AIAA JOURNAL
Vol. 40, No. 7, July 2002

Multiresolution Method on General Unstructured Meshes

B. L. Bihari,* D. K. Ota,† Z. Liu,† and S. V. Ramakrishnan†
Hypercomp Inc., Westlake Village, California 91362-7140

Recent advances are presented in the development, implementation, and application of a novel solution adaptive method, the multiresolution (MR) scheme. The MR method uses the difference in information between adjacent grid levels on a set of nested grids for determining active and inactive grid cells. This amounts to computing the wavelet decomposition of the solution, which is known to be a rich source of regularity information. Shocks, contact discontinuities, reaction fronts, or any other inviscid and viscous flow features are identified and tracked in a time-accurate fashion. In active regions, the underlying finite volume scheme is solved in the usual manner, whereas in smooth regions, an inexpensive interpolation of the numerical divergence replaces both flux computations and reconstruction. Thus, the simulation becomes significantly more efficient without any loss of accuracy compared to the finest grid available. The combined finite volume–MR method will be described followed by two-dimensional and three-dimensional time-accurate examples on unstructured meshes.

I. Introduction

SOLUTION adaptive gridding techniques are designed to redistribute evenly the truncation error by adding and removing cells in selected regions and/or moving existing grid points to locations requiring more spatial resolution. They typically rely on some form of first- or, more frequently, second-derivative computation, whose value (in some norm) is then used to flag areas of refinement/derefinement or node movement. In effect, the gradient or Hessian is used to bound the truncation error, which is assumed to decrease when cells become smaller in regions of large variation. There are numerous approaches to the practical implementation of these ideas; Refs. 1–4 are but a small sample.

For general unstructured meshes composed of hybrid cells, the programming effort itself can be considerable in adding/deleting cells. The run-time overhead generally consists of computing Hessians and interpolations from the old grid onto the new one, as well as of redefining points on the boundaries using the original CAD definition. The memory cost associated with adaptation is usually moderate; although when Hessians are stored, the memory overhead can also become significant. As for the spatial and temporal accuracy of these schemes, the effect of cell shape and quality is not yet well understood, but more important, most adaptive schemes are typically limited to steady-state computations.

We propose the multiresolution method as a complementary tool for, or even as a potentially viable alternative to, traditional adaptive gridding approaches. Instead of adding/deleting/moving grid points, in its current form the method uses existing grid cells on a fine grid that is assumed to be given. From this fine grid, several levels of coarser grids are constructed via the agglomeration process. The biorthogonal wavelet decomposition [also known as multiresolution (MR) analysis] introduced in Ref. 5 is then computed on this nested set of grids. Where the MR coefficients are large there is typically some flow feature that requires those cells on that particular resolution level; otherwise, the cells can be deactivated, and the flux computations skipped. Although highly problem dependent, for typical cases, speed-up factors of 2–6 have been obtained. There are some theoretical guarantees and much numerical evidence that the resulting efficiency gain does not compromise accuracy or stability: We get the solution quality of that of the finest available grid. Earlier work[6–8] provides some results on one- and two-dmensional Cartesian meshes for both inviscid and viscous problems. A fairly recent two-dmensional implementation on structured grids using point value MR regularity analysis is presented in Ref. 9, with very promising performance statistics. Applications of MR on unstructured grids had already been reported in Ref. 10, using only two grid levels.

The MR method is fully time accurate and incurs a relatively low computational overhead. The MR decomposition is essentially independent from the underlying solver or even the physics, for example, it has been already used in conjunction with electromagnetics problems. Because of this modularity, it could be a plug-in to existing computational fluid dynamics codes, such as the ICAT code, which is our basic platform. We now give a brief description of the solver and MR method and present numerical results for inviscid and viscous unsteady flows.

II. Brief Overview of the Finite Volume Scheme

We are numerically simulating the compressible Navier–Stokes equations formulated as the conservation law

$$\boldsymbol{q}_t + \nabla \cdot \boldsymbol{F} = \boldsymbol{0}$$

subject to initial conditions at $t = 0$

$$\boldsymbol{q}(\boldsymbol{x}, 0) = \boldsymbol{q}_0(x)$$

For simplicity, we borrowed this compact notation from Refs. 11 and 12, where the divergence is interpreted componentwise. Each element of the flux vector has an inviscid and a viscous component, the latter of which depends on the gradient as well. (For a full description, see Ref. 12).

This system is approximated via a second-order in space and time, explicit finite volume scheme:

$$\boldsymbol{v}_j^{n+\frac{1}{2}} = \boldsymbol{v}_j^n - \frac{\Delta t}{2|C_j|} \sum_{i=1}^{K_j} \boldsymbol{n}_{ij} \boldsymbol{F}_{ij}^n$$

$$\boldsymbol{v}_j^{n+1} = \boldsymbol{v}_j^n - \frac{\Delta t}{|C_j|} \sum_{i=1}^{K_j} \boldsymbol{n}_{ij} \boldsymbol{F}_{ij}^{n+\frac{1}{2}}$$

for each cell j at time step n. In the preceding equation, $\boldsymbol{v}$ is the mean of the solution over each conservation cell, n is the cell face normal at each face i of cell j, K is the number of faces, and $|C_j|$ is the cell volume for the cell. The numerical flux F is evaluated using Roe's (or possibly some other) Riemann solver, given left and

Received 8 May 2001; revision received 5 January 2002; accepted for publication 22 January 2002.

*Technical Staff Member; currently Applied Mathematician, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, L-560, 7000 East Avenue, Livermore, CA 94550-9234. Member AIAA.

†Technical Staff Member, 31255 Cedar Valley Drive, Suite 327. Member AIAA.

right reconstructed values of the cell averages $\boldsymbol{v}$. For the first stage, we use these cell averages themselves (first-order reconstruction), whereas for the second stage a linear interpolation is performed (second-order reconstruction) at location r:

$$\boldsymbol{v}_{L,R} = \boldsymbol{v}_j^{n+\frac{1}{2}} + (\boldsymbol{r} - \boldsymbol{r}_j) \cdot \boldsymbol{G}_j^n$$

using a gradient G computed from Green's theorem:

$$\boldsymbol{G}_j^n = \sum_{i=1}^{K_j} \boldsymbol{n}_{ij} \boldsymbol{v}_{ij}^{*n} \approx \nabla \boldsymbol{v}_j^n$$

where the asterisk superscript indicates an inverse volume-weighted point value at face i of cell j. As an additional option for shocked flows, we have a cell-based minmod limiter, which picks the smallest gradient of all face neighbors for each cell. A more detailed exposition of the numerical scheme may be found in Refs. 11 and 12.

