NASA Common Research Model (CRM) grid level L1 wing-body, table 2. As we can see from both cases, the number of sweeps strongly affects the convergence rate. For the M6 wing we find a factor of 5 in residual reduction from 1 to 5 sweeps. However, as shown in figure 3, from a certain number of sweeps on, the convergence rate decreases very slowly. Therefore the number of sweeps should be kept below the “knee”. In terms of convergence rate $N_{swp}=5$ would seem to be good, table 1, but computing time considerations as in figure 3 favor $N_{swp}=3$, section V.B.

Table 1. Residual reduction in decades per 1000 iterations; M6 wing and DPW5 CRM L1 wing-body.

N_{swp}	1	2	3	5	8	10
M6	3.25	6.36	9.3	16.27	16.27	16.27
CRM L1	5	8.7	10	11.43	12	12

IV.C.3. Multigrid smoother

The other parameter studied, somewhat independent of the previous ones, is the type of smoother for each MG grid level. The explicit Runge-Kutta and LU-SGS smoothers can be independently applied from the finest to the coarsest grid levels. For 3D RANS cases, the LU-SGS smoother was used only on the finest grid level and the RK smoothing on the coarser meshes. Using LU-SGS smoothing for more levels required increased ϵ_{min} and smaller CFL_{max} numbers.

Our first hypothetical reason for this different configuration concerning the line-implicit method comes from the semicoarsening applied where the grid is stretched, used in the MG agglomeration process, figure 4 and decreasing strongly the aspect ratio. Previous works have also shown that implicitness on coarser grids does not pay off.¹³ Moreover the agglomeration process affects the node (and hence equation) ordering in the coarser meshes. The impact of the node ordering technique on the LU-SGS convergence was shown in a previous investigation¹⁰ which also indicated that the original ordering coming from the mesh generation was close to optimal. Merging the cells in the agglomeration process may give orderings which are bad for LU-SGS. 3D cases will be the most affected since the number of neighboring nodes is much higher than in 2D cases.

Figure 4. Multigrid agglomeration.

IV.C.4. Ordering

The ordering technique¹⁷ is in fact another variable of relevant impact on the convergence rate. The previous investigations,¹⁰ showed that the orderings from mesh generation techniques such as advancing front and Delaunay methods for unstructured and lexicographic for structured grids, are good. Attempts to devise better orderings did not succeed.¹⁰ For parallelization by domain partitioning, we use ordering methods offered by Edge based on a coloring process to reduce the matrix bandwidth and thus improve cache memory coherence. However, the line-implicit LU-SGS implementation¹⁰ does modify the node number ordering by starting with the lines, yet results in successful convergence rate accelerations.¹⁰ From this short presentation it follows that the ordering parameter will not be considered as an adaptive parameter but will be rather set to the original mesh ordering, reordered by lines for the line-implicit LU-SGS solver.

V. Numerical results

Numerical results are shown for 3D RANS computations using the line-implicit LU-SGS solver with the parameter investigation previously described. The simulations are computed for viscous transonic steady flow.

The first cases are for the ONERA M6 wing, figure 5, a unstructured stretched grid with 917K grid nodes. The grid has 30 prismatic layers, 1.2×10^6 prismatic cells and 1.8×10^6 tetrahedral cells. The maximum aspect ratio of the cells on the surface is 12×10^3 . The flow is characterized by a Reynolds number of $Re = 11.72 \times 10^6$ at $\alpha = 3.06^\circ$ angle of attack with a free stream Mach number of $M_\infty = 0.84$.

Figure 5. M6 wing surface mesh. Unstructured stretched grid with 917K grid nodes.

The second case is the CRM L4 wing-body from Boeing, figure 6, used in the fifth AIAA Drag Prediction Workshop (DPW5). It is a multi-block structured stretched grid with 17M grid points with 50 prismatic layers and a maximum aspect ratio of 131×10^3 . The flow conditions are $Re = 5 \times 10^6$ with a free stream Mach number of $M_\infty = 0.85$ and an angle of attack, $\alpha = 2.5^\circ$.

Figure 6. DPW5 CRM L4 surface mesh. Multi-block structured stretched grid with 17M grid points.

