[8]Cebeci, T., and Smith, A. M. O., *Analysis of Turbulent Boundary Layers*, Academic, New York, 1974, pp. 111, 112.
[9]Norris, H. L., and Reynolds, W. C., "Turbulent Channel Flow with a Moving Wavy Boundary," Mechanical Engineering Dept., Rept. TF-7, Stanford Univ., Stanford, CA, 1975.
[10]Jones, W. P., and Launder, B. E., "The Prediction of Laminarization with a Two-Equation Model of Turbulence," *International Journal of Heat and Mass Transfer*, Vol. 15, 1972, pp. 301–314.
[11]Mao, Z.-X., and Hanratty, T. J., "Studies of the Wall Shear Stress in a Turbulent Pulsating Pipe Flow," *Journal of Fluid Mechanics*, Vol. 170, 1986, pp. 545–564.
[12]Johnson, D. A., and King, L. S., "A Mathematically Simple Turbulence Closure Model for Attached and Separated Turbulent Boundary Layers," *AIAA Journal*, Vol. 23, No. 11, 1985, pp. 1684–1692.
[13]Clarkson, J. D., Ekaterinaris, J. A., and Platzer, M. F., "Computational Investigation of Airfoil Stall Flutter," *Unsteady Aerodynamics, Aeroacoustics and Aeroelasticity of Turbomachines and Propellers*, edited by H. M. Atassi, Springer–Verlag, 1991, pp. 415–432.
[14]Menter, F. R., "Performance of Popular Turbulence Models for Attached and Separated Adverse Pressure Gradient Flows," *AIAA Journal*, Vol. 30, No. 8, 1992, pp. 2066–2072.
[15]Launder, B. E., and Jones, W. P., "Sink Flow Turbulent Boundary Layers," *Journal of Fluid Mechanics*, Vol. 38, Pt. 4, 1969, pp. 817–831.
[16]Schlichting, H., *Boundary Layer Theory*, McGraw–Hill, New York, 1979, pp. 604, 605.
[17]Ramaprian, B. R., and Tu, S. W., "Fully Developed Periodic Turbulent Pipe Flow. Part 2. The Detailed Structure of the Flow," *Journal of Fluid Mechanics*, Vol. 137, 1983, pp. 59–81.
[18]Crawford, M. E., and Kays, W. M., "STAN5—A Program for Numerical Computation of Two-Dimensional Internal and External Boundary Layer Flows," NASA-CR 2742, 1976.

C. G. Speziale
Associate Editor

Reordering of Hybrid Unstructured Grids for Lower–Upper Symmetric Gauss–Seidel Computations

Dmitri Sharov* and Kazuhiro Nakahashi†
Tohoku University, Sendai 980-77, Japan

I. Introduction

THE advantages of an unstructured grid over a structured grid are lower cost of grid generation, adaptive refinement/unrefinement capabilities, and savings in total number of grid points. However, most unstructured grid computations are computationally very expensive, especially for high Reynolds number flows, because conventional implicit time-integration methods based on grid line sweeps for structured grids are not suitable for unstructured grids.

Recently the lower–upper symmetric Gauss–Seidel (LU-SGS) approximate factorization,[1] which is very efficient for structured grids, has been implemented to unstructured grids by several authors.[2–4] Those methods do not require extra storage, are free from any matrix inversion, and demonstrate performance similar to the structured grid schemes.

This Note is devoted to the three-dimensional hybrid grid implementation of the LU-SGS method proposed in Ref. 2. The original method was based on sweeps through grid node numbers and was applied to two-dimensional Euler computations. Unfortunately, in its original formulation, the method cannot be vectorized; in addition the method is not compatible with commonly used edge-based data structure. To fit the method to the three-dimensional computations on a vector supercomputer, grid reordering is necessary. In Ref. 3, such a reordering was proposed. However, it was applied to the cell-centered finite volume method, and the LU-SGS formulation was different from that of Ref. 2.

Here a grid reordering method for a cell-vertex scheme on hybrid grids is proposed. This reordering provides good convergence properties, as well as vectorization capability of the code. The method has been tested on several three-dimensional tetrahedral as well as hybrid (prismatic/pyramidal/tetrahedral) grids. The Euler and Navier–Stokes equations have been computed.

Received March 10, 1997; presented as Paper 97-2102 at the 13th Computational Fluid Dynamics Conference, Snowmass Village, CO, June 29–July 2, 1997; revision received Aug. 5, 1997; accepted for publication Nov. 17, 1997.

