AIAA JOURNAL
Vol. 32, No. 5, May 1994

Multigrid Convergence of an Implicit Symmetric Relaxation Scheme

Seokkwan Yoon* and Dochan Kwak†
NASA Ames Research Center, Moffett Field, California 94035

The multigrid method has been applied to an existing three-dimensional compressible Euler solver to accelerate the convergence of the implicit symmetric relaxation scheme. This lower-upper symmetric Gauss-Seidel implicit scheme is shown to be an effective multigrid driver in three dimensions. A grid refinement study is performed including the effects of large cell aspect ratio meshes. Performance figures of the present multigrid code on Cray computers including the new C90 are presented. A reduction of three orders of magnitude in the residual for a three-dimensional transonic inviscid flow using 920 k grid points is obtained in less than 4 min on a Cray C90.

I. Introduction

ALTHOUGH unstructured grid methods have been used successfully in solving the Euler equations for complex geometries, structured grid solvers still remain useful for the Navier-Stokes equations because of their natural advantages in dealing with the highly clustered meshes in the viscous boundary layers. Structured grid methods not only handle reasonably complex geometries using multiple blocks but also offer a hybrid grid scheme to alleviate difficulties which unstructured grid methods have often encountered. Recent developments in structured grid solvers have been focused on efficiency, as well as accuracy, since most existing three-dimensional Navier-Stokes codes are still not efficient enough to be used routinely for aerodynamic design.

Multigrid methods have been useful for accelerating the convergence of iterative schemes. Efficient Euler codes have been developed by Jameson[1] using a full approximation storage method of Brandt[2] in conjunction with the explicit Runge-Kutta scheme. The explicit multigrid method has demonstrated impressive convergence rates by taking large time steps and propagating waves fast on coarse meshes. The explicit multigrid method has been extended to the Navier-Stokes equations by Martinelli.[3] Several explicit multigrid codes for the three-dimensional Navier-Stokes equations have been developed successfully by Vatsa and Wedan,[4] Radespiel et al.,[5] and Turkel et al.[6]

It does not seem to be profitable to consider an unfactored implicit scheme for a multigrid driver since the implicit scheme can take large time steps which are limited by the physics rather than the grid. However, the multigrid method can improve the convergence rates of factored implicit schemes in two dimensions as demonstrated by Yoon[7] and in Refs. 8–10. The implicit multigrid method has been implemented by Caughey[12] to the diagonalized alternating direction scheme,[11] and by Anderson et al.[13] to the three-dimensional alternating direction scheme. Yoon[7] has introduced an implicit algorithm based on a lower-upper (LU) factorization and symmetric Gauss-Seidel (SGS) relaxation. The scheme has been used successfully in computing chemically reacting flows due in part to the algorithm's property which reduces the size of the left-hand side matrix for nonequilibrium flows with finite rate chemistry.[14–16] A recent study[17] shows that the three-dimensional extension of the method using a single grid requires less computational time than most existing codes on a Cray YMP computer. One of the objectives of the present work is to accelerate the convergence of the lower-upper symmetric Gauss-Seidel relaxation scheme in three dimensions by introducing a multigrid technique. The performance of the code is demonstrated for an inviscid transonic flow past an ONERA M6 wing on highly clustered grids.

II. Governing Equations

Let t be time; $\hat{Q}$ the vector of conserved variables; $\hat{E}$, $\hat{F}$, and $\hat{G}$ the convective flux vectors; and $\hat{E}_v$, $\hat{F}_v$, and $\hat{G}_v$ the flux vectors for the viscous terms. Then the three-dimensional Navier-Stokes equations in generalized curvilinear coordinates (ξ, η, ζ) can be written as

$$\partial_t \hat{Q} + \partial_\xi(\hat{E} - \hat{E}_v) + \partial_\eta(\hat{F} - \hat{F}_v) + \partial_\zeta(\hat{G} - \hat{G}_v) = 0 \qquad (1)$$

where the flux vectors are defined in Ref. 17. The Euler equations are obtained by neglecting the viscous terms.

III. Lower-Upper Symmetric Gauss-Seidel Implicit Scheme

An unfactored implicit scheme can be obtained from a nonlinear implicit scheme by linearizing the flux vectors about the previous time step and dropping terms of the second and higher order.

$$[I + \alpha\Delta t(D_\xi \hat{A} + D_\eta \hat{B} + D_\zeta \hat{C})]\delta\hat{Q} = -\Delta t\hat{R} \qquad (2)$$

where the residual $\hat{R}$ is

$$\hat{R} = D_\xi \hat{E} + D_\eta \hat{F} + D_\zeta \hat{G} \qquad (3)$$

and I is the identity matrix. The correction $\hat{Q}^{n+1} - \hat{Q}^n$ is $\delta\hat{Q}$, where n denotes the time level. D_ξ, D_η, and D_ζ are difference operators that approximate ∂_ξ, ∂_η, and ∂_ζ, respectively. $\hat{A}$, $\hat{B}$, and $\hat{C}$ are the Jacobian matrices of the convective flux vectors. For $\alpha = \frac{1}{2}$, the scheme is second-order accurate in time. For other values of α, the time accuracy drops to first order.

