A99-16759

AIAA 99-0935

AN IMPROVED LU-SGS SCHEME WITH FASTER CONVERGENCE FOR UNSTRUCTURED GRIDS OF ARBITRARY TOPOLOGY

R. F. Chen and Z. J. Wang
CFD Research Corporation
Huntsville, Alabama

37th AIAA Aerospace Sciences Meeting and Exhibit
January 11-14, 1999 / Reno, NV

AIAA-99-0935

AN IMPROVED LU-SGS SCHEME WITH FASTER CONVERGENCE FOR UNSTRUCTURED GRIDS OF ARBITRARY TOPOLOGY

R. F. Chen* and Z. J. Wang†
CFD Research Corporation
Huntsville, Alabama

ABSTRACT

An improved implicit Lower-Upper Symmetric Gauss-Seidel (LU-SGS) approximate factorization scheme is developed and implemented for unstructured grids of arbitrary topology including viscous adaptive Cartesian grids. CPU times and memory requirements for the improved LU-SGS, the original LU-SGS, and a Fully Implicit Scheme (FIS) with a preconditioned CGS solver for several test examples are compared. Computational results showed that the improved LU-SGS requires about 20 to 30 percent more memory than the original LU-SGS, but is more robust and converges several times faster than the original LU-SGS. The improved LU-SGS has a convergence rate competitive to the FIS, while requiring much less memory.

INTRODUCTION

The difficulty of generating structured grids and the desire to compute flows over complex geometries spawned a surge of activity in the area of unstructured grids during the last decade. Unstructured grids provide flexibility in tackling complex geometries and for adapting to flow features. Types of unstructured grids include classical triangular or tetrahedral grids[1-5], quadrilateral or hexahedral grids[6], prismatic grids[7], or mixed grids[8]. Tetrahedral grids are the easiest to generate. However, experiences have indicated that they are not as efficient as prismatic or hexahedral grids for viscous boundary layers. On the other hand, prismatic and hexahedral grids are more difficult to generate than tetrahedral grids. Many CFD researchers have come to the conclusion that mixed grids (or hybrid grids) are the way to go. For example, a hybrid tetrahedral/prismatic grid approach[9] was successfully demonstrated for complex geometries. One disadvantage of tetrahedral grids is that tetrahedra are not as efficient as Cartesian cells in filling 3D space given a certain grid resolution. This can be easily understood with the fact that at least five tetrahedra are required to fill a cube without adding any new grid points. Therefore, it seems the most appealing grid topology is a hybrid of Cartesian and viscous layer grids. One example is the viscous Cartesian grid method developed by Wang[10]. This grid generation methodology has the potential of fully automating the grid generation task for complex geometries.

To handle a viscous Cartesian grid, a flow solver capable of handling unstructured grids of arbitrary topology needs to be developed. Furthermore, an implicit time-integration scheme is highly desired for improved efficiency. Many implicit schemes have been developed and applied to unstructured grids to accelerate convergence to steady state successfully[11-15]. Among them, the most commonly used technique is to linearize the residual and form a linearized system of equations whose left hand side corresponds to a lower order approximation of the spatial discretization of the right hand side. The solution of the flow field is then advanced one time step by approximately solving the resulting large sparse linear system. Lower order approximations are used in the left hand side due to storage considerations, computational complexity and the fact that the resulting lower order linear system is better conditioned than the higher-order one. The mismatch between the left hand side matrices and the right hand side discretization, however, results in a sub-optimum convergence rate to steady state. Iterative methods such as GMRES and CGS with an appropriate preconditioner are often used to approximately solve the sparse linear system due to the enormous computational cost and the large memory requirement of direct methods. The memory requirements per grid cell for such implicit technique depend

*Research Engineer, rfc@cfdrc.com
†Group Leader, Member AIAA

on the discretization stencil. For unstructured grids with arbitrary polyhedral cells, the required memory can be enormous due to large stencils used compared to structured grids. Recently, a matrix-free Newton-Krylov method is attracting attentions because it does not need to form the matrices explicitly. Large matrices, however, are still needed for preconditioning purposes[16,17].

Another very attractive implicit scheme, the implicit LU-SGS approximation factorization scheme, which was originally developed for structured grids[18], has been extended and applied to hybrid structured/unstructured grids and tetrahedra/prism unstructured grids[19,20]. Unstructured-grid-based LU-SGS schemes have demonstrated performance similar to that on structured grids. With LU-SGS, a special first order approximation is employed in discretizing the left hand side resulting in the reduction of the block diagonal matrices to diagonal matrices. As a result, the LU-SGS scheme does not require any extra storage compared to explicit methods, and is free from any matrix inversion. All the off-diagonal matrices still contribute to the solution through one forward and one backward sweep of Gauss-Seidel iteration, thus drastically improving efficiency over an explicit scheme. The special first order approximation used in deriving LU-SGS, does degrade convergence, especially after several orders of convergence. As it is indicated in the numerical examples in this study, LU-SGS is not competitive to a Fully Implicit Scheme (FIS) in term of CPU time.

In this paper, an improved LU-SGS(ILU-SGS) scheme is developed. The ILU-SGS is capable of achieving comparable convergence rate with a FIS, but requires much less memory than the FIS. The idea is to retain the block diagonal matrices, but employ LU-SGS-like backward and forward Gauss-Seidel iterations. In the following sections, the implicit finite volume discretization of Navier-Stokes equations is given first. Then the LU-SGS scheme on unstructured grids is presented, followed by the derivation of the ILU-SGS scheme. Several demonstration cases are presented to showcase the performance of the ILU-SGS. Finally the conclusions of the study are given.