*Assistant Professor, Department of Aeronautics and Space Engineering; currently Researcher, Research Center for Computational Science, Fujitsu, Ltd., 9-3, Nakase 1-chome, Mihama-ku, Chiba-shi 261, Japan. Member AIAA.
†Professor, Department of Aeronautics and Space Engineering. Associate Fellow AIAA.

II. Finite Volume Discretization

The governing Navier–Stokes equations are solved by the finite volume cell-vertex scheme. The control volumes are nonoverlapping dual cells constructed around each node i. Each edge connecting two nodes i and j is associated with vector area $\boldsymbol{s}_{ij}$, and gas dynamic fluxes are computed through the areas $\boldsymbol{s}_{ij}$. To enhance the accuracy of the scheme, a linear reconstruction of the primitive gas dynamic variables inside the control volume is used. Venkatakrishnan's limiter[5] is used because of its superior convergence properties. After the reconstruction, a flux quadrature is performed around each control volume using a single point along each face of the control volume. The flux is computed using a Harten–Lax–van Leer–Einfeldt–Wada approximate Riemann solver.[6]

To compute viscous stress and heat flux terms it is necessary to evaluate spatial derivatives of the primitive variables at each control volume face. These spatial derivatives are evaluated directly at the edges. The derivatives are computed by looping once through all grid cells and accumulating the result at the grid edges.

III. LU-SGS Method for Unstructured Grids

The LU-SGS formulation[2] can be described as follows.
Forward sweep:

$$\Delta \boldsymbol{Q}_i^* = \boldsymbol{D}^{-1}\left[\boldsymbol{Res}_i - 0.5\sum_{j:j\in L(i)}\left(\boldsymbol{s}_{ij}\cdot\Delta\boldsymbol{F}_j^* - |\boldsymbol{s}_{ij}|\rho_{A_j}\Delta\boldsymbol{Q}_j^*\right)\right] \quad (1a)$$

Backward sweep:

$$\Delta \boldsymbol{Q}_i = \Delta\boldsymbol{Q}_i^* - 0.5\boldsymbol{D}^{-1}\sum_{j:j\in U(i)}\left(\boldsymbol{s}_{ij}\cdot\Delta\boldsymbol{F}_j - |\boldsymbol{s}_{ij}|\rho_{A_j}\Delta\boldsymbol{Q}_j\right) \quad (1b)$$

where

$$\boldsymbol{D} = \left(\frac{\mathrm{Vol}_i}{\Delta t} + 0.5\sum_{j(i)}\rho_{A_i}|\boldsymbol{s}_{ij}|\right)\boldsymbol{I}$$

and

$$\boldsymbol{Res}_i = -\sum_{j(i)}\boldsymbol{s}_{ij}\cdot\boldsymbol{F}_{ij} + \sum_{j(i)}\boldsymbol{s}_{ij}\cdot\boldsymbol{R}_{ij}$$

where $\boldsymbol{Q}$ is the state vector; $\boldsymbol{F}$ comprises the convective flux vector components $\boldsymbol{F}_x$, $\boldsymbol{F}_y$, and $\boldsymbol{F}_z$; and $\boldsymbol{R}$ is the viscous flux vector components $\boldsymbol{R}_x$, $\boldsymbol{R}_y$, and $\boldsymbol{R}_z$. $\boldsymbol{Res}_i$ is the explicit quadrature of convective fluxes and viscous stresses, and ρ_A is the spectral radius of the Jacobian matrix $\boldsymbol{A} = \delta\boldsymbol{F}/\delta\boldsymbol{Q}$. For stable viscous computations the local diffusion velocity $v/\Delta x$ must be added to the spectral radius.

The method requires no special treatment of boundaries. Contributions from boundary faces must be considered only for the computation of the diagonal matrix $\boldsymbol{D}$.

The important issue for the unstructured grid LU-SGS methodology is the determination of lower and upper matrices; that is, how to determine the sets $L(i)$ and $U(i)$ in Eq. (1). The next section discusses this issue.