An efficient implicit scheme can be derived by combining the advantages of LU factorization and Gauss-Seidel relaxation.

$$LD^{-1}U\delta\hat{Q} = -\Delta t\hat{R} \qquad (4)$$

Received May 17, 1993; presented as Paper 93-3357 at the AIAA Computational Fluid Dynamics Conference, Orlando, FL, July 6–9, 1993; revision received Oct. 4, 1993; accepted for publication Oct. 13, 1993.

*Research Scientist, Computational Algorithms and Applications Branch.
†Chief, Computational Algorithms and Applications Branch.

Here,

$$L = I + \alpha\Delta t(D_\xi^- \hat{A}^+ + D_\eta^- \hat{B}^+ + D_\zeta^- \hat{C}^+ - \hat{A}^- - \hat{B}^- - \hat{C}^-)$$

$$D = I + \alpha\Delta t(\hat{A}^+ - \hat{A}^- + \hat{B}^+ - \hat{B}^- + \hat{C}^+ - \hat{C}^-) \tag{5}$$

$$U = I + \alpha\Delta t(D_\xi^+ \hat{A}^- + D_\eta^+ \hat{B}^- + D_\zeta^+ \hat{C}^- + \hat{A}^+ + \hat{B}^+ + \hat{C}^+)$$

where D_ξ^-, D_η^-, and D_ζ^- are backward difference operators, while D_ξ^+, D_η^+, and D_ζ^+ are forward difference operators.

In the framework of the LU-SGS algorithm, a variety of schemes can be developed by different choices of numerical dissipation models and Jacobian matrices of the flux vectors. Jacobian matrices leading to diagonal dominance are constructed so that + matrices have non-negative eigenvalues whereas − matrices have nonpositive eigenvalues. For example,

$$\hat{A}^\pm = \hat{T}_\xi \Lambda_\xi^\pm \hat{T}_\xi^{-1}$$

$$\hat{B}^\pm = \hat{T}_\eta \Lambda_\eta^\pm \hat{T}_\eta^{-1} \tag{6}$$

$$\hat{C}^\pm = \hat{T}_\zeta \Lambda_\zeta^\pm \hat{T}_\zeta^{-1}$$

where typically $\hat{T}_\xi$ and $\hat{T}_\xi^{-1}$ are similarity transformation matrices of the eigenvectors of $\hat{A}$. Another possibility is to construct the Jacobian matrices of the flux vectors approximately which yield diagonal dominance.

$$\hat{A}^\pm = \tfrac{1}{2}[\hat{A} \pm \tilde{\rho}(\hat{A})I]$$

$$\hat{B}^\pm = \tfrac{1}{2}[\hat{B} \pm \tilde{\rho}(\hat{B})I] \tag{7}$$

$$\hat{C}^\pm = \tfrac{1}{2}[\hat{C} \pm \tilde{\rho}(\hat{C})I]$$

where

$$\tilde{\rho}(\hat{A}) = \kappa \max[|\lambda(\hat{A})|] \tag{8}$$

Here $\lambda(\hat{A})$ represents eigenvalues of the Jacobian matrix $\hat{A}$, and κ is a constant that is greater than or equal to 1. A typical value of κ is 1. However, stability and convergence can be controlled by adjusting κ as the flowfield develops.

IV. Vectorization

The algorithm is completely vectorizable on $i+j+k=$ const diagonal planes of sweep. This is achieved by reordering the arrays.

$$\hat{Q}(i\text{point}, i\text{plane}) = \hat{Q}(i, j, k) \tag{9}$$

where $i$plane is the serial number of the diagonal plane, and $i$point is the address on that plane. The number of diagonal planes is given by

$$n\text{plane} = i\max + j\max + k\max - 5 \tag{10}$$

with the maximum vector length of

$$n\text{point} = (j\max - 1)*(k\max - 1) \tag{11}$$

V. Numerical Dissipation

The cell-centered finite-volume method is augmented by a numerical dissipation model based on blended first- and third-order terms.[18-21] The finite volume method is based on the local flux balance of each mesh cell. For example,

$$\partial_\xi \hat{E} + \partial_\eta \hat{F} + \partial_\zeta \hat{G} = \hat{E}_{i+\frac{1}{2},j,k} - \hat{E}_{i-\frac{1}{2},j,k} + \hat{F}_{i,j+\frac{1}{2},k} - \hat{F}_{i,j-\frac{1}{2},k} + \hat{G}_{i,j,k+\frac{1}{2}} - \hat{G}_{i,j,k-\frac{1}{2}} \tag{12}$$

