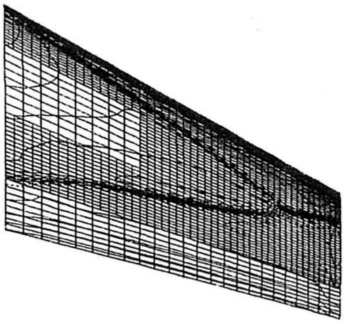

Fig. 2 Adapted grid and computed c_p isobars on the upper side of the wing.

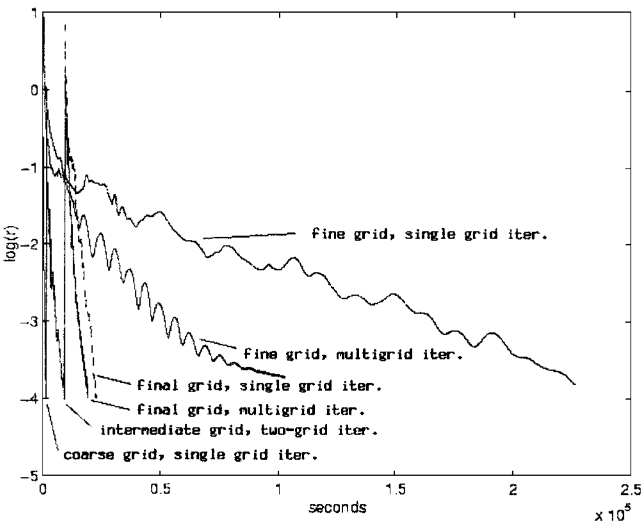

Fig. 3 Iteration on the fine grid compared to the adaptive algorithm.

Conclusion

By adapting the grid with the three-grid algorithm, we reduce not only the number of cells by almost 9/10 but also the CPU time by the same amount compared to single, fine-grid iteration and obtain a solution close to the fine-grid solution. The reduction in CPU time is due to both fewer iterations to convergence and fewer cells. There is a gain in CPU time on the adapted grid with multigrid iteration compared to single-grid iteration. The CPU time is about 35% longer for the single grid.

Acknowledgment

The work of the first author was supported by Swedish Research Council for Engineering Sciences Grant 95-560.

References

[1] Berger, M. J., and Colella, P., "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *Journal of Computational Physics*, Vol. 82, No. 1, 1989, pp. 64–84.

[2] Davis, R. L., and Dannenhoffer, J. F., III, "Three-Dimensional Adaptive Grid-Embedding Euler Technique," *AIAA Journal*, Vol. 32, No. 6, 1994, pp. 1167–1174.

[3] Pantelelis, N. G., and Kanarachos, A. E., "The Parallel Block Adaptive Multigrid Method for the Implicit Solution of the Euler Equations," *International Journal for Numerical Methods in Fluids*, Vol. 22, No. 5, 1996, pp. 411–428.

[4] Powell, K. G., "A Tree-Based Adaptive Scheme for Solution of the Equations of Gas Dynamics and Magnetohydrodynamics," *Applied Numerical Mathematics*, Vol. 14, Nos. 1–3, 1994, pp. 327–352.

[5] Brandt, A., "Multi-Level Adaptive Solutions to Boundary-Value Problems," *Mathematics of Computation*, Vol. 31, No. 138, 1977, pp. 333–390.

[6] Jameson, A., "Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method," *Applied Mathematics and Computation*, Vol. 13, Nos. 3, 4, 1983, pp. 327–355.

[7] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge–Kutta Time-Stepping Schemes," AIAA Paper 81-1259, June 1981.

J. Kallinderis
Associate Editor

Smoothness Improvements of Algebraic Surface Grid Generation

Yih Nen Jeng* and Wei Chung Hsu†
National Cheng Kung University, 70101 Tainan, Taiwan, Republic of China

Introduction

DURING the past decade, the development of computer hardware has significantly enhanced engineering applications of computational fluid dynamics (CFD).[1] As a consequence, the need to generate structured and unstructured volume grids in complicated regions has become more acute. The first step in obtaining a CFD solution is the generation of surface grids from a background grid system.

To the authors' knowledge, the most convenient and fastest method to generate surface grids seems to be the parameter space method developed by Samareh-Abolhassani and Stewart.[2] Assume that a structured quadrilateral background grid system is known, and then define the normalized arc length of the background grid lines and consider the arc lengths along each computational direction to be independent variables. By employing the scaled transfinite interpolation method of Soni,[3] followed by a backward projection to the background grids, a surface grid system can be generated rapidly.