IV. Grid Reordering for LU-SGS

For structured grids, the LU-SGS sweeps are usually performed by using hyperplanes $i + j + k = \text{const}$, where i, j, and k are the indices of a grid point. Thus, forward sweep updates point (i, j, k) using already updated values at $(i-1, j, k)$, $(i, j-1, k)$, and $(i, j, k-1)$, whereas backward sweep uses nodes $(i+1, j, k)$, $(i, j+1, k)$, and $(i, j, k+1)$. This choice of hyperplanes solves two problems: 1) the problem of balance between lower and upper matrices (in the three-dimensional case every node has three neighbors contributing to the lower matrix and three neighbors contributing to the upper matrix) and 2) the vectorization problem (the procedure is vectorizable for each hyperplane).

Generally, the unstructured grids have no grid lines; therefore, some other strategy is necessary to perform the sweeps. In Ref. 2 it was proposed to sweep through node numbers from node 1 to N and back. In this case the forward sweep (lower) summation for node i is over all surrounding nodes whose numbers are lower than the current node number: $j \in L(i):(j(i) < i)$; see Eq. (1a). Backward sweep summation is over surrounding nodes whose numbers are higher than the current node number: $j \in U(i):(j(i) > i)$; see Eq. (1b). This method has shown good results for some two-dimensional Euler problems. However, the lower/upper balance of the method is highly dependent on grid node numbering; in other words, some nodes might be surrounded only by nodes whose numbers are higher (lower) than the current node number. Vectorization of the original formulation is also problematic because nodes are supposed to be processed sequentially one by one.

In Refs. 3 and 4 a grid reordering was proposed for the cell-centered scheme. We propose here a reordering algorithm for a cell-vertex LU-SGS scheme on an arbitrary unstructured grid. As in a structured grid case, we pursue two ends: balance between lower and upper matrices for majority of grid nodes and vectorization of the code. We group grid nodes so that 1) the majority of nodes from a group have connections to nodes from groups with lower group marks as well as to nodes from groups with higher group marks and 2) every node from one group has no connections with other nodes of the same group.

The reordering algorithm is as follows.

1) Mark an arbitrary grid node as belonging to hyperplane number 1. Set the current hyperplane number Np to 1. In Fig. 1 this node is marked as the starting node.

2) Assign mark $Np + 1$ to all nonmarked nodes connected to the nodes marked as Np. Increment the current plane number Np by 1.

3) If not all nodes are marked, repeat step 2. The result of this preliminary coloring is shown in Fig. 1. One can see that nodes belonging to one group may be connected to each other. To avoid such connections, step 4 is required.

4) Examine nodes from each hyperplane. For each hyperplane, mark (color) the nodes so that no nodes of the same color are connected to each other. This coloring procedure is similar to one usually applied for vectorization of unstructured grid codes. According to the colors, assign new group marks to the nodes. Numbers in Fig. 2 show the final group marks, and the final groups are highlighted.

5) To make the algorithm suitable for edge-based data structure, color the edges for forward and backward sweeps. For forward sweep, color each edge by its largest endpoint mark. For backward sweep, color each edge by its smallest endpoint mark.

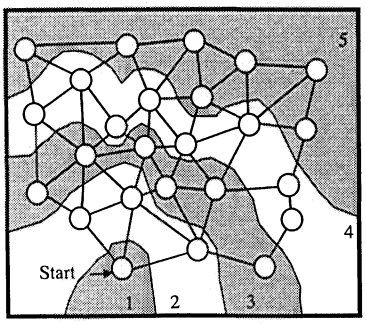

Fig. 1 Preliminary partitioning into hyperplanes.

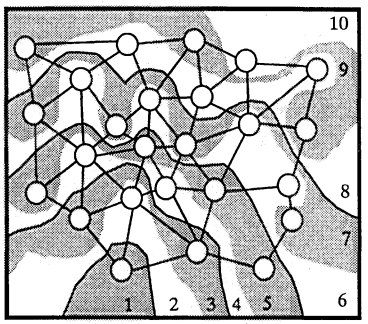

Fig. 2 Final partitioning into hyperplanes.

Thus, the lower matrix in Eq. (1a) is computed by surrounding nodes $j(i)$ whose group marks are lower than the group mark of node i, $j \in L(i)$:[mark(j) < mark(i)], whereas the upper matrix is computed by surrounding nodes $j(i)$ whose group marks are higher than the current group mark, $j \in U(i)$:[mark(j) > mark(i)]. The sweeps are performed by increasing (decreasing) the group mark numbers. For all nodes of the same group, the computation can be done concurrently, thus, the algorithm can be vectorized.

It is remarkable that this reordering, when applied to the structured grid by taking the corner point $\boldsymbol{x}(1, 1, 1)$ as a starting point, produces hyperplanes $i + j + k = \text{const}$, which is equivalent to the conventional structured grid LU-SGS method.