IMPLICIT FINITE VOLUME DISCRETIZATION ON UNSTRUCTURED GRIDS

The governing equations for compressible viscous flow can be written in integral form over a control volume V as:

$$\int_V \frac{\partial Q}{\partial t} dV + \int_{\partial V} (F - F_v) dS = 0 \qquad (1)$$

where Q is the vector of conserved variables, F and F_v comprise the inviscid and viscous flux vectors respectively. Spatial discretization of equation (1) in grid cell i gives:

$$V_i \frac{\partial Q_i}{\partial t} + \sum_{j \in N(i)} (\tilde{F}_{ij} - \tilde{F}_{v,ij}) S_{ij} = 0 \qquad (2)$$

where V_i is the volume of cell i. Q_i is the cell averaged vector of conserved variables. Equation (2) is still true for Q_i evaluated on the cell center of cell i by neglecting a second order temporal term. As a matter of fact the time accuracy is not important because only steady flow is considered in this study. Hereafter let Q_i be the vector of conserved variables evaluated at the cell center of cell i. $N(i)$ is the set of neighbor cells of cell i. S_{ij} is the area of the cell face between cell i and cell j. $\tilde{F}$ and $\tilde{F}_v$ are the second order accurate numerical inviscid and viscous flux vectors at the face respectively. The numerical inviscid flux vector $\tilde{F}_{ij}$ is computed using Roe's approximate Riemann solver[21] with reconstructed variables at both sides of the face. A least squares linear reconstruction scheme of the primitive variables is used. Venkatarkrishnan's limiter[22] is employed to make the scheme monotone. Other details of the flow solver are contained in Reference 23. Applying backward Euler scheme for time integration in equation (2), we obtain:

$$\frac{V_i}{\Delta t_i}(Q_i^{n+1} - Q_i^n) + \sum_{j \in Ni} (\tilde{F}_{ij}^{n+1} - \tilde{F}_{v,ij}^{n+1}) S_{ij} = 0 \qquad (3)$$

where Δt_i is the local time step. Equation (3) can be rewritten in the following delta form:

$$\frac{V_i}{\Delta t_i}\Delta Q_i^n + \sum_{j \in N(i)} (\Delta\tilde{F}_{ij}^n - \Delta\tilde{F}_{v,ij}^n) S_{ij} = \mathrm{Res}_i^n \qquad (4)$$

where Δ is the forward difference in time, for example,

$$\Delta Q_i^n = Q_i^{n+1} - Q_i^n \tag{5}$$

and the right hand side residual can be written as:

$$\mathrm{Res}_i^n = -\sum_{j \in N(i)} (\tilde{F}_{ij}^n - \tilde{F}_{v,ij}^n) S_{ij} \tag{6}$$

In practice, $\Delta\tilde{F}$ and $\Delta\tilde{F}_v$ are usually approximated by their first order counterparts $\Delta\bar{\bar{F}}$ and $\Delta\bar{\bar{F}}_v$ respectively, and equation (4) becomes

$$\frac{V_i}{\Delta t_i}\Delta Q_i^n + \sum_{j \in N(i)} \left(\Delta\bar{\bar{F}}_{ij}^n - \Delta\bar{\bar{F}}_{v,ij}^n\right) S_{ij} = \mathrm{Res}_i^n \tag{7}$$

Note that

$$\begin{aligned} &\Delta\bar{\bar{F}}_{ij}^n - \Delta\bar{\bar{F}}_{v,ij}^n \\ &= \left[\bar{\bar{F}}(Q_i^{n+1}, Q_j^{n+1}) - \bar{\bar{F}}(Q_i^n, Q_j^{n+1})\right] \\ &+ \left[\bar{\bar{F}}(Q_i^n, Q_j^{n+1}) - \bar{\bar{F}}(Q_i^n, Q_j^n)\right] \\ &- \left[\bar{\bar{F}}_v(Q_i^{n+1}, Q_j^{n+1}) - \bar{\bar{F}}_v(Q_i^n, Q_j^{n+1})\right] \\ &- \left[\bar{\bar{F}}_v(Q_i^n, Q_j^{n+1}) - \bar{\bar{F}}_v(Q_i^n, Q_j^n)\right] \end{aligned} \tag{8}$$

Linearizing the first and the third terms in the right hand side of equation (8), we have

$$\bar{\bar{F}}(Q_i^{n+1}, Q_j^{n+1}) - \bar{\bar{F}}(Q_i^n, Q_j^{n+1}) \approx \frac{\partial\bar{\bar{F}}_{ij}}{\partial Q_i}\Delta Q_i^n \tag{9}$$

$$\bar{\bar{F}}_v(Q_i^{n+1}, Q_j^{n+1}) - \bar{\bar{F}}_v(Q_i^n, Q_j^{n+1}) \approx \frac{\partial\bar{\bar{F}}_{v,ij}}{\partial Q_i}\Delta Q_i^n \tag{10}$$