The authors' experience has shown that the method proposed by Samareh-Abolhassani and Stewart[2] works very well for many engineering applications provided that the range of the arc length normalization just covers the target region of surface grid generation. However, if the total arc lengths between two adjacent lines of the background grids differ significantly from each other, numerical tests show that grid line oscillation occurs. These extreme cases correspond to a fin extruding from a smooth surface, where the exact geometry of the fin is not too important in CFD simulation.

To remove the described grid oscillation, three modifications were proposed in Ref. 4. The method that adds a uniform smoothing term to the arc-length parameter space (APS) method proposed in Ref. 2 effectively improves grid smoothness provided that the smoothing factor is large enough. However, as the smoothing factor increases, it approaches the uniform parameter space (UPS) method of Ref. 2, which cannot handle the nonsmoothness of the background grids. To avoid the requirement of a large smoothing factor, four different modifications have been examined. However, only the best modification is presented here due to length limitations.

Generally speaking, a surface definition is frequently obtained from the CAD software that provides the patched grid system. If the patched grids are not of a structured quadrilateral system, the user must reorganize them into a structured quadrilateral background grid system. If the background grid system is not smooth, numerical tests show that neither the methods in Refs. 2–4 nor the modification mentioned earlier can always eliminate this oscillation. Further, numerical tests show that, if the background grid lines do not seriously overlap, it is helpful to smooth the background grids point by point before the surface grid generation is done. Note that such a smoothing may include three stages: projecting neighboring points to a tangent plane passing through a point, finding the new point via local grid smoothing,[5] and projecting the new point backward onto the original surface. Otherwise, reorganization of the background grids is necessary.

Presented as Paper 97-0203 at the AIAA 35th Aerospace Sciences Meeting, Reno, NV, Jan. 6–9, 1997; received April 28, 1997; revision received Aug. 5, 1998; accepted for publication Aug. 21, 1998.

*Professor, Department and Institute of Aeronautics and Astronautics. Member AIAA.

†Graduate Student, Department and Institute of Aeronautics and Astronautics.

Formulation

For the sake of completeness, aspects of Refs. 2 and 4 are restated. Samareh-Abolhassani and Stewart[2] proposed several mappings onto parameter space, among which UPS mapping and APS are of particular interest to this study. Note that Samareh-Abolhassani and Stewart recommended APS mapping because it is the most representative of the background grids.

Assume that, on a curved surface, a system of quadrilateral background grids is known. Let the region enclosed by grid lines with indices $i = 0, i_m$ and $j = 0, j_m$ be the smallest region covering the target region. The UPS mapping is constructed by

$$U_{i,j} = i/\,i_m, \qquad V_{i,j} = j/\,j_m \tag{1}$$

where U and V are independent variables of the parameter space. The APS mapping computes

$$\begin{aligned} u_{0,j} &= 0, & v_{i,0} &= 0 \\ u_{i,j} &= u_{i-1,j} + \Delta s_i^j, & v_{i,j} &= v_{i,j-1} + \Delta s_j^i \end{aligned} \tag{2}$$

$$\begin{aligned} \Delta s_i^j &= \sqrt{(x_{i,j} - x_{i-1,j})^2 + (y_{i,j} - y_{i-1,j})^2 + (z_{i,j} - z_{i-1,j})^2} \\ \Delta s_j^i &= \sqrt{(x_{i,j} - x_{i,j-1})^2 + (y_{i,j} - y_{i,j-1})^2 + (z_{i,j} - z_{i,j-1})^2} \end{aligned} \tag{3}$$

and defines the independent variables via the normalization

$$U_{i,j} = u_{i,j}/\,u_{i_m,j}, \qquad V_{i,j} = v_{i,j}/\,v_{i,j_m} \tag{4}$$

After distributing boundary grids along edge boundaries of the target region, the corresponding boundaries $U_{i,j}$ and $V_{i,j}$ are evaluated by definition. Then the scaled transfinite interpolation method of Ref. 3 is employed to generate the interior $(U_{i,j}, V_{i,j})$.