Here, the conserved variables are averaged at the cell faces to evaluate the fluxes. The dissipative flux d is added to the convective flux in a conservative manner.

$$(\hat{E}_{i+\frac{1}{2},j,k} - \hat{E}_{i-\frac{1}{2},j,k} + \hat{F}_{i,j+\frac{1}{2},k} - \hat{F}_{i,j-\frac{1}{2},k} + \hat{G}_{i,j,k+\frac{1}{2}} - \hat{G}_{i,j,k-\frac{1}{2}}) - (d_{i+\frac{1}{2},j,k} - d_{i-\frac{1}{2},j,k} + d_{i,j+\frac{1}{2},k} - d_{i,j-\frac{1}{2},k} + d_{i,j,k+\frac{1}{2}} - d_{i,j,k-\frac{1}{2}}) \tag{13}$$

For simplicity, $d_{i+\frac{1}{2},j,k}$ is denoted by $d_{i+\frac{1}{2}}$ hereafter.

$$d_{i+\frac{1}{2}} = \epsilon^{(2)}_{i+\frac{1}{2}}(\hat{Q}_{i+1} - \hat{Q}_i) - \epsilon^{(4)}_{i+\frac{1}{2}}(\hat{Q}_{i+2} - 3\hat{Q}_{i+1} + 3\hat{Q}_i - \hat{Q}_{i-1}) \tag{14}$$

The coefficients of the dissipative terms are the directionally scaled spectral radii of the Jacobian matrices. The use of directional scaling provides anisotropic dissipation to each direction, resulting in improved performance on meshes with high aspect ratio cells. Third-order terms formed from fourth differences provide the background damping. First-order terms are added by second differences near shock waves under the control of a sensor $\tilde{\nu}$.

$$\tilde{\nu}_{i+\frac{1}{2}} = \max(\nu_{i+1}, \nu_i) \tag{15}$$

where

$$\nu_i = \max(\nu_i^p, \nu_i^T) \tag{16}$$

$$\nu_i^p = |p_{i+1} - 2p_i + p_{i-1}|/(p_{i+1} + 2p_i + p_{i-1}) \tag{17}$$

$$\nu_i^T = |T_{i+1} - 2T_i + T_{i-1}|/(T_{i+1} + 2T_i + T_{i-1}) \tag{18}$$

Here p and T are the pressure and the temperature. The low-order dissipative coefficient is proportional to the sensor $\tilde{\nu}$ as

$$\epsilon^{(2)}_{i+\frac{1}{2}} = \kappa^{(2)} r(\hat{A})_{i+\frac{1}{2}} \tilde{\nu}_{i+\frac{1}{2}} \tag{19}$$

where $r(\hat{A})$ denotes the spectral radius of the Jacobian matrix $\hat{A}$. The high-order dissipative coefficient is controlled by the sensor.

$$\epsilon^{(4)}_{i+\frac{1}{2}} = \max[0, \kappa^{(4)} r(\hat{A})_{i+\frac{1}{2}} - \epsilon^{(2)}_{i+\frac{1}{2}}] \tag{20}$$

Here $\kappa^{(2)}$ and $\kappa^{(4)}$ are constants which are different from other κ in Eq. (8).

Dissipative terms for the coarse grids in the multigrid levels are formed from second differences with constant coefficients.

Since the constant total enthalpy is not preserved in general except for the Euler equations, the dissipation for the energy equation is based on the total energy rather than the total enthalpy. To reduce the amount of dissipation in the direction normal to the body surface inside boundary layers, Swanson and Turkel[20] provided additional scaling by multiplying a spectral radius in the normal direction by a function of the local Mach number. Although the Mach number scaling technique may improve the accuracy of the Navier-Stokes solutions, it has not been used for the Euler solutions. It has been shown that the convergence rate on high cell aspect ratio meshes can be enhanced by multiplying Martinelli's[3] scaling factor based on local cell aspect ratio to the dissipative coefficients. However, this technique has not been used here because the aspect ratio based scaling factor seems to compromise the accuracy of the solution.[22]

