JOURNAL OF COMPUTATIONAL PHYSICS 117, 121–131 (1995)

An Accuracy Assessment of Cartesian-Mesh Approaches for the Euler Equations

WILLIAM J. COIRIER

NASA Lewis Research Center, Cleveland, Ohio 44135

AND

KENNETH G. POWELL

The University of Michigan, Ann Arbor, Michigan 48109-2140

Received May 26, 1993; revised August 15, 1994

A critical assessment of the accuracy of Cartesian-mesh approaches for steady, transonic solutions of the Euler equations of gas dynamics is made. An exact solution of the Euler equations (Ringleb's flow) is used not only to infer the order of the truncation error of the Cartesian-mesh approaches, but also to compare the magnitude of the discrete error directly to that obtained with a structured mesh approach. Uniformly and adaptively refined solutions using a Cartesian-mesh approach are obtained and compared to each other and to uniformly refined structured mesh results. The effect of cell merging is investigated as well as the use of two different *K*-exact reconstruction procedures. The solution methodology of the schemes is explained and tabulated results are presented to compare the solution accuracies.

I. INTRODUCTION

With the advent of unstructured meshes, it is becoming possible to perform high quality calculations of flows of increasing geometric and physical complexity. This geometric flexibility is obtained by using an unstructured grid data structure that, if formulated properly, can allow mesh enrichment by cell division. In this way, unstructured solvers with adaptive mesh refinement can resolve disparate length scales on geometrically complicated domains and perhaps provide a means to achieve automatic mesh convergence. Mesh redistribution schemes have the benefit of being able to use existing, structured mesh flow solvers with few modifications, but suffer from the constraints borne by the structured mesh data structure. As pointed out in [1], it can be problem dependent whether mesh enrichment or redistribution is better, although both schemes can be an improvement to the non-adaptive approach. Through the proper formulation of data structures and by an efficient implementation of non-traditional algorithms, the use of unstructured meshes approaches can complement standard, structured-mesh approaches.

The method assessed here is a Cartesian-mesh approach. Cartesian-mesh approaches have been in the literature for a number of years. In [2, 4] unsteady shock hydrodynamic problems were computed on a Cartesian mesh on a Cartesian domain. Adaptive mesh refinement was achieved by adding collections of cells, grouped into contiguous grids about fronts in the field, using front detection algorithms based upon pattern recognition. In [5], a similar approach was used to compute highly resolved, unsteady shock hydrodynamic problems, where the grids used were based upon a background, body-fitted mesh. By using unique data structures coupled with non-traditional algorithms, these approaches were able to compute highly resolved flows with multiple fronts and length scales.

Cartesian-mesh-based approaches have been used with great success for computing flows about complicated geometries by solving the full potential equation in a finite-element formulation in [6]. In this application, full use was made of the underlying hierarchical structure of the grid through the use of an octree-based data structure. The Cartesian-mesh approach was extended to solving the Euler equations in a finite volume formulation in [7] for computing the transonic flow about multi-element airfoils, although no mesh refinement was performed. The basic premise behind the Cartesian-mesh approach is the use of a regular, background mesh composed entirely of Cartesian cells to allow a local description of arbitrarily complex boundaries. This results in either irregularly shaped cells at the boundaries (for a finite volume approach) or special boundary procedures (for a finite difference approach). For the finite-volume formulation, it is preferred to "cut" the geometry out of the background mesh, resulting in irregularly shaped cells upon which a flux balance must be performed. It is the treatment of the boundaries and the resulting irregularly shaped cells that has been a pacing item for the finite volume form of this technique. In [8], a unique method of cell cutting for the finite volume variant

0021-9991/95 $6.00

was shown and impressively demonstrated for an unsteady, geometrically complicated flow.

In [3, 8–13] reliance upon a locally structured background mesh was used for gradient reconstruction and to help formulate the flux balance on irregular boundaries and coarse/fine mesh interfaces. This close coupling to the background structured mesh is advantageous when refinement is made by adding grid patches. In [14] a linear reconstruction technique (Green–Gauss reconstruction) was used on the Cartesian-cell mesh. This reconstruction technique is locally second-order accurate and is used in many unstructured mesh approaches. This decoupling of the reconstruction process from the flux computation sets this approach apart from the previous Cartesian-cell work in that a true decoupling from the background mesh was achieved, resulting in what at first appears to be an unstructured mesh approach. But, by exploiting the hierarchical properties obtainable through the genesis of the mesh, the resulting approach can be more aptly named a hierarchically structured, Cartesian-mesh approach. This approach was applied successfully to investigate many interesting flow fields [15, 18, 21], but the accuracy of the approach has not been benchmarked. Since this approach would like to promise automatic mesh generation along with automatic mesh convergence, the accuracy of the approach needs to be closely examined.