Substituting equations (8), (9) and (10) back to equation (7), we obtain:

$$\begin{aligned} &D\Delta Q_i^n \\ &+ \sum_{j \in N(i)} \left[\bar{\bar{F}}(Q_i^n, Q_j^n + \Delta Q_j^n) - \bar{\bar{F}}(Q_i^n, Q_j^n)\right] S_{ij} \\ &- \sum_{j \in N(i)} \left[\bar{\bar{F}}_v(Q_i^n, Q_j^n + \Delta Q_j^n) - \bar{\bar{F}}_v(Q_i^n, Q_j^n)\right] S_{ij} \\ &= \mathrm{Res}_i^n \end{aligned} \tag{11}$$

where matrix D is given by

$$D = \frac{V_i}{\Delta t_i} I + \sum_{j \in N(i)} \left(\frac{\partial\bar{\bar{F}}_{ij}}{\partial Q_i} - \frac{\partial\bar{\bar{F}}_{v,ij}}{\partial Q_i}\right) S_{ij} \tag{12}$$

where I is the identity matrix.

ORIGINAL LU-SGS

In the original LU-SGS approach, the first order numerical flux vectors in the left hand side of (11) are chosen as

$$\bar{\bar{F}}_{ij} - \bar{\bar{F}}_{v,ij} = \frac{1}{2}\{F_i + F_j - \lambda_{ij}(Q_j - Q_i)\} \tag{13}$$

where λ_{ij} is the spectral radius of the flux Jacobian matrix at the cell face:

$$\lambda = |V \cdot n| + a + \frac{2(\mu + \mu_t)}{\rho|n \cdot (r_j - r_i)|} \tag{14}$$

where n is the normal of the cell face pointing from cell i to cell j. r_i and r_j are the position vectors of cell centers of cell i and cell j respectively, V is the velocity vector, ρ is the density, a is the speed of sound, μ and μ_t are kinematic and turbulent viscosity respectively.

Since each control volume is closed, for cell i we have

$$\sum_{j \in N(i)} \frac{\partial F_i}{\partial Q_i} S_{ij} = 0 \tag{15}$$

Substituting equation (13) into equation (12) and using equations (14) and (15), we obtain

$$D = \left(\frac{V_i}{\Delta t} + \frac{1}{2}\sum_{j \in N(i)} \lambda_{ij} S_{ij}\right) I \tag{16}$$

which is reduced to the identity matrix with a scale factor.

Equation (11) is then solved using one sweep of symmetric Gauss-Seidel iteration as shown in the following:

Forward sweep:

$$D\Delta Q_i^* = \mathrm{Res}_i^n - 0.5 \cdot \sum \left[(F(Q_j^n + \Delta Q_j^*) - F(Q_j^n)) - \lambda_{ij}\Delta Q_j^*\right] S_{ij} \tag{17}$$

Backward sweep:

$$\Delta Q_i = \Delta Q_i^* - 0.5 D^{-1} \cdot \sum_{j \in U(i)} \left\{[F(Q_j^n + \Delta Q_j) - F(Q_j^n)] - \lambda_{ij}\Delta Q_j\right\} S_{ij} \tag{18}$$

where $L(i)$ and $U(i)$ represent the lower and upper neighbor cells of cell i according to the cell ordering.

IMPROVED LU-SGS

Unlike the original LU-SGS where the special first order accurate numerical flux vectors of equation (13) are used, the first order numerical flux vectors which are consistent with the second order flux vectors in the spatial discretization are used in the left hand side discretization for the improved LU-SGS. In this study, for example, Roe's approximate Riemann solver with variable values on the cell centers at both sides of the face is used for the numerical inviscid flux vector in the left hand side of equation (11). Equation (11) is also solved using symmetric Gauss-Seidel iterations. Unlike the original LU-SGS, multiple number of inner iterations are possible. It depends on a prescribed convergence tolerance and a maximum number of sweeps.

To be more specific, given the solutions $\Delta Q^{(k-1)}$ at sweep level k-1, we compute the solution at the kth sweep using the followings:

Forward sweep:

$$D\Delta Q_i^* + \sum_{j \in L(i)} \left[\tilde{F}(Q_i^n, Q_j^n + \Delta Q_j^*) - \tilde{F}(Q_i^n, Q_j^n) - \tilde{F}_v(Q_i^n, Q_j^n + \Delta Q_j^*) - \tilde{F}_v(Q_i^n, Q_j^n)\right] S_{ij} + \sum_{j \in U(i)} \left[\tilde{F}(Q_i^n, Q_j^n + \Delta Q_j^{(k-1)}) - \tilde{F}(Q_i^n, Q_j^n) - \tilde{F}_v(Q_i^n, Q_j^n + \Delta Q_j^{(k-1)}) + \tilde{F}_v(Q_i^n, Q_j^n)\right] S_{ij} = \mathrm{Res}_i^n \tag{19}$$