VI. Multigrid Method

In the present multigrid method, part of the task of tracking the evolution of the solution is transferred through a sequence of successively coarser meshes. The use of larger control volumes on the coarser meshes tracks the large-scale evolution,

with the consequence that global equilibrium can be more rapidly attained. This evolution on the coarse grid is driven by the solution of the fine grid equations. The solution vector on a coarse grid is initialized as

$$Q_{2h}^{(0)} = \Sigma S_h Q_h / S_{2h} \tag{21}$$

where the subscripts denote values of the mesh spacing parameter h, S is the cell volume, and the sum is over the eight cells of the fine grid which compose each cell of the coarse grid. After updating the fine grid solution, the values of the conserved variables are transferred to the coarse grid using Eq. (21). The pressure is calculated on the coarse grid using the transferred variables. Then a forcing function is defined as

$$P_{2h} = \Sigma R_h(Q_h) - R_{2h}(Q_{2h}^{(0)}) \tag{22}$$

The residual on the coarse grid is given by

$$R_{2h}^{*} = R_{2h}(Q_{2h}) + P_{2h} \tag{23}$$

For the next coarser grid, the residual is calculated as

$$R_{4h}^{*} = R_{4h}(Q_{4h}) + P_{4h} \tag{24}$$

where

$$P_{4h} = \Sigma R_{2h}^{*} - R_{4h}(Q_{4h}^{(0)}) \tag{25}$$

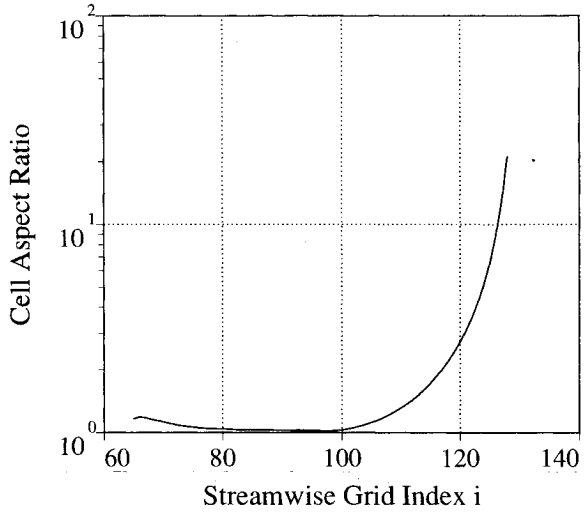

Fig. 1 Distribution of cell aspect ratios (CAR) for the low CAR grid (140 k grid points).

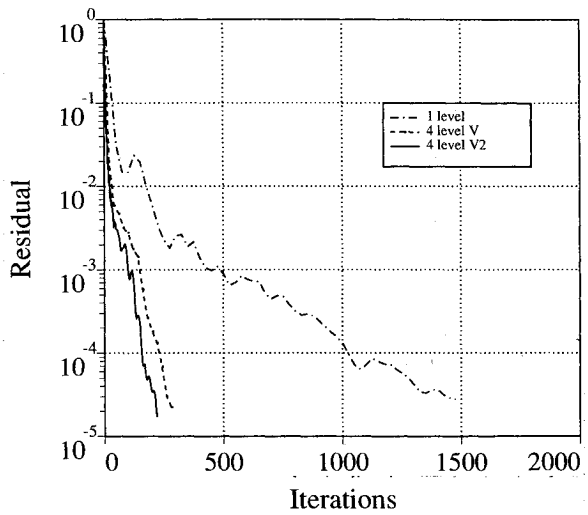

Fig. 2 Convergence histories on the low CAR grid.

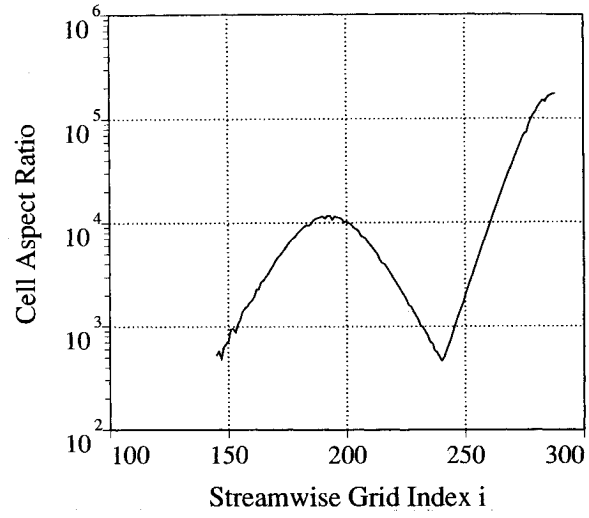

Fig. 3 Distribution of cell aspect ratios for the high CAR grid (920 k grid points).