The reordering is performed only once before iterations are started. This method requires no matrix inversion. The CPU cost of one LU-SGS iteration is approximately 70% of one vectorized explicit time step. This means that one implicit iteration is even faster than a two-step explicit iteration. In the scalar mode, the overhead is only 60% of the explicit scheme. In our computations, we used Courant–Friedrichs–Lewy (CFL) numbers up to 1×10^6.

Memory overhead of the method is minimal. Extra memory is required only to store grid reordering marks at nodes and edges. Because the LU-SGS procedure is completely separated from the flux computation procedure, memory, which is used to compute fluxes, can be used by the LU-SGS.

V. Results

Several inviscid and viscous three-dimensional problems were computed to validate the method. Let us consider here only an example of the viscous flow past a double ellipsoid.

The hybrid grid was automatically generated by the method described in Ref. 7. The grid contains 178,418 nodes, 228,065 prisms, 1709 pyramids, and 322,643 tetrahedra. Freestream Mach number M_∞ is 8.15, angle of attack $\alpha = 30$ deg, Reynolds number $Re = 1.67 \times 10^6$, wall temperature $T_w = 288$ K, and freestream temperature $T_\infty = 56$ K. Grid details, as well as streamlines, are shown in Fig. 3.

Figure 4 shows three convergence histories: one-step explicit computation, LU-SGS implicit computation without reordering, and LU-SGS implicit computation with reordering. All three of the computations were fully vectorized computations performed on a Fujitsu VX computer. The comparison is given by the number of iterations because the CPU time of one implicit iteration is a simple superposition of CPU time of one explicit time step and the LU-SGS sweeps.

Application of the method without reordering is not trivial because of the aforementioned vectorization problems. To make the nonreordered code vectorizable and compatible with the edge-based data structure, we have performed a node-coloring procedure, which is very similar to the described procedure for reordering. Unlike the reordered case, steps 2 and 3 in the reordering algorithm were omitted. As a result, all nodes are considered to be in one big preliminary hyperplane and we perform node coloring only to avoid having two neighboring nodes of the same color; see Sec. IV, step 4. Now we can sweep using the node's colors. This procedure is a good test of the nonreordered method because it destroys any grid order, which could be produced by a grid generator.

In both implicit computations the CFL number was equal to 10^4. Choice of the CFL number deserves some comments. From

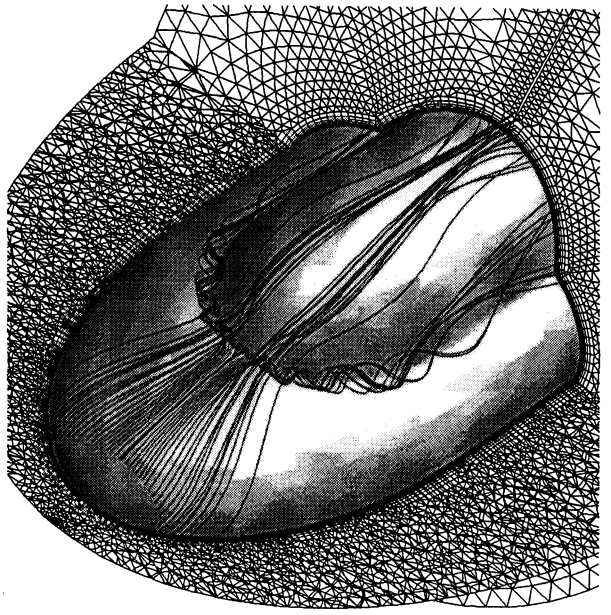

Fig. 3 Double ellipsoid; streamlines and grid.

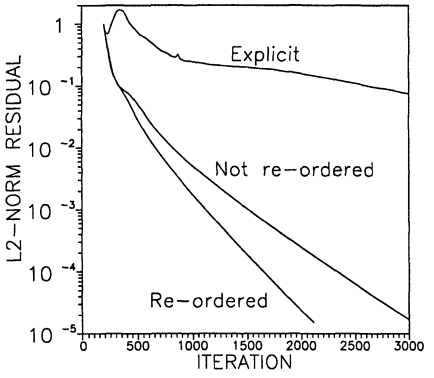

Fig. 4 Convergence history for double ellipsoid.

our experience, the discussed method is unconditionally stable for isotropic grids; however, for highly stretched grids with aspect ratios of order of 1000, the method becomes conditionally stable. It coincides with similar behavior of the structured grid LU-SGS method.