The last case is also a CRM wing-body coming from DLR. The difference from the previous model is the horizontal tail and the mesh is an unstructured stretched grid with 34M grid points, figure 7. The mesh has been generated with Solar and has been used in the fourth Drag Prediction Workshop (DPW4). It contains 38.6×10^6 tetrahedral elements, 72.7×10^3 prismatic elements and has a maximum aspect ratio of 15×10^3 . The flow conditions are the same for both CRM models.

Figure 7. DPW4 CRM fine surface mesh. Unstructured stretched grid with 34M grid points.

Table 2 summarizes the properties of the meshes mentioned through the paper.

Table 2. Data of the grids used for computations.

Mesh	No. of Prisms	No. of Tetrahedra	Max. aspect ratio	No. grid points
M6	1.2×10^6	1.8×10^6	12×10^3	917K
CRM				
L1 struct.	-	-	48×10^3	660K
L4 struct.	-	-	131×10^3	17M
Coarse	18.4×10^3	5.3×10^6	13.5×10^3	4M
Medium	34.2×10^3	14.3×10^6	15×10^3	11M
Fine	72.7×10^3	38.6×10^6	15×10^3	34M

The Spalart-Allmaras one-equation turbulence model¹⁸ is used for all the calculations.

V.A. CFL and LHS dissipation variation

In this section we show the changes in convergence rate by varying either ϵ , figure 8 left, or the CFL number, figure 8 right. As mentioned in section IV.C.1, the increase of the ϵ value deteriorates the convergence rate and here by a factor of the order of 2 going from $\epsilon = 0.2$ to $\epsilon = 1$. Concerning the CFL number, we see a considerable acceleration from $CFL = 10$ to $CFL = 100$. However bigger values may not provide much more speedup. As already explained in section IV.C.1, to increase the CFL number over the $CFL_{max} = 1000$ given by $\epsilon_{min} = 0.2$, ϵ needs to be increased. The effect coming from an increase of ϵ seems to lead the overall convergence which decelerates despite a bigger CFL number. The same results are observed in time.

Figure 8. Influence of ϵ and CFL on the residual convergence in iterations. RANS turbulence model computations on M6 wing. Left: ϵ variation with $CFL = 1000$. Right: CFL variation with $\epsilon = 0.2$. $N_{swp} = 1$.

V.B. Number of sweeps variation

In figure 9, different number of sweeps are applied for the default combination $\epsilon = 0.2$, $CFL = 1000$, section IV.C.1. In this case it is important to compare the convergence in time to define $N_{swp_{opt}}$. The comparisons in time show clearly that more than 3 sweeps increases the computing time. For finer meshes the same behavior is observed but with $N_{swp_{opt}} = 10$.

Figure 9. $Nswp$ variation for $CFL = 1000$ and $\epsilon = 0.2$. Influence on the residual convergence in iterations (left) and time (right). RANS turbulence model computations on M6 wing.

V.C. Accelerations achieved

The following figures give an idea of the order of acceleration achieved for 3D RANS computations using full multigrid W-cycles. The simulations are run on the M6 wing, figure 10, and the CRM wing-bodies, figures 11, 12, and 13 with the default configuration $CFL = 1000$ and $\epsilon = 0.2$ with $Nswp = 3$ and $Nswp = 10$ respectively. The LU-SGS MG smoother is only applied on the finest grid. A comparison is done between line-implicit LU-SGS, implicit LU-SGS (without lines), line-implicit and explicit Runge-Kutta. An acceleration in time at least of the order of two has been generally observed, figures 10, 11 and 12, with some variations depending on the magnitude of residual reduction and the mesh size, figure 13. Moreover in many cases as in figure 12, LU-SGS can converge to much smaller residuals than the Runge-Kutta solver. Comparisons between line-implicit LU-SGS and implicit LU-SGS, show the high efficiency of the combination with the line-implicit method for boundary layers. For example on the M6 wing, figure 10, the acceleration in time with line-implicit LU-SGS is of the order of 5 compared to implicit LU-SGS.