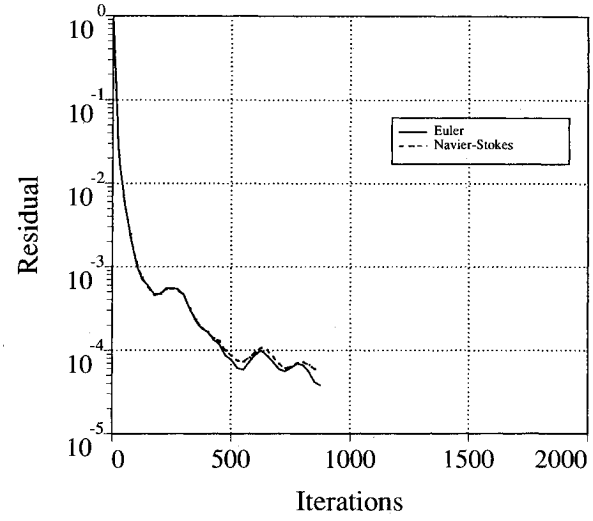

Fig. 4 Convergence histories for the Euler and Navier-Stokes equations.

The process is repeated on successively coarser grids. Multiple iterations can be done on each coarse grid. Finally, the correction calculated on each grid is interpolated back to the next finer grid. Let $\tilde{Q}_{2h}$ be the final value of Q_{2h} resulting from both the correction calculated on grid $2h$ and the correction transferred from the grid $4h$. Then

$$\tilde{Q}_h = Q_h + I_{2h}^{h}(\tilde{Q}_{2h} - Q_{2h}^{(0)}) \tag{26}$$

where Q_h is the solution on grid h before the transfer to the grid $2h$, and I is a trilinear interpolation operator. Since the evolution on a coarse grid is driven by residuals collected from the next finer grid, the final solution on the fine grid is independent of the choice of boundary conditions on the coarse grids.

VII. Results

To investigate the effectiveness of the full approximation storage multigrid method in conjunction with the lower-upper symmetric Gauss-Seidel relaxation scheme, transonic flow calculations have been carried out for an ONERA M6 wing. The freestream conditions are at a Mach number of 0.8395, and a 3.06-deg angle of attack. Since this is an unseparated flow case, only the solution of the three-dimensional Euler equations is considered.

All of the calculations are performed without the aid of a couple of conventional techniques which have been crucial to ensure the robustness of the multigrid method. The aspect ratio based scaling factor for the numerical dissipation is not employed because of the reason stated in Sec. V. The enthalpy damping technique, which is not valid for the Navier-Stokes equations, is not used here.

The treatment of the far-field boundary condition is based on the Riemann invariants for a one-dimensional flow normal to the boundary.

To study the effects of mesh cell aspect ratios on the convergence rates, a low cell aspect ratio $129 \times 33 \times 33$ C-H grid (140,481 points) is used first. Figure 1 shows the distribution of geometric cell aspect ratios of the first normal mesh cells at the body from the leading edge to the downstream boundary. Although geometric aspect ratios are less accurate than spectral radius based aspect ratios, they seem to be useful. The cell aspect ratios are of the order of 1 at the wing surface. Figure 2 shows convergence histories of the root-mean-squared residuals which correspond to the density corrections. The residuals are normalized by their initial values. The chained line indicates the single grid convergence history. The dashed line indicates the multigrid convergence history using a four-level V cycle with one iteration on each coarse grid. The solid line indicates the multigrid convergence history using a four-level V cycle with two iterations on each coarse grid. Clearly, the two iteration strategy converges faster than the single iteration cycle. Although not shown here, the three-iteration cycle does not improve the performance due to a slight increase in work per cycle. Hence, all of the following multigrid calculations have been performed using the V cycle with two coarse grid iterations. Four orders of residual drop requires 1022 and 156 iterations for single and multigrid, respectively. The convergence has been accelerated by a factor of 6.5 by the use of the multigrid method on this low cell aspect ratio grid.

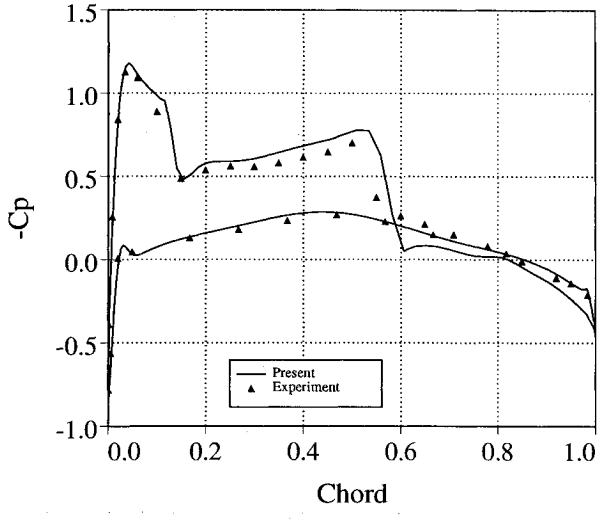

Fig. 5 C_p at 44% span for ONERA M6 wing.