For the Riemann solver, we used Roe's approximate linearization with extra dissipation added for zero-speed contact waves and rarefactions (sonic fix).

III. Multiresolution Scheme

A. Agglomeration

The MR scheme assumes an initial fine grid and a sequence of coarser grids of decreasing resolution obtained by some sort of agglomeration process. Each coarser grid cell is a nonoverlapping union of its finer grid children in a recursive manner:

$$C_j^k = \bigcup_{l=1}^{q_j} C_{jl}^{k-1}$$

for each resolution level $k = 1, \ldots, L$, where q_j is the number of children for cell j and j_l are their actual indices. The agglomeration process for the hypothetical grid of Fig. 1a is shown in Figs. 1b and 1c for two additional coarse grid levels.

Depending on the cell topology, the agglomeration scheme can be trivial, such as for hexahedral cells where 4 (two dimensional) or 8 (three dimensional) cells are grouped together, but could also be very complex and heuristic. In the current study, we used a combination of node-based agglomeration and a variation of the METIS domain decomposition software,[13] which in addition to minimizing subdomain interface surface also maximizes the cell aspect ratio.[14] We found that the latter technique is capable of delivering contiguous groups of acceptable aspect ratios from levels 2 and above (level 0 being the original fine grid). The node-based agglomeration scheme that collects cells around grid points was a necessary modification to the k-way partitioning scheme on level 1 because the approximation to aspect ratios incurs large errors for cells with few faces such as our primary grid composed of prisms or tetrahedra.

Note that the sole purpose of the agglomeration is to construct the nested set of grids such that the fine mesh is known a priori. It is possible to start with a coarse grid and hierarchically subdivide it one or more times to arrive at an appropriate fine grid. In this case, however, subsequent subdivisions of boundary faces will require access to the geometry near the boundaries.

B. MR Analysis

The first stage of the MR scheme is to compute the MR analysis or decomposition (also referred to as encoding), which has two sets of information.

1) The first is the solution $\boldsymbol{v}_j^k$ on each coarse grid level, which is computed via the simple volume-weighted averages:

$$\boldsymbol{v}_j^k = \frac{1}{|C_j^k|} \sum_{l=1}^{q_j} |C_{jl}^{k-1}| \boldsymbol{v}_{l_l}^{k-1}$$

2) The second set comprises the detail components $\boldsymbol{d}_j^k$, which are equal to the difference between the cell average and the value interpolated from the next coarser level:

$$\hat{\boldsymbol{v}}_j^{k-1} = \boldsymbol{I}\left(\boldsymbol{x}_j^{k-1}; \hat{\boldsymbol{v}}_j^k\right), \qquad \boldsymbol{d}_j^{k-1} = \hat{\boldsymbol{v}}_j^{k-1} - \boldsymbol{v}_j^{k-1}$$

For the current second order in space and time scheme, the interpolation I evaluated at cell centroid x_j^{k-1} is performed using a linear interpolation identical to that of the reconstruction outlined in the preceding section: The gradient G is approximated via Green's theorem. The resulting multidimensional polynomial is then evaluated at the cell centroids x_j^{k-1} of the fine grid, instead of the face centroids as was done for flux computations in Sec. II.

C. Truncation

These two sets of outputs from the encoding can also be thought of as low- and high-frequency components, respectively. If the details are all kept, the fine grid values are recoverable exactly without any information loss whatsoever. However, significant data compression is possible if small details are discarded. For our purposes, the size of the detail coefficients is used to flag or clear cells for flux computation: Small values indicate that the cell in question is in a smooth region, and interpolation of the numerical divergence [or right-hand side (RHS)] is sufficient (see Ref. 15 for a more detailed analysis). Each cell on each grid level has a binary flag i_j^k corresponding to its active or inactive status. This second stage is often called truncation or thresholding and is summarized by

$$\text{if } \left(\left|\boldsymbol{d}_j^{k-1}\right| > \varepsilon_k\right) \text{ then : } i_j^k = 1$$

$$\text{else : } i_j^k = 0$$

As justified in detail in Refs. 5 and 15, the truncation parameter used is scaled for each level by

$$\varepsilon_k = \tfrac{1}{2}\varepsilon_{k-1}, \qquad \varepsilon_0 = \varepsilon$$

where ε is on the order of the truncation error and is a given run parameter.

D. Flux Computation

The third stage is where the gain in flux computations is actually realized. This is a multilevel flux computation module that has been modified from the original single level one. Fluxes are first computed at all faces on the coarsest grid, which includes all boundary faces as well. Then on each successive level, fluxes are computed only for faces of flagged cells and added to the numerical divergence or RHS of the cell. This selective flux routine will typically eliminate 50–90% of the total faces from consideration.

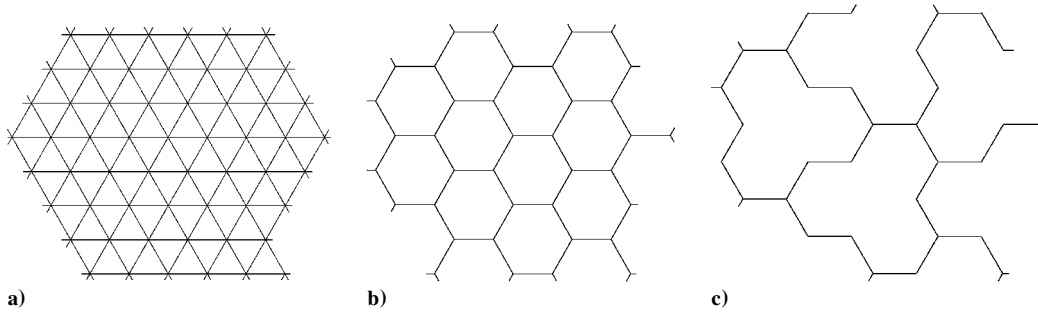


Fig. 1 Original triangular grid and its two agglomerated grids.