This study performs a critical assessment of a Cartesian-mesh approach by using an analytical solution to the Euler equations (Ringleb's flow) and directly compares the solution accuracy to that obtained with a structured mesh calculation. Uniform mesh refinement is performed for both the structured and Cartesian approaches. Adaptive refinement via mesh enrichment for the Cartesian grids is also made and compared to the uniformly refined results. The framework of using an exact, analytic solution to the Euler equations allows other important procedures to be quantitatively analyzed. Two linear reconstruction procedures (Green–Gauss and minimum-energy) are evaluated and the approach of cut-cell data merging is outlined and examined in the framework of Ringleb's flow.

II. A CARTESIAN-MESH APPROACH

The approach presented here solves the Euler equations of compressible fluid dynamics using a cell centered, finite volume, upwind scheme. A linear reconstruction of the primitive variables is used to determine the state quantities at cell to cell interfaces which are then used to compute the flux using an approximate Riemann solver. Typically a coarse, base mesh is generated, upon which a solution is obtained. Then, based upon this (coarse) solution, the mesh is adaptively refined, and a new solution is found on this refined mesh. This process is repeated a number of times, solving and then refining, until a specified degree of adaptation is achieved. This automatic, adaptive mesh refinement, coupled with the automatic mesh generation, attempts to gain grid converged solutions with minimal user intervention. The following sections briefly outline the approach.

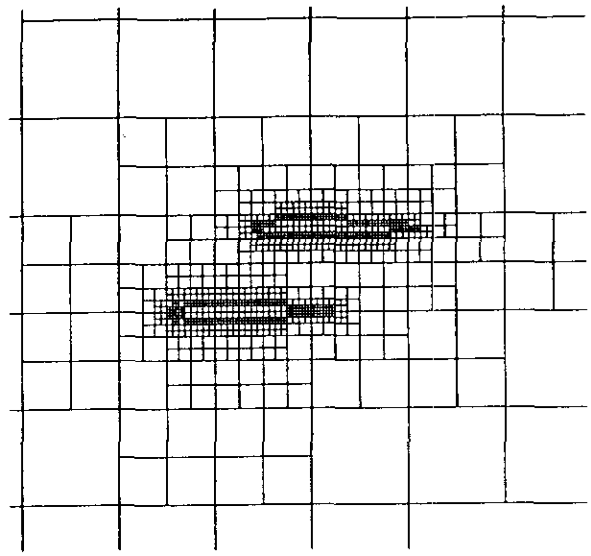

FIG. 1. Staggered biplane mesh: Prior to cell cutting.

II.a. Mesh Generation/Data Structure

One objective of this approach is to create a mesh generation procedure that can be automated for complex geometries. In addition, the resulting method and data structure should easily allow for mesh refinement and mesh coarsening. The approach presented here is based upon using Cartesian cells of unit aspect ratio that are "cut" into irregular cells whenever the boundary of a body intersects any of its surfaces. By using Cartesian cells of unit aspect ratio and by splitting these cells into four children the resulting mesh is highly uniform and smooth in uncut regions. The mesh generation proceeds automatically, once the geometry is suitably defined, by finding the intersections of the body surfaces with each cell boundary and by determining whether these intersections (if any) satisfy a simple set of cell cutting rules. If a cell is intersected, and it does not satisfy the cell cutting rules, it is refined (split into four cells). This proceeds recursively over all current cells in the mesh, only refining the offending cells, until the resulting mesh is suitable for the cutting of the geometry. Arbitrary numbers of bodies can be used to form the geometry, where each body can be represented by discrete (pointwise data) or arbitrary functional descriptions. The staggered biplane configuration computed in [7] is used here to illustrate the mesh generation process. Figures 1 and 2 show a portion of the domain before and after cutting the geometry out of the background mesh.

As opposed to using an array or some other approach using "flat-lists," a hierarchical data structure is used to store the mesh. The data structure is based upon a binary tree, where each node of the tree represents a cell at some stage of the cell spawning process. Figure 3 is an illustration showing the stages

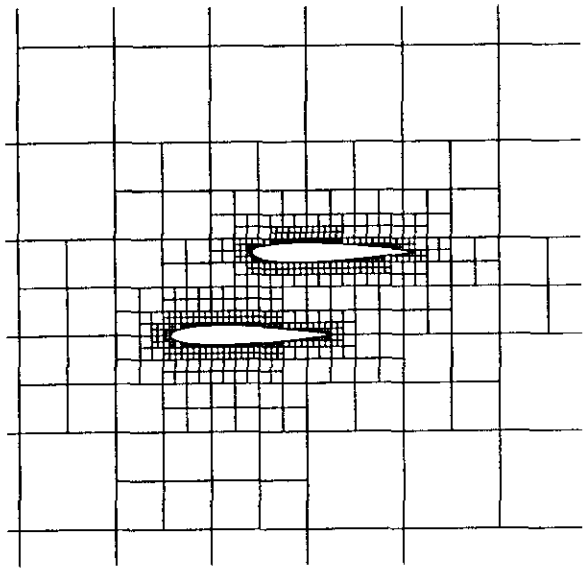

FIG. 2. Staggered biplane mesh: Immediately after cell cutting.