For every interior point, the backward mapping stage involves three steps: 1) search for a patch in the background parameter space that surrounds the grid point, 2) compute the local coordinates of the patch using bilinear or higher-order interpolation, and 3) compute the coordinates of the grid points in the physical space. During the second step, Newton–Raphson iteration is employed to solve local coordinates. In Ref. 4, the angular search employed in Ref. 2 is replaced by the following criteria: Suppose that we examine whether a point P is located in the patch defined by the points, for example, 1, 2, 3, and 4 in the counterclockwise sense. If all of the following inequalities are satisfied, point (i, j) is surrounded by the patch

$$\begin{aligned} A_1 &= \boldsymbol{r}_{12} \times \boldsymbol{r}_{1P} \geq 0, & A_2 &= \boldsymbol{r}_{23} \times \boldsymbol{r}_{2P} \geq 0 \\ A_3 &= \boldsymbol{r}_{34} \times \boldsymbol{r}_{3P} \geq 0, & A_4 &= \boldsymbol{r}_{41} \times \boldsymbol{r}_{4P} \geq 0 \end{aligned} \tag{5}$$

which is equivalent to whether

$$\operatorname{sign}(A_1 + \epsilon) + \operatorname{sign}(A_2 + \epsilon) + \operatorname{sign}(A_3 + \epsilon) + \operatorname{sign}(A_4 + \epsilon) = 4 \tag{6}$$

where $\boldsymbol{r}_{12}$ is the vector measured from point 1 to point 2 on the parameter space, etc., and ϵ is a small positive constant in the order of machine error to include the case that P is located on line 1–2, line 2–3, line 3–4, or line 4–1. This search is simpler than the angular search because the evaluation of the $\tan^{-1}$ function is not necessary.

To remedy grid oscillation in extreme cases, three modifications of APS mapping are examined in Ref. 4. The most effective modification is a combination of UPS and APS mappings. Further study showed that there are still simpler and more effective methods. Among these methods, the most effective method is described next.

Careful examination of the extreme cases considered in Ref. 4 reveals that there are two types of variation that cause grid oscillation: The first is the variation of total arc length between grid lines, and the second is the variation of arc length along grid lines. In Ref. 4, the idea of including adaptive information from adjacent grid lines[5–8] is employed to partially eliminate the first variation. In this study, the nonuniform weighting across grid lines is modified to be uniform weighting. Although the method of uniform averaging the arclength term of Eq. (3) over a large region can effectively eliminate these two variations, it turns out to be the UPS method that has certain drawbacks, as shown in Ref. 2. Therefore, the following method, which performs local averaging over a limited region surrounding point (i, j), is employed:

$$\begin{aligned} u_{i,j} &= u_{i-1,j} + \frac{\omega}{i_m} + (1-\omega) \\ &\quad \times \left[(1-\alpha)\Delta s_i^j + \alpha \frac{\sum_{k=-n}^{n} \sum_{\ell=-n}^{n} \Delta s_{i+\ell}^{j+k}}{(2n+1)^2} \right] \\ v_{i,j} &= v_{i,j-1} + \frac{\omega}{j_m} + (1-\omega) \\ &\quad \times \left[(1-\alpha)\Delta s_j^i + \alpha \frac{\sum_{k=-n}^{n} \sum_{\ell=-n}^{n} \Delta s_{j+\ell}^{i+k}}{(2n+1)^2} \right] \\ U_{i,j} &= \frac{u_{i,j}}{u_{i_m,j}}, \qquad V_{i,j} = \frac{v_{i,j}}{v_{i,j_m}} \end{aligned} \tag{7}$$

where ω is a user-specified constant and $\alpha = 1$ and $n = 4$ are used in this study. If $\omega \to 1$, the mapping becomes UPS mapping, and if $\omega \to 0$, it becomes a less effective method. Generally, neither $\omega \to 0$ nor $\omega \to 1$ is recommended.

Results and Discussion

To demonstrate the extreme situations where the total arc lengths between two adjacent lines of background grids differ significantly from each other, many extreme examples were considered. However, because of space limitations, only a typical example is shown here (Fig. 1). This example is whether or not a grid line passing through the extruding fin significantly affects the total arc length.

Using the background grid system shown in Fig. 1, Ref. 4 shows the application of APS mapping of Eqs. (2) and (4) to obtain a background mesh on the (U, V) plane, which shows grid overlapping and is not shown here due to space limitation. If the grid system on the parameter space is employed to generate surface grids, Newton–Raphson iteration in the backward mapping stage might diverge for points located at the overlapping region.