E. RHS Computation

At the end of the third stage, there will be some cells for which the RHS values are incomplete or "blank." These are the unflagged cells. These missing values are filled in by interpolation (same I as that described in Sec. III.B) from the next coarser level. It starts from the coarsest grid, which is always complete with actual flux values:

$$\text{if } \left(i_j^k = 1\right) \text{ then : } \mathrm{RHS}_j^{k-1} = \sum_{i=1}^{K_j} \boldsymbol{n}_{ij} \boldsymbol{F}_{ij}$$

$$\text{else : } \mathrm{RHS}_j^{k-1} = \boldsymbol{I}\left(\boldsymbol{x}_j^{k-1};\ \mathrm{RHS}_j^k\right)$$

The underlying assumption is that this interpolation is (much) cheaper than the cost of the necessary flux computations plus that of the reconstruction. For all cases run so far, this was always true. Note that the RHS includes both inviscid and viscous fluxes, as well as turbulent and reactive source terms.

F. Time Update

The RHS computation of the preceding section ultimately builds the RHS on the finest grid level either by using direct flux computations or interpolations. Once this is complete, the time update is performed on the finest grid level using the conventional second-order semidiscrete framework outlined in Sec. II.

IV. Numerical Results

In assessing the performance of the scheme, we typically look at two indicators: the number of active cells and the number of active faces, as measured by their respective data compression. The latter is given by

$$\mu_c = c_{\text{fine}}/c_{\text{MR}}, \qquad \mu_f = f_{\text{fine}}/f_{\text{MR}}$$

for cells and faces, respectively. Here the "fine" vs "MR" subscripts refer to the total number of fine cells or faces vs the number of active cells or faces. Note that the denominator always includes the cells or faces of the coarsest grid, which are all active. The numerator, on the other hand, refers to the original fine grid. The number of boundary faces on each level is the same as on the fine grid, and it is included in the total number of faces.

The ultimate indicator of performance is the run-time speed up, of course. We define it as the ratio of the fine grid run time vs the actual run time of the MR code:

$$\mu = r_{\text{fine}}/r_{\text{MR}}$$

For the present explicit scheme, the actual speed up μ is most closely related to μ_f because the fluxes are face by face and they are the most computationally intensive part of the code. The difference between μ_f and μ_c can also be an indicator of the quality of the agglomeration scheme. If a cell C_j^k has K_j^k number of faces, then we may write

$$c_{\text{MR}} = \sum_{k=0}^{L-1} \sum_{j=1}^{N_k} i_j^{k+1} + N_L$$

$$f_{\text{MR}} = \frac{1}{2} \sum_{k=0}^{L-1} \sum_{j=1}^{N_k} i_j^{k+1} \sum_{i=1}^{K_j^k} \chi_{ij}^k + M_L$$

where i_j^k is the binary flag defined in Sec. III.C, N_L and M_L are the number of cells and faces on the coarsest level, respectively, and χ_{ij}^k is a binary flag indicating whether a face j of cell i has been part of any flagged cells on coarser levels, that is, whether the flux has already been computed. If the agglomeration scheme is a good one, then K_j^k will be small with a constant K_k across all cells j on the same resolution level k. For such an agglomeration the following estimate holds:

$$f_{\text{MR}} \le \frac{1}{2} \sum_{k=0}^{L-1} K_k \sum_{j=1}^{N_k} i_j^{k+1} + \frac{1}{2} K_L N_L + B$$

where B is the number of boundary faces. Clearly, the larger K_k is, the lower the face data compression will be. In the limiting case of no flagged cells, that is, $i_j^{k+1} = 0$, we have $f_{\text{MR}} = \frac{1}{2} K_L c_{\text{MR}} + B$. On the other hand, if $i_j^{k+1} = 1$ everywhere, then $f_{\text{MR}} = \frac{1}{2} K_0 N_0 + B = f_{\text{fine}}$, whereas $c_{\text{MR}} > c_{\text{fine}}$.

We now turn to present several recent applications of the MR scheme to the computation of unsteady inviscid and viscous flowfields.

A. Von Kármán Vortex Street

The classical von Kármán vortex street problem is the first test case for unsteady viscous flow. Because it is subsonic, the only flow features we expect to see are vortices and boundary layers, which are harder to detect and track than sharp discontinuities. We ran this case at a Reynolds number of 700, where the boundary layer is still in the laminar regime, but the unsteady vortex shedding has already commenced. The ICAT code was run using 24,576 hexahedral (three-dimensional) cells (48,960 interior faces) converted from a corresponding single-zone structured grid. The contours of the Mach number are shown in Fig. 2a, and Fig. 2b shows all active cells on a total of five grid levels at the same instance in time as Fig. 2a. The agglomeration in this case was trivial: Each parent cell had four children. This resulted in 6144, 1536, 384, and 96 cells and 24,384, 12,096, 5952, and 2880 interior faces on levels 1–4, respectively. Closeups of Figs. 2a and 2b are shown in Figs. 3a and 3b.

The total number of active cells shown in Fig. 2 was 6964, bringing the cell data compression to 3.53. The face data compression, on the other hand, was 2.63. Note that the cell faces on the coarse levels appear to be curved because they are composed of several small fine grid faces. The MR solution was also compared to the non-MR one, and the two were undistinguishable in terms of contour lines as well as shedding frequency and drag coefficient. The actual run-time speed up as measured over the compute loops was about 2.3, that is, the MR solution was 2.3 times faster. A tolerance value of $\varepsilon = 0.005$ was used in this case.