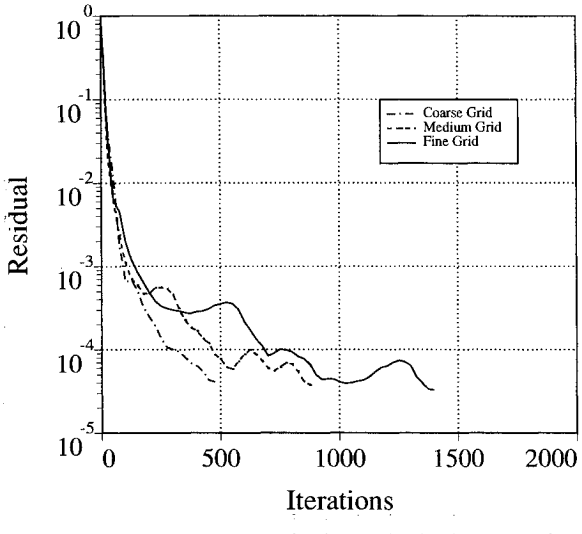

Fig. 6 Single grid convergence histories on the high CAR grids.

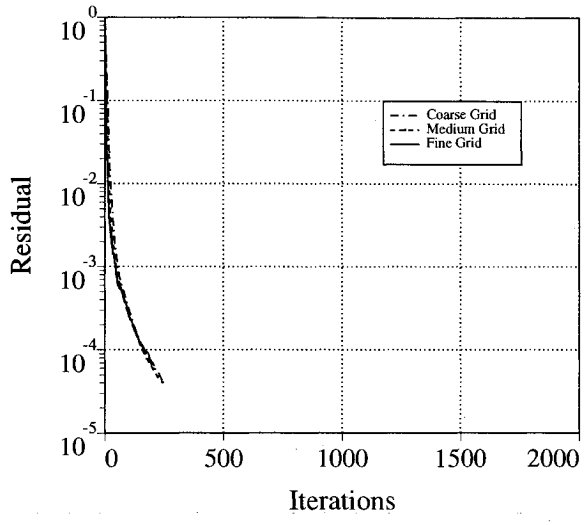

Fig. 7 Multigrid convergence histories on the high CAR grids.

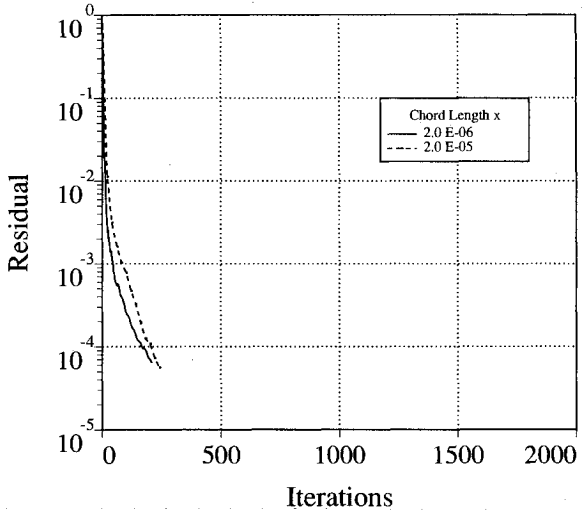

Fig. 8 Effect of the first grid distance on the multigrid convergence.

A high cell aspect ratio $289 \times 65 \times 49$ C-H grid (920,465 points) gives a severe test to the code. Here 65 grid points are used in the normal direction. The distance of the first normal grid point from the wing surface is 2.0×10^{-6} times the chord length. Figure 3 shows that the cell aspect ratios on the wing reach as high as 10,000. This grid will be known as the fine grid. Two more grids are prepared for a grid refinement study. A $145 \times 33 \times 25$ medium grid (119,625 points) is generated by eliminating every other grid point from the fine grid. A $73 \times 17 \times 13$ coarse grid (16,133 points) is generated from the medium grid using the same process. Figure 4 shows single grid convergence histories of both the Euler and the Navier-Stokes equations on the medium grid. For the present attached flow case, there is not much difference between the two. This suggests that the convergence rate is controlled not by the equations but by the grid.