Backward sweep:

$$D\Delta Q_i^{(k)} + \sum_{j \in L(i)} \left[\tilde{F}(Q_i^n, Q_j^n + \Delta Q_j^*) - \tilde{F}(Q_i^n, Q_j^n) - \tilde{F}_v(Q_i^n, Q_j^n + \Delta Q_j^*) + \tilde{F}_v(Q_i^n, Q_j^n)\right] S_{ij} + \sum_{j \in U(i)} \left[\tilde{F}(Q_i^n, Q_j^n + \Delta Q_j^{(k)}) - \tilde{F}(Q_i^n, Q_j^n) - \tilde{F}_v(Q_i^n, Q_j^n + \Delta Q_j^{(k)}) + \tilde{F}_v(Q_i^n, Q_j^n)\right] S_{ij} = \mathrm{Res}_i^n \tag{20}$$

where the initial value for the inner sweep $\Delta Q^{(0)} = 0$. The number of inner sweeps is controlled by the conditions,

$$\left\|\Delta Q^{(k)} - \Delta Q^{(k-1)}\right\| / \left\|\Delta Q^{(1)}\right\| \le \varepsilon \tag{21}$$

$$k \le K_{\mathrm{MAX}} \tag{22}$$

where ε is the convergence tolerance and K_{MAX} a prescribed maximum number of inner sweeps. The flow variables at time level $n+1$ are updated by

$$Q^{n+1} = Q^n + \Delta Q^{(k)} \tag{23}$$

Note that in this approach, matrix D is no longer an identity matrix with a scale factor. Instead it is a 5×5 matrix in three-dimension and a 4×4 matrix in two-dimension. Therefore the extra memory is required to store the block diagonal matrix in the ILU-SGS approach. Equations (19) and (20) are then solved exactly with the LU decomposition method.

FULLY IMPLICIT SCHEME

In a fully implicit approach, the off-diagonal terms in the left hand side of equation (11) are also linearized. After the linearization, equation (11) reduces to the following large sparse linear system:

$$D\Delta Q_i^n + \sum_{j \in N(i)} \left(\frac{\partial \tilde{F}_{ij}}{\partial Q_j} - \frac{\partial \tilde{F}_{v,ij}}{\partial Q_j} \right) S_{ij} \Delta Q_j^n = \mathrm{Res}_i^n \tag{24}$$

The linear system is usually solved with an iterative method such as GMRES and CGS with proper preconditioning. The number of inner sweep in solving equation (24) is also controlled by a prescribed convergence tolerance and a maximum number of sweeps. Equation (24) requires the storage of not only the diagonal block matrix, but also off-diagonal block matrices which depend on the number of neighboring cells. Therefore the storage required by the fully implicit scheme is much more than that of the original and improved LU-SGS depending on the local discretization stencil. In this study, equation (24) is solved with a CGS solver with a block incomplete LU (ILU) preconditioner.

NUMERICAL RESULTS

Several test cases are simulated to compare the CPU times and memory requirements of LU-SGS, the ILU-SGS, and the FIS.

Transonic Flow over NACA 0012 Airfoil

Transonic flow over NACA 0012 airfoil with a free stream Mach number of 0.85 and an angle of attack 1^0 is chosen as the first test case. The CFL number for the ILU-SGS and the FIS increases linearly from 5 at the first step to 200 after 40 time steps. The CFL for the original LU-SGS is chosen as 10^{40}. The tolerance ε and the maximum inner sweeps are 0.1 and 10 respectively for both the ILU-SGS and the FIS. Figure 1 shows the adaptive Cartesian/Quad unstructured grid around the airfoil. Figure 2 and 3 show the percentage comparison of storage requirements and the comparison of convergence histories versus work units. The ILU-SGS requires about 35 percent more memory than the original LU-SGS. The ILU-SGS, however, converges several times faster than the original LU-SGS. Theses figures also show that the ILU-SGS requires less computational time than the FIS and half the memory required by the FIS.

Inviscid Flow Over A 3D Blunt Body

Inviscid flow over a three-dimensional blunt body with a free stream Mach number of 0.3 and an angle of attack 0^0 is chosen as another test case. The CFL numbers for the ILU-SGS and the FIS increase linearly from 10 at the first step to 200 after 100 time steps. The CFL number for the original LU-SGS is again chosen as 10^{40}. The tolerance ε and the maximum number of inner sweeps are set to 0.01 and 40 respectively for both the ILU-SGS and the FIS. Figure 4 shows the adaptive Cartesian/prism unstructured grid around the body. Figures 5 and 6 present the percentage comparison of storage requirements and the comparison of convergence histories versus work units. It is shown that the ILU-SGS requires about 24 percent more memory than the original. The FIS requires twice more memory than the ILU-SGS. Note that the residual with the original LU-SGS does not drop to machine zero. The ILU-SGS performs nearly as well as the FIS in terms of CPU time.

Supersonic Turbulent Flow Over A 3D Forebody

Supersonic turbulent flow over a forebody with an analytically given shape is chosen as another test for the ILU-SGS. Figure 7 shows the surface of the forebody and a viscous Cartesian grid around the forebody. The flow conditions are: free stream Mach number $M_\infty = 1.7$, angle of attack $\alpha = -5^0$, angle of sideslip $\beta = -0.04^0$ and the Reynolds number based on the free stream flow properties and body characteristic length $Re = 2.33 \times 10^6$. In the computation, the $k-\varepsilon$ turbulence model with a wall function is used. The analytical description of the forebody and experimental data can

be found in Reference 24.