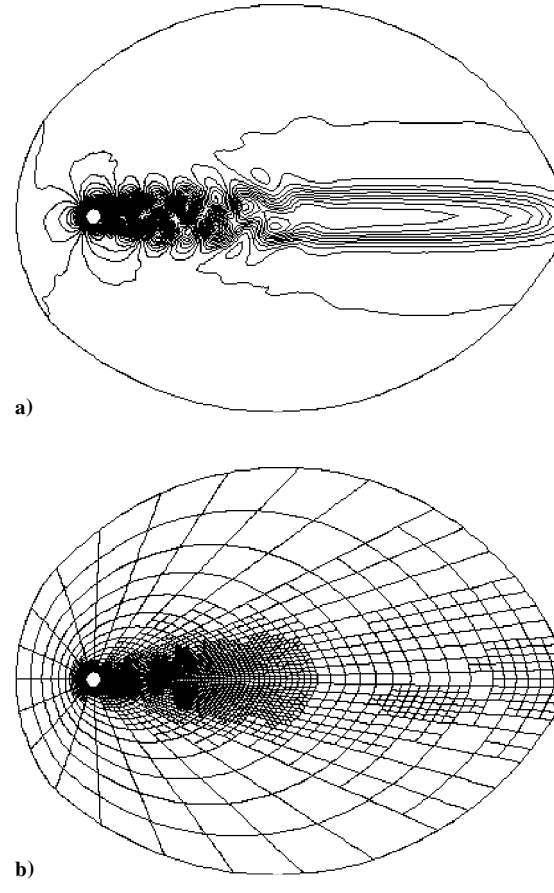
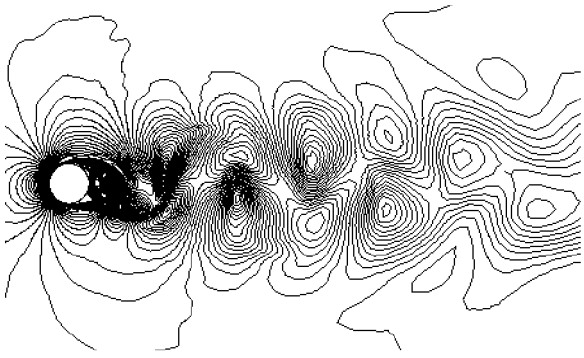
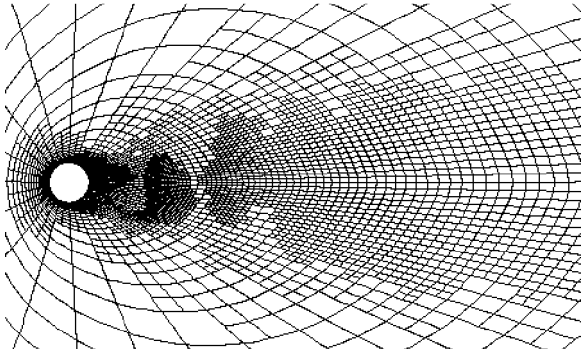


Fig. 2 Snapshot of Mach number contours and active cells for von Kármán vortex street; $Re = 700$.



a)



b)

Fig. 3 Closeup of Figs. 2a and 2b.

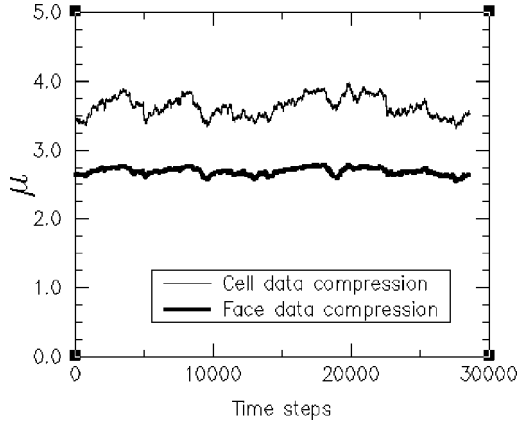


Fig. 4 Ratio of fine grid vs active cells and faces for the von Kármán vortex street.

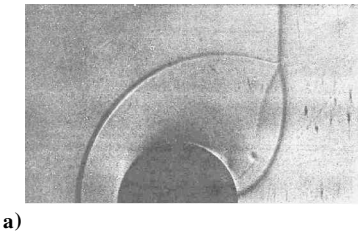
Note that this is in effect a two-dimensional case run in three-dimensional mode because the ICAT code is inherently three dimensional. Although the two sets of symmetry boundary faces on the two sides of this layer of hexahedrons (hexes) were manually eliminated from consideration, all gradient, metric and flux computations were performed in a fully three-dimensional mode. For truly three-dimensional cases we, therefore, expect an even closer correlation between flux data compression and speed up. Also for three-dimensional cases the data compression itself should be higher.

The run-time speed up μ is the average of each instantaneous speed up measured over each time step. A history plot of μ gives us an insight into what is the behavior of the solution and/or the numerical error during the run. Large error accumulations will degrade the speed up, as will development of new flow features. For the current case, we plotted the speed-up log for a time slice taken after shedding already commenced. As expected, Fig. 4 shows a fairly even distribution of μ over time. The cell and face data compressions were fairly close due to the trivially optimal agglomeration and the relatively low overall cell data compression.

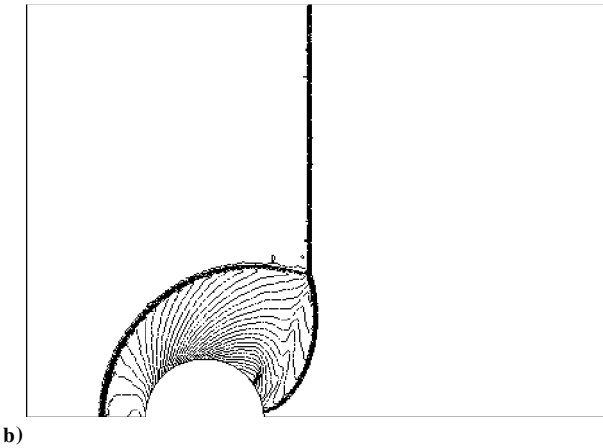
B. Shock Diffraction Around a Cylinder

This case has been extensively studied in Ref. 16 from a theoretical and experimental viewpoint. It involves a Mach 2.82 unsteady planar shock hitting a circular cylinder. For the grid we used a layer of 119,893 triangular prisms, which were not regular but fairly evenly distributed. We used Neumann boundary conditions at the top boundary, inviscid wall on the cylinder surface, symmetry boundary condition (BC) in front of and behind the cylinder, and supersonic outflow at the right exit. For the top boundary we augmented the zero gradient condition with a velocity vector reset for one layer of cells. The alternate method of imposing the exact shock speed there resulted in some numerical error near the top boundary, as it did in Ref. 17, although it did not contaminate the diffractive patterns.