Figure 14 gives the corresponding normalized drag coefficient convergences of figures 10, 11 and 12, showing that for the LU-SGS solver, the drag coefficient converges faster to within 0.1% of its steady state value compared to Runge-Kutta. The convergence of the drag coefficient stabilizes first with line-implicit LU-SGS on the M6 wing and the CRM fine wing-body, up to twice faster than with implicit LU-SGS, figure 14(a), and up to three to four times faster than with Runge-Kutta, figure 14(b) and (a) respectively. These cases show also that the explicit Runge-Kutta solver can be more stable than its line-implicit approach as clearly shows figure 14(b) where the drag coefficient with line-implicit Runge-Kutta has still not converged to within 0.1% of its steady state value. Figure 14(b) shows in fact that the residual convergence does not provide any information about the convergence of the forces since whereas all the solvers have converged up to seven orders of magnitude residual reduction in figure 11, not all the drag coefficients have converged. For most of the cases the order of magnitude residual reduction for which the drag coefficient has become sufficiently stable is around five and six.

Figure 10. Comparison of density residual convergence. RANS turbulence model parallel computations on unstructured stretched grid for M6 wing with 24 processors. Residual convergence in iterations (left) and time on single processor (right). Parameter setting: $CFL = 1000$, $\epsilon = 0.2$ and $N_{swp} = 3$.

Figure 11. Comparison of density residual convergence. RANS turbulence model parallel computations on structured stretched grid for DPW5 CRM L4 wing-body with 384 processors. Residual convergence in iterations (left) and time on single processor (right). Parameter setting: $CFL = 1000$, $\epsilon = 0.2$ and $N_{swp} = 10$.

Figure 12. Comparison of density residual convergence. RANS turbulence model parallel computations on unstructured stretched grid for DPW4 CRM fine wing-body with 384 processors. Residual convergence in iterations (left) and time on single processor (right). Parameter setting: $CFL = 1000$, $\epsilon = 0.2$ and $N_{swp} = 10$.

Figure 13. Comparison of density residual convergence in time on single processor, for different mesh sizes of the DPW4 CRM wing-body. RANS turbulence model parallel computations with 96 (a), 240 (b) and 384 (c) processors. Parameter setting: $CFL = 1000$, $\epsilon = 0.2$ and $N_{swp} = 10$.

Figure 14. Convergence in time on single processor of normalized drag coefficient. Dash line: within the 0.1% of the steady state.

VI. Discussion

From these results it is clear that the LU-SGS solver requires proper parameter settings to obtain maximum efficiency. Whereas a good parameter configuration can give significant acceleration, improper configurations can also cause extremely slow convergence. The default parameter configuration provided gives the best convergence rate for the cases presented but may require some slight modifications for other cases following the description of the parameter influence. Hence, the comprehension of the impact of the parameter is as important as the definition of optimized settings. Concerning the CFL number, though larger values are recommended, the most important is to keep the CFL number equal or bigger than 100 without exceeding values of 10000 which may start decelerating the convergence, seen also for 2D cases.¹⁶ Another observation is that one should not expect the same level of convergence acceleration for all the cases even after tuning the parameters. For the convergence of the solution, it is not sufficient to monitor the convergence of the residuals but also the forces as the drag force. We have to note that for engineering purposes, about 6 orders of magnitude residual reduction is generally enough. The performance of the line-implicit LU-SGS solver will depend on the properties of the mesh, mainly in the boundary layer¹³ and the order of magnitude of residual reduction required. Other flow conditions than the ones presented here may allow or require larger latitude in choice of parameters. For example, stronger artificial dissipation - produced by increasing ϵ - is always more robust and may be required for strong shocks. On the other hand, low Reynolds number cases would allow $\epsilon = 0$.

VII. Conclusion

A parameter investigation has been carried out for convergence acceleration of the line-implicit LU-SGS solver for 3D turbulent RANS computations. The different parameters studied were presented showing their influence on the convergence as well as their interaction. For some cases mathematical expressions were found supported by numerical results. The main observations are that the minimization of the ϵ value combined with a large CFL number minimizes the computing time. Moreover the increase of the number of sweeps decreases the computing time until some optimized number of sweeps. Fortunately optimized default combinations could be defined, namely a CFL number of 1000 and an ϵ value of 0.2. The number of sweeps showed to be mesh size dependent with an optimized number of 3 for coarser meshes and of 10 for finer meshes. The LU-SGS multigrid smoother was applied only to the finest grid level for 3D RANS computations. The ordering is important but conclusive results on optimal orderings have not been found and require further research. The default parameter configuration has been applied to different cases to compare the convergence in iterations and time of the line-implicit LU-SGS solver with the implicit LU-SGS, line-implicit Runge-Kutta and explicit Runge-Kutta solvers. For most of the cases line-implicit LU-SGS has clearly shown relevant acceleration of the order of 50%. Further quantitative investigation is needed on the influence of grid properties and on applications to complete aircraft configurations with details such as gaps at control surfaces.