In this test it was found that the simulation using the original LU-SGS from the free stream condition could not proceed even with a CFL number as small as 0.5. In order to make comparisons of the improved and the original LU-SGS, we first ran the simulation using the ILU-SGS from free stream with CFL=5 for 20 steps. Then both LU-SGS and ILU-SGS restarted from this solution. A CFL number of 20 was chosen for both methods. It was found that CFL larger than 20 caused instability for the original LU-SGS in the simulation. The maximum inner sweep number is chosen to 1 in the ILU-SGS for more accurate comparison. Venkatakrishnan's limiter was used for both methods. The ILU-SGS required about 27 percent more memory than the original LU-SGS. Figure 7 and Figure 8 show that residual history versus iterations and work units respectively. It is seen that the ILU-SGS requires about one third the iterations and half the computational time of the original LU-SGS. More significantly, the ILU-SGS can start the simulation from the free stream, which indicates that it is more robust than the original LU-SGS. Figure 9 to Figure 12 show the comparison of surface pressure coefficients between the computation and experimental data in different locations. There are some wiggles in the numerical solutions. This is because the unstructured grid points on the body do not exactly line up with the given planes where experimental data are available. The overall numerical pressure prediction, however, agrees very well with the experimental data.

CONCLUSIONS

An improved LU-SGS with increase efficiency and robustness is developed. The improved LU-SGS is significantly more efficient and robust than the original LU-SGS, while requiring only 20 to 30 percent more memory. The convergence rate of the improved LU-SGS can compete with the fully implicit scheme while requiring much less memory than the fully implicit scheme.

ACKNOWLEDGMENTS

The authors would like to thank Drs. Andrzej Przekwas and Ashok Singhal of CFD Research Corporation for many helpful discussions. We gratefully acknowledge the financial support from NSF.

REFERENCES

1. Barth, T.J., and Frederickson, P.O., "High-Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction," AIAA Paper 90-0013, 1990.

2. Rausch, R.D., Batina, J.T. and Yang, H.T., "Spatial Adaptation Procedures on Unstructured Meshes for Accurate Unsteady Aerodynamic Flow Computation," AIAA Paper 91-1106, 1991.

3. Aftosmis, M., Gaitonde, D., and Tavares, S., "On the Accuracy, Stability and Monotonicity of Various Reconstruction Algorithms for Unstructured Meshes," AIAA Paper 94-0415, 1994.

4. Venkatakrishnan, V., "A Perspective on Unstructured Grid Flow Solvers," AIAA Paper 95-0667, 1995.

5. Marvriplis, D.V., "Unstructured Mesh Generation and Adaptivity," ICASE Report No. 95-26, 1995.

6. Schneiders, R., "Automatic Generation of Hexahedral Finite Element Meshes," in Proceedings of 4th International Meshing Roundtable'95, October 1995, Albuquerque, NM.

7. Nakahashi, K., "Adaptive Prismatic Grid Method for External Viscous Flow Computations," AIAA Paper 93-3314-CP, 1993.

8. Coirier, W.J., and Jorgenson, P.C.E., "A Mixed Volume Grid Approach for the Euler and Navier-Stokes Equations," AIAA Paper 96-0762, 1996.

9. Kallinderis, Y., Khawaja, A. and McMorris, H., "Hybrid Prismatic/Tetrahedral Grid Generation for Complex Geometries," AIAA Paper 95-0210, 1995.

10. Wang, Z.J., "An Automated Viscous Adaptive Cartesian Grid Generation Method for Complex geometries," in Proceeding of 6th International Conference on Numerical Grid Generation in Computational Field Simulations, Ed. Cross, M., *et al.*, Univ. Greenwich, UK., 1998.

11. Venkatakrishnan, V. and Mavriplis, D.J., "Implicit Solvers for Unstructured Meshes," AIAA Paper 91-1537-CP, 1991.

12. Venkatakrishnan, V. "Implicit Schemes and Parallel Computing in Unstructured Grid CFD," ICASE report No. 95-28, 1995.

13. Soetrisno, M., Imlay, S.T., and Roberts, D.W., "A Zonal Implicit Procedure for Hybrid Structured-Unstructured Grids," AIAA Paper 94-0645, 1994.

14. Frink,N.T., "Assessment of an Unstructured-Grid Method for Predicting 3-D Turbulent Viscous Flows," AIAA Paper 96-0292, 1996.

15. Blanco, M., and Zingg, D.W., "A Fast Solver for the Euler Equations on Unstructured Grids Using a Newton-GMRES Method," AIAA Paper 97-0331, 1997.

16. Tidriri, M.D. "Krylov Methods for Compressible Flows," ICASE report No. 95-48, 1995.

17. Meister, A., "Comparison of Different Krylov Subspace Methods Embedded in an Implicit Finite Volume Scheme for the Computation of Viscous and Inviscid Flow Fields on Unstructured Grids," Journal of Computational Physics, Vol. 140, pp. 311-345, 1998.

18. Jamson, A. and Yoon, S., "Lower-Upper Implicit Schemes with Multiple Grids for the Euler Equations," AIAA Journal 25, No. 7, pp. 929-935, 1987.

19. Sharov, D. and Nakahashi, K. "Reordering of 3D Hybrid Unstructured Grids for Vectorized LU-SGS Navier-Stokes Computations," AIAA paper 97-2102, 1997.

20. Sharov, D. "Low Speed Preconditioning and LU-SGS Scheme for 3D Viscous Flow Computations on Unstructured Grids," AIAA paper 98-0614, 1998.

21. Roe, P.L, "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," J. Computational Physics, vol. 43, 1981.

22. Venkatakrishnan, V. "on the Accuracy of Limiters and Convergence to Steady State Solutions," AIAA paper 93-0880, 1993.

23. Wang, Z.J., "A Fast Nested Multi-Grid Viscous Flow Solver for Adaptive Cartesian/Quad Grids," AIAA Paper 96-2091, 1996.