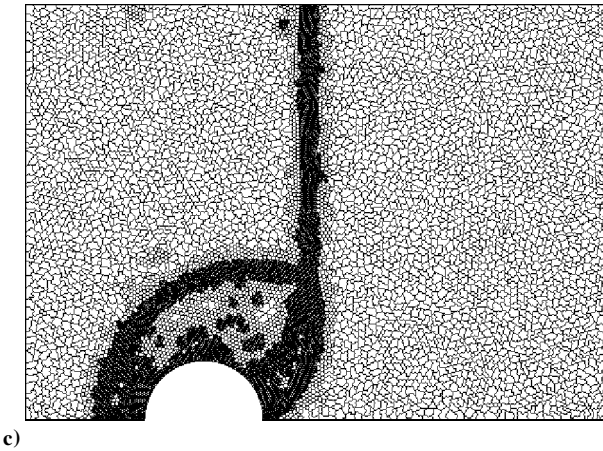
The agglomeration method in this case was the hybrid node-centered/k-way algorithm, which yielded 16,973 cells on level 1 and 4,194 cells on level 2. The number of faces agglomerated from the original 179,405 interior faces became 59,227 and 33,145 on levels 1 and 2, respectively. A model schlieren plot of the solution from Ref. 16 is shown in Fig. 5a. It is clear that at the early stages already the flow pattern is quite complicated with contact discontinuities, Mach shocks, incident and reflected shocks, and vortices forming. Figure 5b shows density contours of the computed MR solution at about the same physical time as in Fig. 5a. The computed solution compares well with the experimental data despite



a)



b)



c)

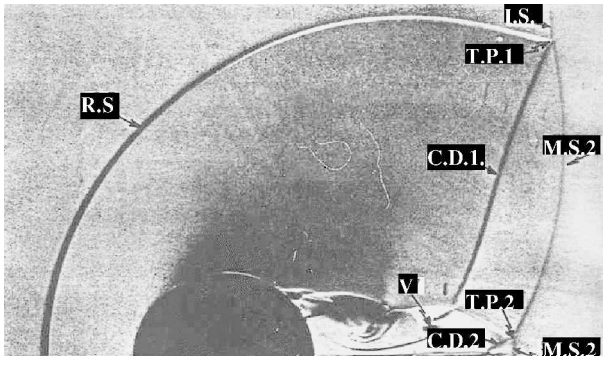
Fig. 5 Schlieren plot, density contours, and active cells for cylinder diffraction at $n = 1400$.

the relatively coarse prismatic mesh. It also compares remarkably well with Ref. 18 (except for the resolution of the vortex), considering the fact that there a fifth-order ENO scheme and RK3 time stepping was used on a quite fine Cartesian mesh.

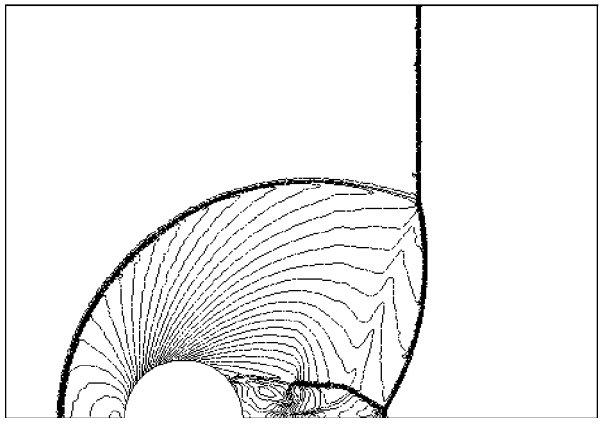
The corresponding MR diagram of Fig. 5c shows all active grid cells on all levels including the coarse level where all cells participate. The cell and face data compressions at this instance were 8.93 and 3.58, respectively. The MR-flagging follows faithfully the flow features as they develop in time. Although the contact discontinuity is difficult to capture and is relatively weak, the MR algorithm is able to identify it.

At a later time, the flow pattern is even more complicated, with several shocks, contact discontinuities, and triple points present. The schlieren diagram (Fig. 6a), shows reflected shock (RS), incident shock (IS), contact discontinuity (CD), Mach shock (MS), triple point (TP), and vortex V, respectively. All of the structures are accurately captured in Fig. 6b, although the CD seems weakly resolved. However, according to the MR diagram of Fig. 6c, the method detected some of the slip line at this later time as well. The cell and face data compressions were 6.79 and 3.13, respectively. There was no noticeable difference between the MR and non-MR solution. (The latter not shown in any of the cases.)

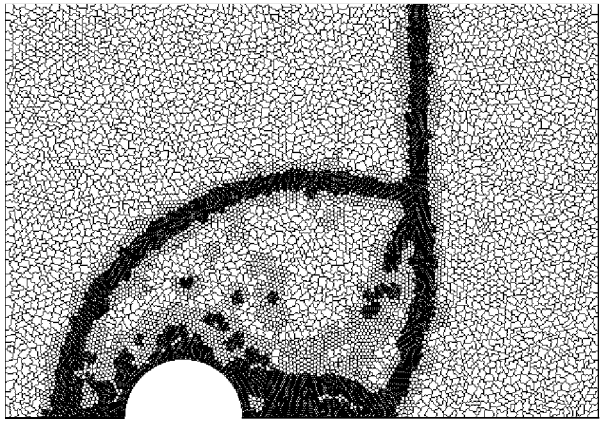
Finally, we show the solution and MR diagram at the instance when the incident shock just exited the flowfield (Figs. 7a and 7b). Again, features are well identified by the MR algorithm. The cell and face data compressions are 5.84 and 2.89, respectively, a slight increase from immediately preceding time steps because one of the discontinuities (the IS) has just left the flowfield. A closeup of the region behind the cylinder (Figs. 7c and 7d) shows that flagged cells on both of the two finer levels are tracking the wake and one of the contact discontinuities (CD2). Note, also, the presence of many



a)

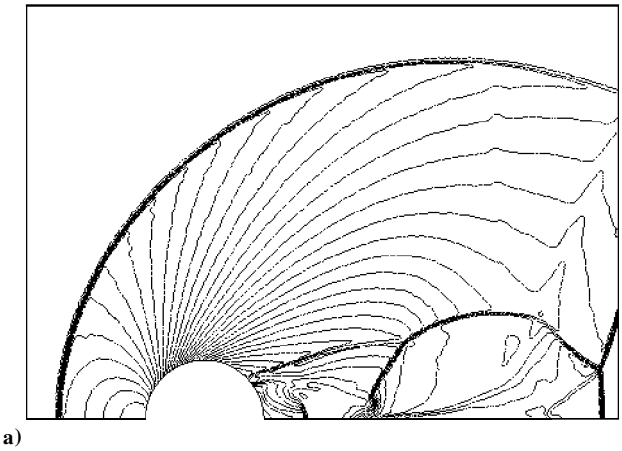


b)

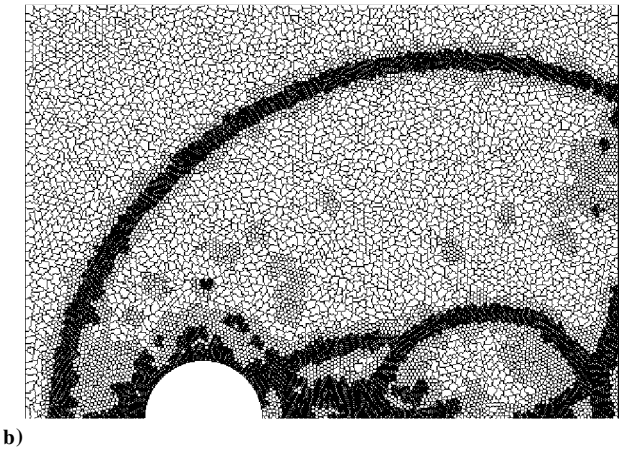


c)

Fig. 6 Schlieren plot, density contours, and active cells for cylinder diffraction at $n = 2400$.