Acknowledgments

Financial support from FoT Flygteknik 10-12 Nationell Aerodesignmetodik is gratefully acknowledged.

References

- ¹Yoon, S. and Jameson, A., "Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes Equations," *AIAA Journal*, Vol. 26, No. 9, 1988, pp. 1025–1026.
- ²Jameson, A. and Yoon, S., "Lower-Upper Implicit Schemes with Multiple Grids for the Euler Equations," *AIAA Journal*, Vol. 25, No. 7, 1987, pp. 929–935.
- ³Soetrino, M., Imlay, S., Roberts, D., and Taffin, D., "Development of a 3-D Zonal Implicit Procedure for Hybrid Structured-Unstructured Grids," Aiaa paper 96–0167, Jan. 1996.
- ⁴Sharov, D. and Nakahashi, K., "Reordering of Hybrid Unstructured Grids for Lower-Upper Symmetric Gauss-Seidel Computations," *AIAA Journal*, Vol. 36, No. 3, 1998, pp. 484–486.
- ⁵Chen, R. F. and Wang, Z. J., "Fast, Block Lower-Upper Symmetric Gauss-Seidel Scheme for Arbitrary Grids," *AIAA Journal*, Vol. 38, No. 12, 2000, pp. 2238–2245.
- ⁶KIM, J. S. and KWON, O. J., "An Efficient Implementation of Implicit Operator for Block LU-SGS Method," *Computational Fluid Dynamics*, Vol. 13, No. 2, 2005, pp. 154–159.

- ⁷Dwight, R. P., *Efficiency Improvements of RANS-Based Analysis and Optimization using Implicit and Adjoint Methods on Unstructured Grids*, Ph.D. thesis, School of Mathematics, University of Manchester, 2006.
- ⁸Sharov, D., Luo, H., Baum, J. D., and Löhner, R., “Implementation of Unstructured Grid GMRES+LU-SGS Method on Shared-Memory, Cache-based Parallel Computers,” Aiaa paper 2000-0927, 2000.
- ⁹Nejat, A. and Ollivier-Gooch, C., “A High-Order Accurate Unstructured GMRES Algorithm for Inviscid Compressible Flows,” Aiaa paper 2005-5341, June 2005.
- ¹⁰Otero, E. and Eliasson, P., “Improving the Performance of the CFD code Edge using LU-SGS and Line-implicit Methods,” *Proceedings of the 4:th CEAS Conference in Linköping, 2013*, Linköping University Electronic Press, Linköping, Sweden, 2013, pp. 124–134.
- ¹¹“Edge, Theoretical Formulation,” Tech. Rep. 03-2870, FOI, 2010.
- ¹²Brandt, A., “Multi-Level Adaptive Solutions to Boundary-Value Problems,” *Mathematics of Computation*, Vol. 31, No. 138, 1977, pp. 333–390.
- ¹³Eliasson, P., Weinerfelt, P., and Nordström, J., “Application of a Line-implicit Scheme on Stretched Unstructured Grids,” Aiaa paper 2009-163, Jan. 2009.
- ¹⁴Langer, S. and Li, D., “Application of Point Implicit Runge-Kutta Methods to Inviscid and Laminar Flow Problems using AUSM and AUSM⁺ Upwinding,” *International Journal of Computational Fluid Dynamics*, Vol. 25, No. 5, 2011, pp. 255–269.
- ¹⁵Langer, S., “Application of a Line Implicit Method to Fully Coupled System of Equations for Turbulent Flow Problems,” *International Journal of Computational Fluid Dynamics*, Vol. 27, No. 3, 2013, pp. 131–150.
- ¹⁶Otero, E. and Eliasson, P., “Convergence Acceleration of the CFD code Edge by LU-SGS,” *3rd CEAS Air&Space Conference*, Venice, Italy, 2011, pp. 606–613.
- ¹⁷Sharov, D. and Nakahashi, K., “Reordering of 3-D Hybrid Unstructured Grids For Vectorized LU-SGS Navier-Stokes Computations,” Aiaa paper 97-2102, 1997.
- ¹⁸Spalart, P. R. and Allmaras, S. R., “A one-equation turbulence model for aerodynamic flows,” Aiaa paper 1992-0439, 1992.