Figure 5 shows good pressure coefficient agreement between experimental data[23] and the multigrid computations on the

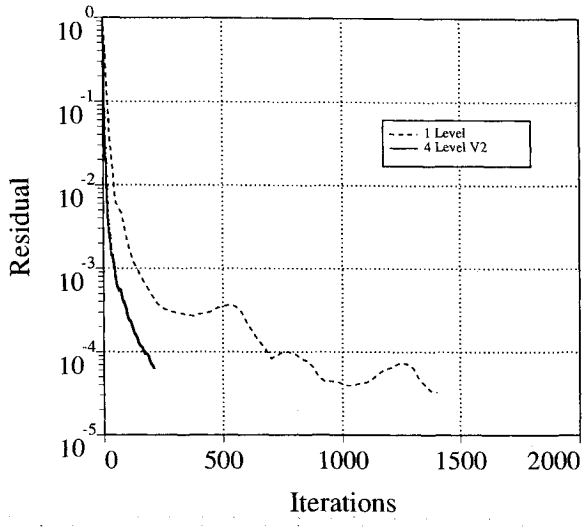

Fig. 9 Convergence histories on the fine grid (iterations).

Table 1 Performance on Cray computers (single processor)

Computer	Grid	CPU[a]	Mflops	Note
YMP	Single	7.4	170	Euler
YMP	Multiple	11.1	160	Euler
C90	Single	3.1	410	Euler
C90	Multiple	4.7	390	Euler

[a]CPU μs per grid point per iteration.

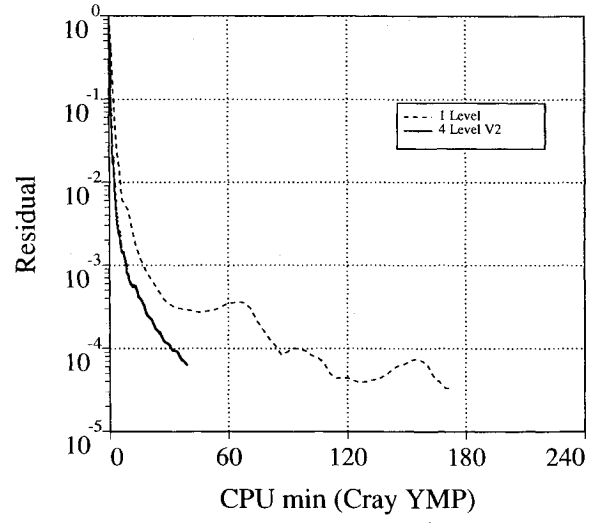

Fig. 10 Convergence histories on the fine grid (CPU time on a Cray YMP).

fine grid at the 44% semispan station. Single grid convergence histories are compared in Fig. 6 and show that the convergence rate slows down significantly as the number of grid points increases. Figure 7 shows that multigrid convergence histories of the three grids are almost identical. Here, the coarse, medium, and fine grids use two-, three-, and four-level V cycles, respectively. The results demonstrate grid independent convergence rates which are achieved by the multigrid method.

To study the effect of grid stretching on the convergence, another fine grid whose first grid distance is 2.0×10^{-5} is generated. Figure 8 shows that the present numerical method is insensitive to grid clustering despite the fact that cell aspect ratios differ by an order of magnitude. The convergence rate on the highly stretched grid appears to be slightly better than the less stretched one for this case. Although not shown here, the rate of convergence slows down significantly after the

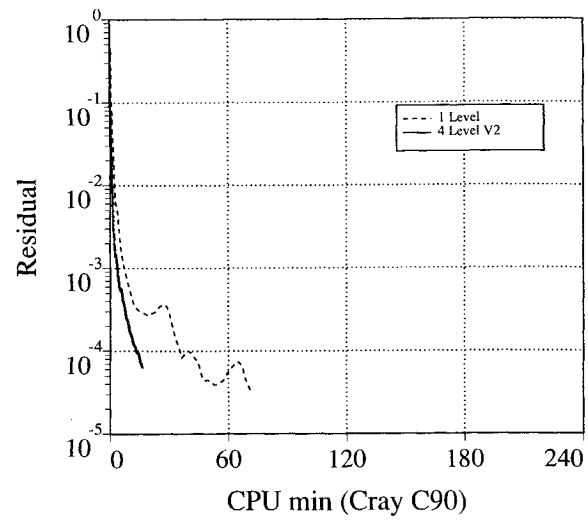

Fig. 11 Convergence histories on the fine grid (CPU time on a Cray C90).

residual drops about five orders of magnitude on the highly stretched grid.

Figure 9 shows that the convergence on the multigrid is about 6.5 times faster than on the single grid in terms of iterations. However, a multigrid cycle requires more time per iteration than a single grid cycle not only because of additional operations for transfer and interpolation but because of the short vector lengths on the coarse grids. As presented in Table 1, the overhead for the multigrid cycle is approximately 50%. Nevertheless, the present CENS3D code requires less than 5 μs per grid point per iteration on a Cray C90 computer. With the single grid mode, the code needs only 3 μs at the sustained rate of 410 Mflops. The CPU times on a Cray YMP and a Cray C90 are compared in Figs. 10 and 11, respectively. The CPU time is reduced by a factor of four by the use of the multigrid method. Figure 11 shows that the residuals for the multigrid drop three orders of magnitude in 50 iterations or 4 CPU min and four orders in 160 iterations or 13 CPU min on a Cray C90.