From Fig. 4 we can see that the implicit method converges much faster than the explicit one. For inviscid computations, which are not shown here, the implicit scheme is at least 10 times faster. For viscous computations the gain of the implicit scheme is even greater, because unlike the implicit scheme, the explicit scheme does not always converge. The comparison of the reordered and nonreordered versions of the method shows that the method with reordering is about 30% faster. This is a typical gain in performance, which was obtained for all of the computed problems.

VI. Conclusions

The implicit LU-SGS method has been implemented to the three-dimensional hybrid unstructured grid cell-vertex finite volume computations. The method has shown its robustness for tetrahedral and hybrid unstructured grids. The implicit part of the method is fully vectorized and takes 70% of time required for explicit flux computations. The method is approximately 10 times faster (depending on the concrete problem) as compared to the explicit method with local time stepping. The proposed reordering algorithm is approximately 30% faster than the method without reordering. Extra storage of the proposed method is approximately 5% of the total memory requirements.

References

[1]Jameson, 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.

[2]Men'shov, I., and Nakamura, Y., "Implementation of the LU-SGS Method for an Arbitrary Finite Volume Discretization," *9th Japanese Symposium on CFD*, Chuo Univ., Tokyo, Japan, 1995, pp. 123,124.

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

[4]Soetrisno, M., Imlay, S. T., Roberts, D. W., and Taflin, D. E., "Development of a 3-D Zonal Implicit Procedure for Hybrid Structured–Unstructured Grids," AIAA Paper 96-0167, Jan. 1996.

[5]Venkatakrishnan, V., "On the Accuracy of Limiters and Convergence to Steady State Solutions," AIAA Paper 93-0880, Jan. 1993.

[6]Obayashi, S., and Guruswamy, G. P., "Convergence Acceleration of an Aeroelastic Navier–Stokes Solver," AIAA Paper 94-2268, June 1994.

[7]Sharov, D., and Nakahashi, K., "Hybrid Prismatic/Tetrahedral Grid Generation for Viscous Flow Applications," *AIAA Journal*, Vol. 36, No. 2, 1998, pp. 157–162 (AIAA Paper 96-2000).

J. Kallinderis
Associate Editor

Numerical Simulation of Two-Dimensional Transverse Gas Injection into Supersonic External Flows

R. Dhinagaran* and T. K. Bose†
Indian Institute of Technology, Madras 600 036, India

Introduction

THE flowfield resulting from the injection of a transverse gas jet into a supersonic crossflow finds many applications such as thrust vector control in rocket motors, reaction control of missiles, and fuel injection in supersonic combustors and in supersonic fluidics. A typical jet interaction flowfield is shown in Fig. 1a. The injected underexpanded jet acts as an obstacle in the inviscid region inducing a bow-shaped shock called the induced bow shock. The pressure rise due to the induced shock separates the boundary layer ahead of the jet causing, in general, a separation shock. The strength of the separation shock depends very much on the nature of the boundary layer, that is, laminar, turbulent, or transitional. Typical pressure distributions for these types of separation are shown in Fig. 1b. In the case of a laminar boundary layer, separation is provoked even at a small adverse pressure gradient because the shear stress, which balances the pressure gradient, in the laminar regime is minimum. Therefore, the pressure peak is also small. The value of peak pressure for transitional separation can vary from slightly greater than that for pure laminar to a value less than that for turbulent separation depending on the position of transition as is shown in Fig. 1b. The jet upon exiting from the slot expands rapidly until it attains pressure equilibrium with the freestream through a normal shock, often termed the Mach disk. Just downstream of the slot, the jet expands, resulting in a pressure drop below that of the undisturbed value. Farther downstream, the flow is compressed to move parallel to the plate through a recompression shock. Two counter-rotating vortices upstream and a recirculation zone downstream of the slot are observed. The present study is aimed at numerically simulating the experimental data of Spaid and Zukoski[1] (runs 10–13, 16, and 17) corresponding to turbulent and transitional separations for a wide range of secondary jet to freestream total pressure ratios ($0.187 < Po_s/Po_p < 4.306$).

Received Nov. 1, 1994; revision received Oct. 9, 1997; accepted for publication Dec. 4, 1997.

*Research Scholar, Department of Aerospace Engineering.

†Professor, Department of Aerospace Engineering. Associate Fellow AIAA.