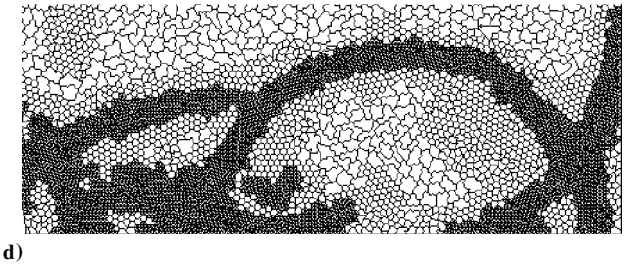


a)



b)

c)



d)

Fig. 7 Density contours and active cells for cylinder diffraction at $n = 3800$; normal and closeup views.

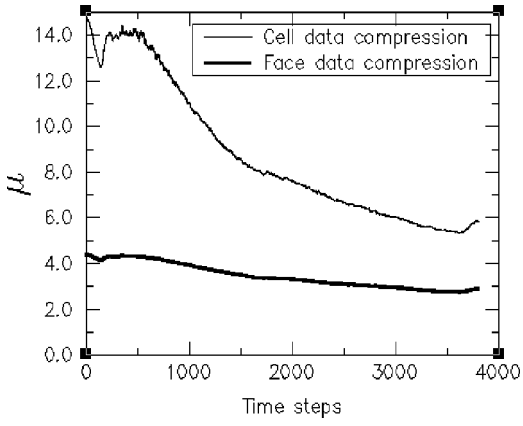


Fig. 8 Ratio of fine grid vs active cells and faces for the shock diffraction over a cylinder.

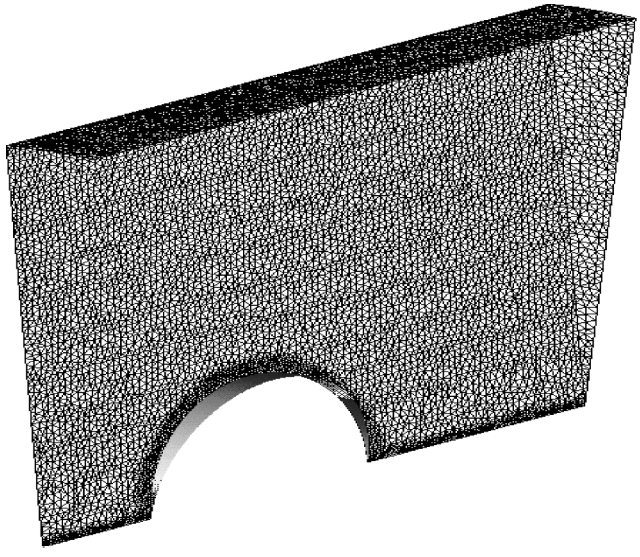


Fig. 9 Sphere geometry and grid setup.

hexagonal cells on level 1 because they are collected around grid nodes, as opposed to the k-way partitioned level 2 cells.

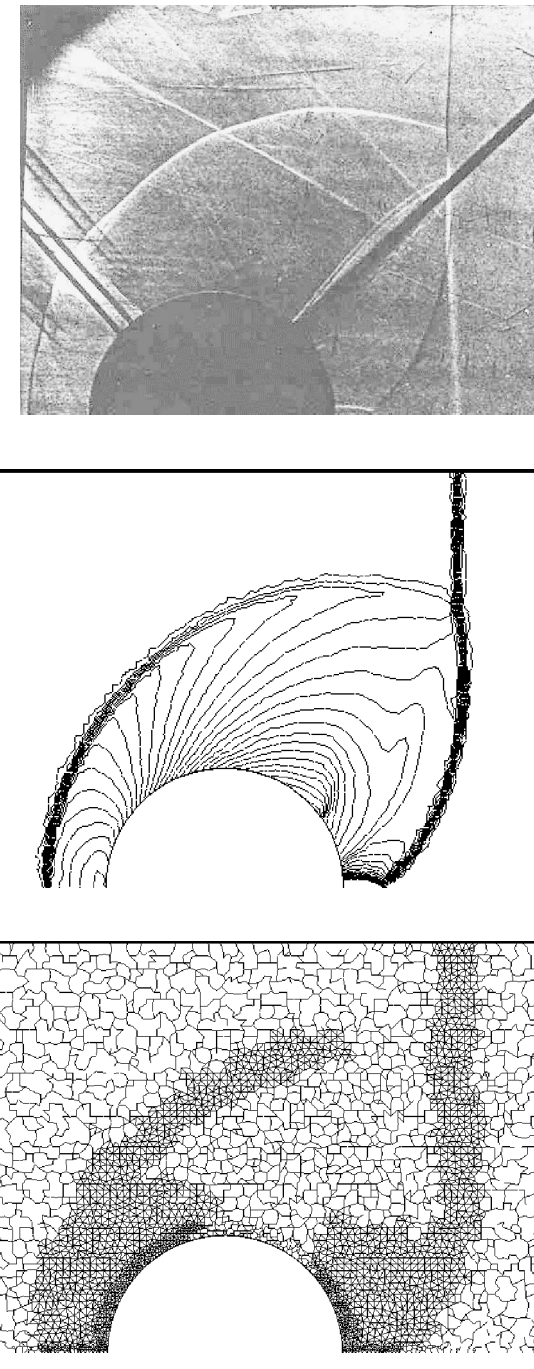
The time history of the data compression plotted in Fig. 8 correlates with the physical evolution of the shock. Whereas μ_f is considerably lower than μ_c, it basically follows the trend of the latter. The average for the face data compression was 3.55 with actual run-time speed up of 2.9. A threshold value of $\varepsilon = 0.01$ was used throughout this computation.