24. Townsend, J.C., Howell, D.T., Collins, I.K. and Hayes, C., "Surface Pressure Data on a Series of Analytic Forebodies at Mach Number From 1.70 to 4.50 and Combined Angles of Attack and Sideslip," NASA Technical Memorandum 80062, 1979.

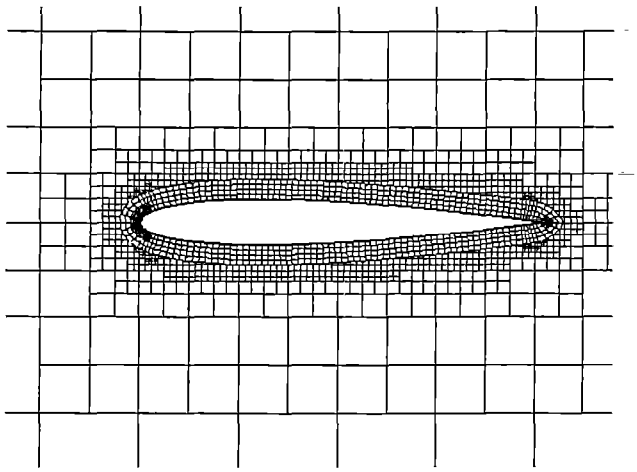


Figure 1. Adaptive Cartesian/Quad Unstructured Grid around the NACA 0012 Airfoil

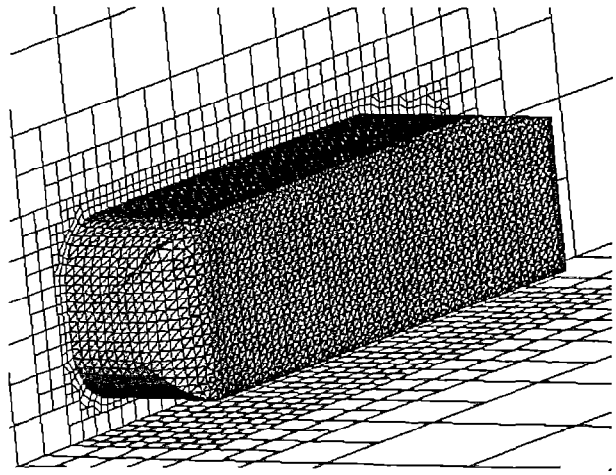


Figure 4. Adaptive Cartesian/Prism Unstructured Grid around the 3D Blunt Body

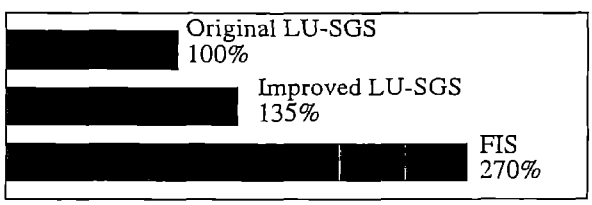


Figure 2. Percentage Comparison of Memory Requirements for the Simulations of the Transonic Flow Over the NACA 0012 Airfoil

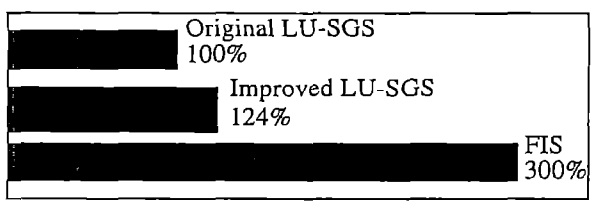


Figure 5. Percentage Comparison of Memory Requirements for the Simulations of the Inviscid Flow Over the 3D Blunt Body

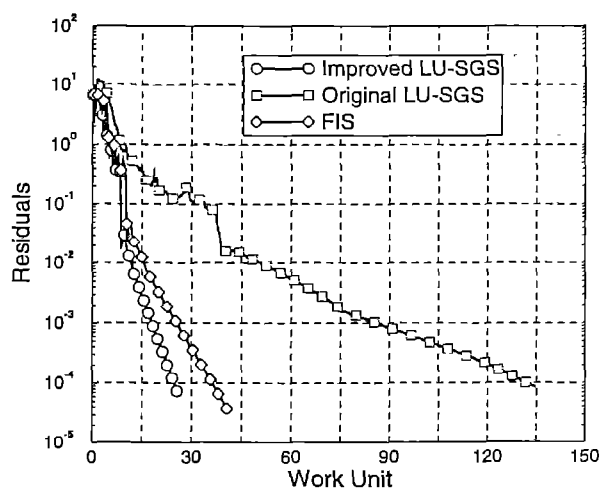


Figure 3. Convergence Histories versus Work Units for the Simulations of the Transonic Flow Over the NACA 0012 Airfoil

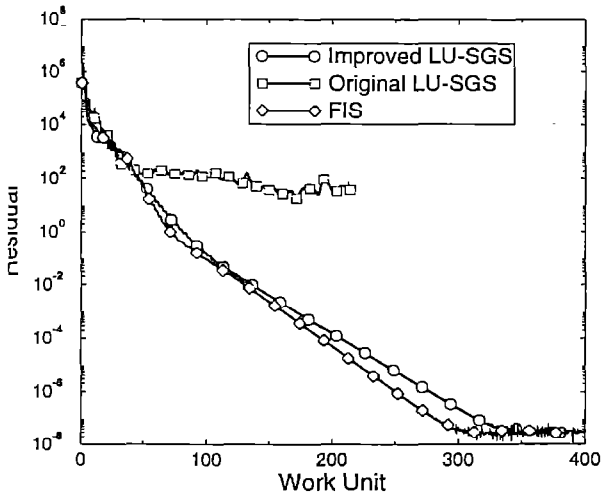


Figure 6. Convergence Histories versus Work Units for the Simulations of the Inviscid Flow Over the 3D Blunt Body