The background grids on the proposed mapping are shown in Fig. 2, where $n = 4$, $\alpha = 1$, and $\omega = 0.5$. Grid smoothness is preserved very well over the entire domain on the parameter space. The final grid distribution shown in Fig. 3 has satisfactory grid smoothing.

Numerical tests using $n < 4$ and/or $\alpha < 1$ (not shown here) result in grid systems less smooth than that shown in Fig. 3. For those cases with less complicated geometry, the case with $n = 4$ is only slightly worse than that using $n < 4$. As for the smoothing factor ω, numerical tests show that $\omega = 0.5 \sim 0.8$ is a suitable range. In some still more complicated problems, where the differences of total arclength between adjacent grid lines are vary large, the resulting grid smoothness employing $\omega = 0.5$ is slightly less effective than that of $0.5 < \omega \leq 0.8$. Because the surface grid generation procedure is very fast, it is recommended to define a grid quality indicator[1]

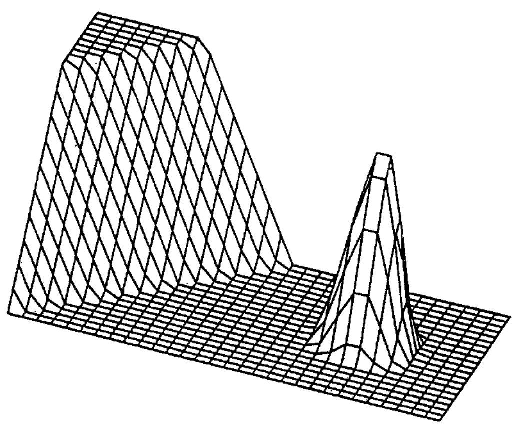

Fig. 1 Background grids of an extreme case on the physical domain.

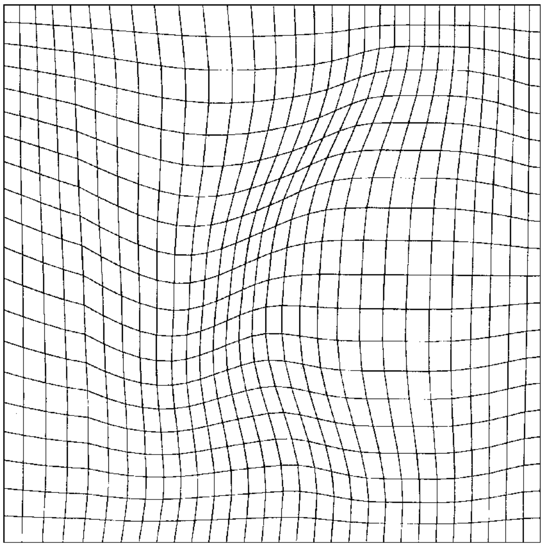

Fig. 2 Background grids on the (U, V) space by the proposed mapping with $n = 4$, $\alpha = 1$, and $\omega = 0.5$.

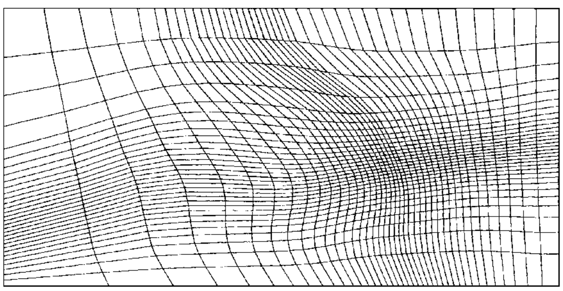

Fig. 3 Top view of the final surface grids.

such that the grid solver is automatically repeated with a larger ω whenever the grid quality is poor.

Conclusions

A modification of APS mapping previously proposed is shown. The new proposed method is a weighted average between UPS mapping and APS mapping that averages arclength over a large region. Numerical tests show that it improves grid smoothness of an extreme case while being very simple.

Acknowledgment

The work presented was supported by Taiwan, Republic of China, National Science Council Grant 85-2212-E006-090.

References

[1]Thompson, J. F., "A Reflection on Grid Generation in the 90s: Trends, Needs, and Influences," *Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Field Simulations*, edited by B. K. Soni, J. F. Thompson, J. Häuser, and P. R. Eiseman, Mississippi State Univ., Mississippi State, MS, 1996, pp. 1029–1110.