C. Shock Diffraction Around a Sphere

Following the setup of the preceding case and of Ref. 16, we also computed the diffraction pattern over a section of a sphere at the same Mach number. The grid was composed of 362,429 tetrahedral cells (and 710,236 interior faces) in a wedge constructed around a 10-deg section of a sphere (Fig. 9). We reduced the computational domain from that of the cylinder case and roughly doubled the cell size as well. The handling of the boundary conditions was the same. The same agglomeration code was used, which resulted in 15,462 cells and 199,492 interior faces on level 1 and 2,004 cells and 119,245 interior faces on level 2.

The flow structure starts out to be quite similar to that of the cylinder case; however, once the Mach shock leaves the sphere's surface, a reflected shock forms behind the sphere, which now penetrates the Mach shock. A first instance of this focusing phenomenon is shown in Figs. 10a and 10b, by the schlieren plot (the radial lines are nylon strings supporting the sphere) and the density contours.

The MR diagram of Fig. 10c again shows good feature tracking by the flagged cells. We have plotted the intersection of the interior faces of active cells with boundary faces. Note that the agglomerated cells show much more irregularity than in any of the two-dimensional cases. (Even before the agglomeration, the aspect ratios in the original grid are already difficult to control in three dimensions.) It is, in fact, difficult to distinguish between level 1 and level 2 cells on Fig. 10c. It is precisely because of this difficulty in creating good cells on both the original fine and the agglomerated coarse levels that, whereas the cell data compression was 20.69 here, the corresponding value for faces was only 2.65.



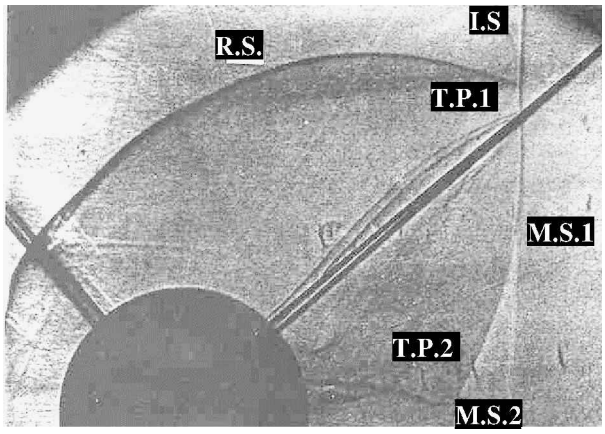
a)

b)

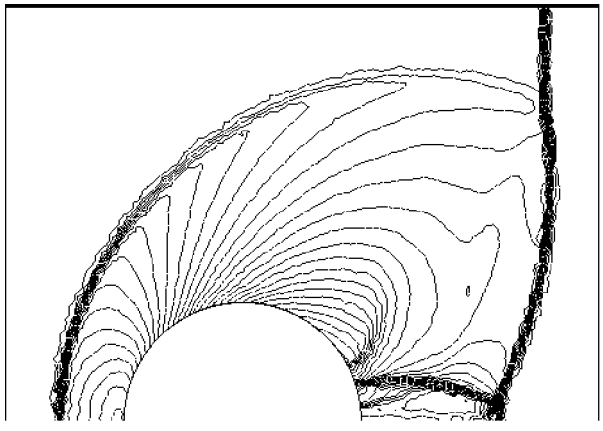
c)

Fig. 10 Schlieren plot, density contours, and active cells for sphere diffraction at $n = 3410$.

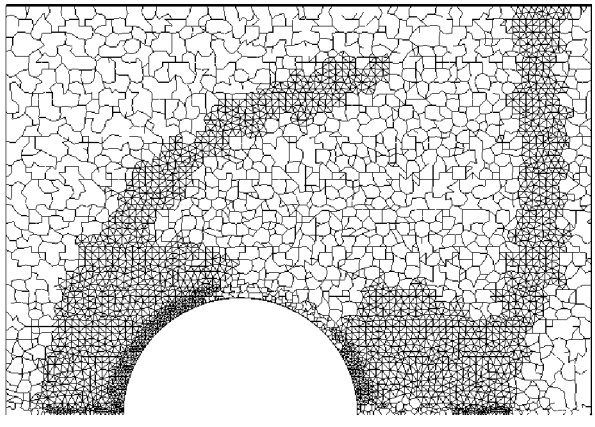
Another snapshot of the experimental data, solution, and MR diagram is shown in Fig. 11. Here the reflected spherical shock behind the sphere has begun to diffuse, and the multiple structures similar to the cylinder case are already developed. The contact discontinuity emanating from the first triple point is barely noticeable on both the photograph and the contour plots. The flow patterns between Figs. 11a and 11b are remarkably similar, even behind the sphere, where new slip lines and vortices are being formed. The MR diagram on the finest grid has started to open up near the symmetry line behind



a)



b)



c)

Fig. 11 Schlieren plot, density contours, and active cells for sphere diffraction at $n = 4290$.

the sphere. The cell and face data compressions at this instance were 18.82 and 2.51, respectively.

The time plot of the data compressions (Fig. 12) shows a similar, but more accentuated, behavior when compared to that of the preceding subsection. We get excellent cell data compression starting with over 50 and an average of over 18 throughout the run, but the face data compression is consistently between 2.5 and 4, with an average of 2.95, the speed up being 2.51. Because of the coarser and lower quality grid than that of the cylinder, we increased the MR thresholding constant to 0.08.

Although the actual runtime speed up was still good with no loss of accuracy, this first truly three-dimensional case underlines the importance of the agglomeration algorithm. The closer the agglomerated cells approximate a sphere, the better the face data compression is, in addition to the obvious benefit of improving their aspect ratio for the numerical approximation itself. Again, note that the

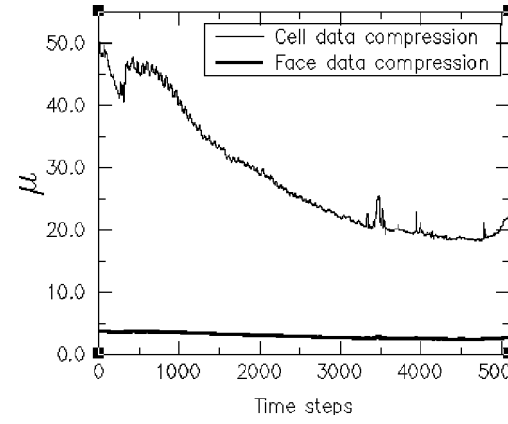


Fig. 12 Ratio of fine grid vs active cells and faces for the shock diffraction over a sphere.