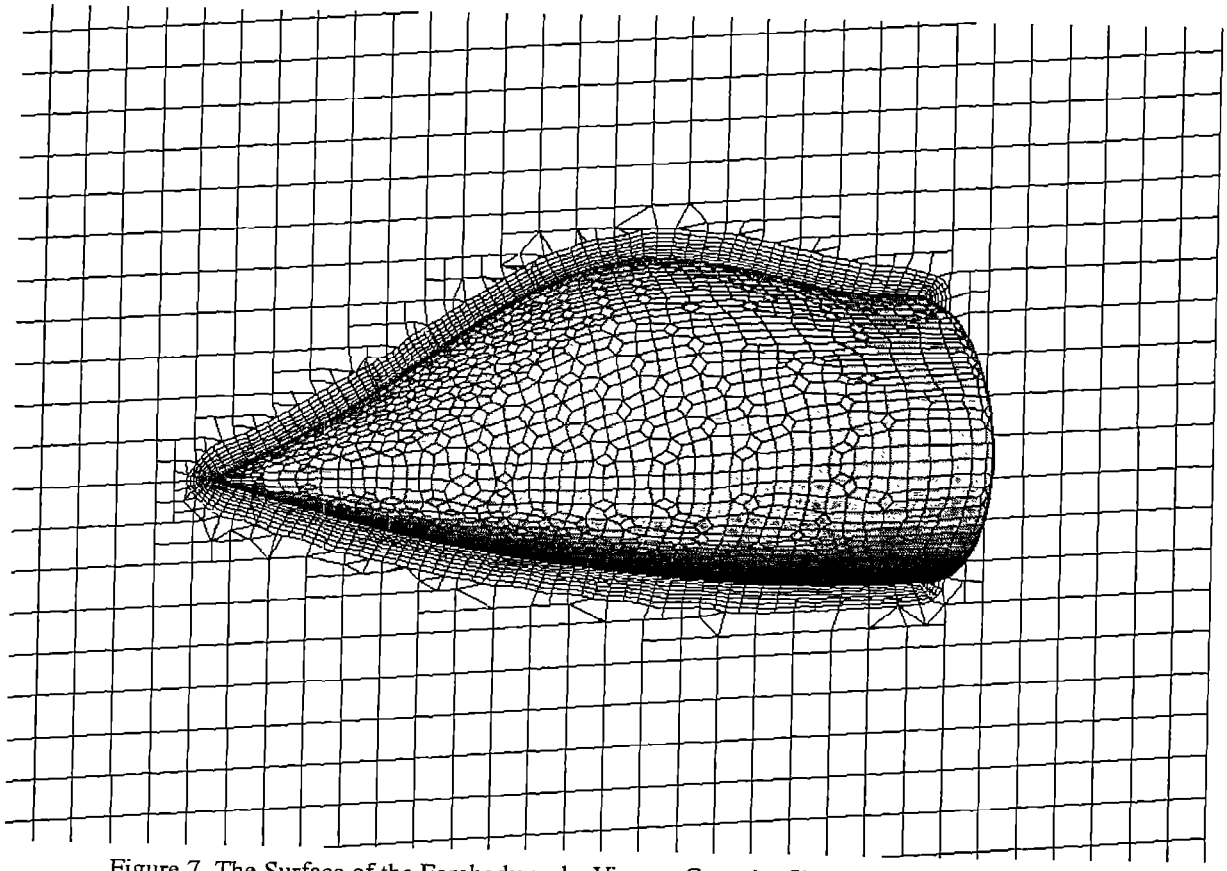


Figure 7. The Surface of the Forebody and a Viscous Cartesian Unstructured Grid Around It

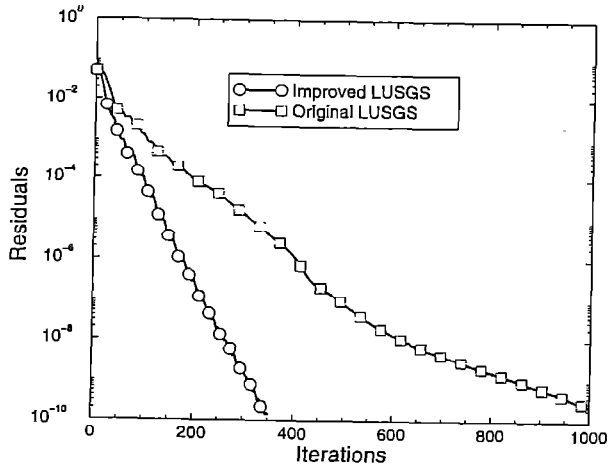


Figure 8. Convergence Histories versus Iterations for the Simulations of the Supersonic Turbulent Flow Over the Forebody