[2]Samareh-Abolhassani, J., and Stewart, E., "Surface Grid Generation in a Parameter Space," *Journal of Computational Physics*, Vol. 113, No. 1, 1994, pp. 112–121.

[3]Soni, B. K., "Two- and Three-Dimensional Grid Generation for Internal Flow Applications of Computational Fluid Dynamics," AIAA Paper 85-1526, July 1985.

[4]Jeng, Y. N., and Hsu, W. C., "Enhancements upon the Algebraic Surface Grid Generation Method on Parameter Space," AIAA Paper 97-0203, Jan. 1997.

[5]Jeng, Y. N., and Lee, Z. S., "Revisit to the Modified Muliple One-Dimensional Adaptive Grid Method," *Numerical Heat Transfer*, Pt. B, Vol. 29, No. 3, 1996, pp. 305–323.

[6]Jeng, Y. N., and Liou, S. C., "Modified Multiple One-Dimensional Adaptive Grid Method," *Numerical Heat Transfer*, Pt. B, Vol. 15, No. 2, 1989, pp. 241–247.

[7]Lee, D., and Tusei, Y. M., "A Modified Adaptive Grid Method for Recirculating Flow," *International Journal for Numerical Methods in Fluids*, Vol. 14, No. 7, 1992, pp. 775–791.

[8]Lee, D., and Tusei, Y. M., "A Hybrid Adaptive Gridding Procedure for Recirculating Fluid Flow Problems," *Journal of Computational Physics*, Vol. 108, No. 1, 1993, pp. 122–141.

J. Kallinderis
Associate Editor

Efficiency Improvement of Unified Implicit Relaxation/Time Integration Algorithms

Kozo Fujii*
Institute for Space and Astronautical Science, Sagamihara, Kanagawa 229-8510, Japan

Introduction

FOR space-oriented applications of computational fluid dynamics, subsonic flow regions are locally confined and the steady flowfields to be simulated are mostly governed by supersonic flows. For the simulation of such flows, relaxation-type schemes are much more efficient compared with conventional time-integration schemes. Local iterations at each streamwise station are defined, and the converged solution is obtained there before moving to the next streamwise station. Since the middle of the 1980s quite a few papers have been written on this subject.[1–4] However, conventional time-integration schemes that use only global iterations are still more popular because relaxation-type computations are very efficient for supersonic flows but are usually not as efficient for subsonic and transonic flows. Recent implicit time-integration schemes[5,6] simplified the implicit operations and are much more efficient for subsonic and transonic flows.

In the present Note, relaxation-type algorithms developed in the 1980s are revisited, and their efficiency is improved by adopting the idea of both the lower–upper-symmetric Gauss–Seidel (LU-SGS) and the lower–upper alternating direction implicit (LU-ADI) schemes. Although the code structure is that of relaxation-type schemes, the present scheme is as efficient as conventional efficient schemes in the time-marching mode. When using the local iteration process, the scheme becomes much more efficient for simulations of flows primarily governed by the supersonic region.

Formulations

We start with the two-factored scheme proposed by Yin and Steger.[3] When using diagonally dominant lower–diagonal–upper (LDU) factorization instead of the LU factorization of the original formulation and dividing the equations by Δt, the modified two-factored implicit scheme becomes

$$\left(\frac{I}{\Delta t} - \frac{1}{\Delta \xi}\hat{A}_j^- + \delta_\xi^b \hat{A}^+ + \delta_\zeta \hat{C}\right)\left[\frac{I}{\Delta t} + \frac{1}{\Delta \xi}\left(\hat{A}_j^+ - \hat{A}_j^-\right)\right]^{-1}$$
$$\times \left(\frac{I}{\Delta t} + \frac{1}{\Delta \xi}\hat{A}_j^+ + \delta_\xi^f \hat{A}^- + \delta_\eta \hat{B}\right)\Delta \hat{Q}^n$$
$$= -\left(\frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} + \frac{\partial \hat{G}}{\partial \zeta}\right)^n \qquad (1)$$

Presented as Paper 97-2105 at the AIAA 13th Computational Fluid Dynamics Conference, Snowmass Village, CO, June 29–July 2, 1997; received July 26, 1997; revision received Aug. 19, 1998; accepted for publication Sept. 9, 1998.

*Professor, Space Transportation Research Division. Associate Fellow AIAA.