Conclusions

An efficient three-dimensional multigrid code has been developed for inviscid compressible flows. The lower-upper symmetric Gauss-Seidel implicit scheme is shown to be an effective multigrid driver in three dimensions. The present numerical method appears to be insensitive to grid refinement or to highly clustered grids with high mesh cell aspect ratios. Grid independent convergence rates are achieved by the multigrid method. A reduction of three orders of magnitude in the residual for a three-dimensional transonic inviscid flow using 920 k grid points is obtained in less than 4 min on a Cray C90.

References

[1]Jameson, A., "Solution of the Euler Equations for Two-Dimensional Transonic Flow by a Multigrid Method," *Applied Mathematics and Computation,* Vol. 13, Nov. 1983, pp. 327–356.

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

[3]Martinelli, L., "Calculation of Viscous Flows with a Multigrid Method," Princeton Univ., MAE Rept. 1754T, Princeton, NJ, June 1987.

[4]Vatsa, V. N., and Wedan, B. W., "Development of an Efficient Multigrid Code for 3-D Navier-Stokes Equations," AIAA Paper 89-1791, June 1989.

[5]Radespiel, R., Rossow, C., and Swanson, R. C., "An Efficient Cell-Vertex Multigrid Scheme for the Three-Dimensional Navier-Stokes Equations," AIAA Paper 89-1953, June 1989.

[6]Turkel, E., Swanson, R. C., Vatsa, V. N., and White, J. A.,

"Multigrid for Hypersonic Viscous Two- and Three-Dimensional Flows," AIAA Paper 91-1572, June 1991.

[7]Yoon, S., "Numerical Solution of the Euler Equations by Implicit Methods with Multiple Grids," Princeton Univ., MAE Rept. 1720T, Princeton, NJ, Sept. 1985.

[8]Jameson, A., and Yoon, S., "Multigrid Solution of the Euler Equations Using Implicit Schemes," *AIAA Journal,* Vol. 24, No. 11, 1986, pp. 1737–1743.

[9]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.

[10]Yoon, 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.

[11]Pulliam, T. H., and Chaussee, D. S., "A Diagonal Form of an Implicit Approximate Factorization Algorithm," *Journal of Computational Physics,* Vol. 39, No. 2, 1981, pp. 347–363.

[12]Caughey, D., "Diagonal Implicit Multigrid Algorithm for the Euler Equations," *AIAA Journal,* Vol. 26, No. 7, 1988, pp. 841–851.

[13]Anderson, W. K., Thomas, J. L., and Rumsey, C. L., "Extension and Applications of Flux Vector Splitting to Unsteady Calculations on Dynamic Meshes," AIAA Paper 87-1152, June 1987.

[14]Shuen, J. S., and Yoon, S., "Numerical Study of Chemically Reacting Flows Using a Lower-Upper Symmetric Successive Overrelaxation Scheme," *AIAA Journal,* Vol. 27, No. 12, 1989, pp. 1752–1760.

[15]Park, C., and Yoon, S., "Calculation of Real-Gas Effects on Blunt-Body Trim Angles," *AIAA Journal,* Vol. 30, No. 4, 1992, pp. 999–1007.

[16]Park, C., and Yoon, S., "A Fully Coupled Implicit Method for Thermo-Chemical Nonequilibrium Air at Suborbital Flight Speeds," *Journal of Spacecraft and Rockets,* Vol. 28, No. 1, 1991, pp. 31–39.

[17]Yoon, S., and Kwak, D., "Implicit Navier-Stokes Solver for Three-Dimensional Compressible Flows," *AIAA Journal,* Vol. 30, No. 11, 1992, pp. 2653–2659.

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

[19]Pulliam, T. H., "Artificial Dissipation Models for the Euler Equations," *AIAA Journal,* Vol. 24, No. 12, 1986, pp. 1931–1940.

[20]Swanson, R. C., and Turkel, E., "Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations," AIAA Paper 87-1107, June 1987.

[21]Yoon, S., and Kwak, D., "Artificial Dissipation Models for Hypersonic Internal Flow," AIAA Paper 88-3277, July 1988.

[22]Jou, W. H., Wigton, L. B., Allmaras, S. R., Spalart, P. R., and Yu, N. J., "Towards Industrial-Strength Navier-Stokes Codes," Fifth Symposium on Numerical and Physical Aspects of Aerodynamic Flows, Long Beach, CA, Jan. 1992.

[23]Schmitt, V., and Charpin, F., "Pressure Distributions on the ONERA M6 Wing at Transonic Mach Numbers," AGARD AR-138-B1, 1979.