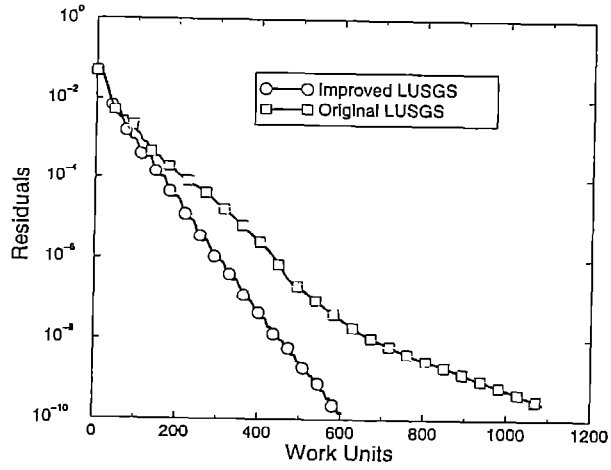
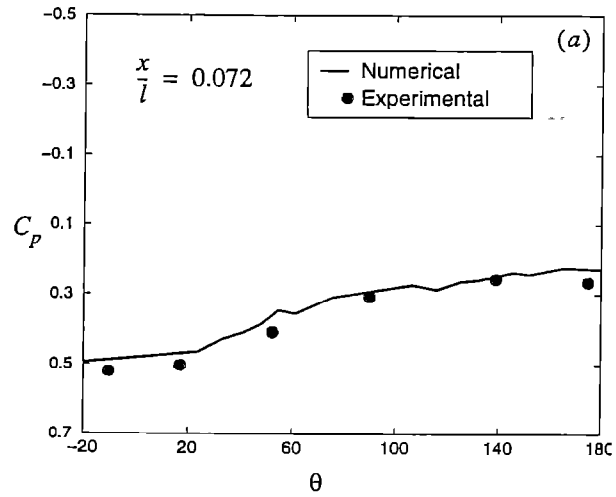


Figure 9. Convergence Histories versus Work Units for the Simulations of the Supersonic Turbulent Flow Over the Forebody



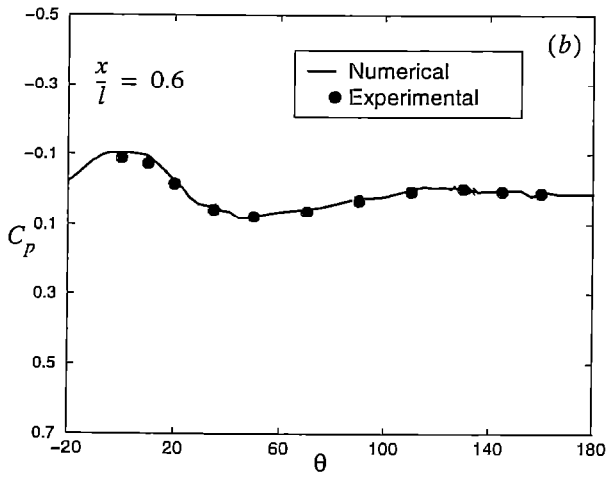
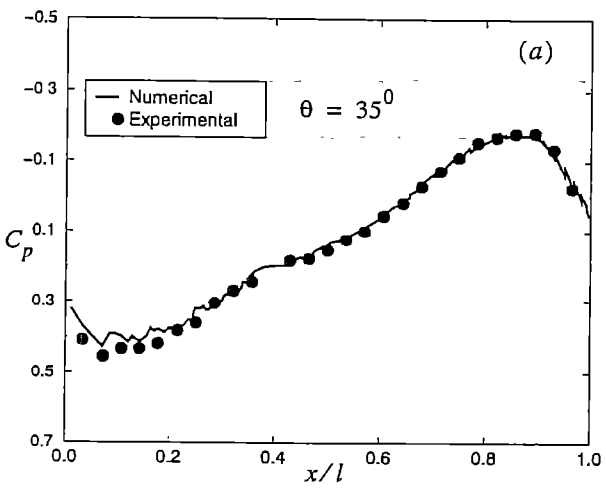


Figure 10. Comparison of Azimuthal Variation of the Surface Pressure Coefficients Computed using the Improved LU-SGS with the Experimental Data at Different Axial Locations: (a) $x/l = 0.072$, (b) $x/l = 0.6$



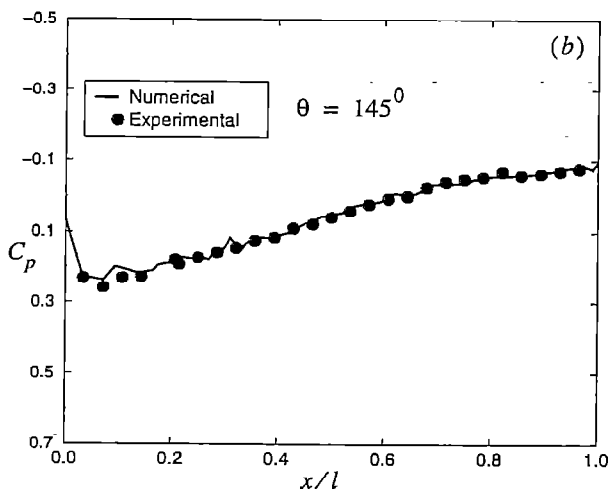


Figure 11. Comparison of Longitudinal Variation of the Surface Pressure Coefficients Computed using the Improved LU-SGS with the Experimental Data at Different Azimuthal Angle: (a) $\theta = 35^0$, (b) $\theta = 145^0$