MR detail coefficients are, in essence, interpolation errors. As such, they measure not only irregularities of the flowfield, but those of the grid as well.

V. Conclusions

We continued the development of the MR scheme on general unstructured meshes with new applications to unsteady inviscid and viscous flows, including the first three-dimensional case. The method continues to perform well for a variety of flow physics and grid topologies; our present tests included hexahedral, prismatic, and tetrahedral cells. Although it has already been used for steady-state problems (for example, Ref. 17), its true power is more obvious for unsteady applications, for which, to our knowledge, there is no comparably performing adaptive grid method.

The present study also hints at the yet untapped potential of the MR method. With cell data compressions of over 10 and at times around 50, for cell-intensive solvers such as implicit schemes, the MR scheme could offer an even higher payoff. Our present explicit scheme has a face-based loop structure whose most expensive ingredient is the flux computation, and thus, the speed up is bound by the face data compression.

It is clear that the quality of the agglomeration and the original fine grid is very important for multiresolution efficiency. This sensitivity might be reduced if the creation of the grid hierarchy was accomplished by a global refinement scheme instead. Both this idea and the implicit implementation of MR schemes will be investigated in future research.

Acknowledgments

We gratefully acknowledge the support of The Boeing Company for sponsoring most of this research. The authors express their appreciation to Irene Moulitsas of the University of Minnesota for making the agglomeration version of Metis available to us in source code form. We also wish to thank Touraj Sahely for creating the tetrahedral mesh for the sphere section.

References

[1]Shepard, M. S., Flaherty, J. E., de Cougny, H. L., Ozturan, C., Bottasso, C. L., and Beall, M. W., "Parallel Automated Adaptive Procedures for Unstructured Meshes," *AGARD-FDP-VKI Special Course on Parallel Computing in CFD*, Rept. R-807, AGARD, May 1995, pp. 6-1–6-49.

[2]Kallinderis, Y., and Vijayan, P., "Adaptive Refinement-Coarsening Scheme for Three-Dimensional Unstructured Meshes," *AIAA Journal*, Vol. 31, No. 8, 1993, pp. 1440–1447.

[3]Liu, F., Ji, S., and Liao, G., "An Adaptive Grid Method and Its Applications to Steady Euler Flow Calculations," *SIAM Journal on Scientific and Statistical Computing*, Vol. 20, No. 3, 1998, pp. 811–825.

[4]Tam, A., Robichaud, M. P., Tremblay, P., Habashi, W. G., Hohmeyer, M., Guevremont, G., Peeters, M. F., and Germain, P., "Three-Dimensional Adaptive Anisotropic Method for External and Internal Flows," AIAA Paper 98-0771, 1998.

[5]Bihari, B. L., and Harten, A., "Multiresolution Schemes for the Numerical Solution of 2-D Conservation Laws, I," *SIAM Journal on Scientific Computing*, Vol. 18, No. 2, 1997, pp. 315–354.

[6]Bihari, B. L., and Harten, A., "Application of Generalized Wavelets: An Adaptive Multiresolution Scheme," *Journal of Computational Applied Mathematics*, Vol. 61, No. 3, 1995, pp. 275–321.

[7]Bihari, B. L., "Multiresolution Scheme for Conservation Laws with Viscosity," *Journal of Computational Physics*, Vol. 123, No. 1, 1996, pp. 207–225.

[8]Bihari, B. L., and Schwendeman, D., "Multiresolution Schemes for the Reactive Euler Equations," *Journal of Computational Physics*, Vol. 154, 1999, pp. 197–230.

[9]Chiavassa, G., and Donat, R., "Point-Value Multiscale Algorithms for 2D Compressible Flows," *SIAM Journal on Scientific Computing*, Vol. 23, No. 3, 2001, pp. 805–823.

[10]Abgrall, R., "Multiresolution in Unstructured Meshes: Application to CFD," *Numerical Methods for Fluid Dynamics*, Vol. 5, Oxford Univ. Press, Oxford, 1996, pp. 271–277.

[11]Bihari, B. L., Ramakrishsnan, S. V., Shankar, V., and Palaniswamy, S., "Massively Parallel Implementation of an Explicit CFD Algorithm on Unstructured Grids," *15th International Conference on Numerical Methods in Fluid Dynamics, Proceedings*, Springer-Verlag, 1997, pp. 438–443.

[12]Bihari, B. L., Shankar, V., and Palaniswamy, S., "Massively Parallel Implementation of an Explicit CFD Algorithm on Unstructured Grids, II," *Parallel Computational Fluid Dynamics*, Elsevier, New York, 1998, pp. 241–248.

[13]Karypis, G., and Kumar, V., "Metis: Unstructured Graph Partitioning and Sparse Matrix Ordering System," Dept. of Computer Science, TR 96-036, Univ. of Minnesota, Minneapolis, MN, 1996.

[14]Moulitsas, I., and Karypis, G., "Multilevel Algorithms for Generating Coarse Grids for Multigrid Methods," Dept. of Computer Science, TR 01-021, Univ. of Minnesota, Minneapolis, MN, 2001; also *Proceedings of Supercomputing 2001 Conference* (to be published).

[15]Harten, A., "Multiresolution Algorithms for the Numerical Solution of Hyperbolic Conservation Laws," *Communications on Pure and Applied Mathematics*, Vol. 48, 1995, pp. 1305–1342.

[16]Bryson, A. E., and Gross, R. W., "Diffraction of Strong Shocks by Cones, Cylinder, and Spheres," *Journal of Fluid Mechanics*, Vol. 10, No. 1, 1961, pp. 1–22.

[17]Bihari, B. L., "Solution Adaptation on Unstructured Grids via the Multiresolution Method," *Numerical Grid Generation in Computational Field Simulations*, edited by B. K. Soni, J. Hauser, J. F. Thompson, and P. Eiseman, International Society of Grid Generation, Whistler, BC, Canada, 2000, pp. 511–520.

[18]Xu, S., Aslam, T., and Stewart, D. S., "High Resolution Numerical Simulation of Ideal and Non-ideal Compressible Reacting Flows with Embedded Internal Boundaries," *Combustion Theory and Modeling*, Vol. 1, 1997, pp. 113–142.

J. Kallinderis
Associate Editor