of splitting a Cartesian cell isotropically into four children. Referring to this figure, consider cell A, which is first split in x. This results in two cells, B and C, that are the children of A. If these are then split in y, the result is a set of four children, cells D, E, F, and G, that are contained geometrically within the parent cell, A, and hierarchically below A in the tree.

The data structure mimics this hierarchical nature by having a pointer to the cell parent and two pointers to its children, in addition to geometric data (cell centroid and cell size) and tree level data. It should be noted that if the mesh were everywhere Cartesian, all the geometric data could be directly inferred from the tree. Another useful property of the tree data structure is that cell-to-cell connectivity can be inferred from the tree via logical tree traversals based on centroid compares and face-matching procedures.

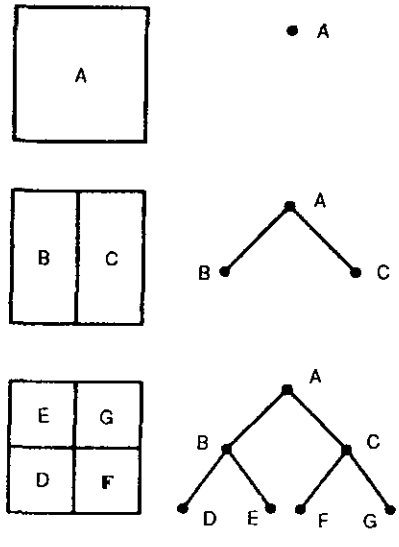

FIG. 3. Illustration of cell splitting tree structure.

II.b. Flow Solver Formulation

The Euler equations are solved using a cell centered, finite volume, upwind based scheme. Roe's approximate Riemann solver [20] is used to formulate the equivalent one-dimensional flux through the cell-to-cell interfaces. The states at these interfaces are found using a two-dimensional interpolation of cell primitives that is formed using linear reconstruction. Two reconstruction procedures are explained and investigated here; Green–Gauss and minimum-energy. The resulting schemes, when applied in one-dimension, are similar to Fromm's scheme. A generic multi-stage scheme is used to drive the residuals to zero. The solution methodology follows the standard finite-volume approach, in that the procedure can be broken into three separate stages: reconstruction, flux computation, and evolution. A brief outline of each segment is given below.

Reconstruction

The primitive variables are reconstructed in each cell using a linear reconstruction. The process of reconstruction to arbitrary degrees of accuracy on unstructured meshes is presented in detail by Barth in [16, 17] and by Godfrey *et al.* in [19], of which a distilled presentation of the minimum-energy reconstruction is shown below. The process of reconstruction can be considered as the discrete inverse of a cell-averaging process. In other words, given the distribution of a function within a cell, one can compute the cell average by

$$\bar{u} = \frac{1}{A}\int_A u(x, y)\, dA = A_n[u(x, y)]. \tag{1}$$

Reconstruction solves the inverse of this problem: find the expansion about the cell centroid to kth order, $u^k(x, y)$, using the cell averaged data of the cell to be reconstructed and a set of support cells. The support of cells used in the reconstruction is typically taken to be nearest neighbor cells; that is cells that share a face and/or a vertex with the cell where the reconstruction is desired.

Minimum-Energy Reconstruction. By expanding $u^k(x, y)$ in terms of zero mean basis polynomials, conservation of the mean of the object cell is ensured, resulting in the general expansion

$$u^k(x, y) = \bar{u} + \sum_j \alpha_j \psi_j(x, y), \tag{2}$$

where the ψ_j are constructed such that their cell average is identically zero. The reconstruction process finds the α_j above by minimizing the least square error, S, with respect to the neighbor cell averages of the u^k. That is, minimize with respect to the α_j,

$$S = \sum_n \omega_n [A_n (u^K(x, y)) - \bar{u}_n]^2. \tag{3}$$

This results in a linear system for the α_j,

$$L_{ij}\alpha_j = b_i, \tag{4}$$

where

$$L_{ij} = \sum_n \omega_n A_n [\Psi_i(x, y)] A_n [\Psi_j(x, y)] \tag{5}$$

$$b_i = \sum_n \omega_n (\bar{u}_n - \bar{u}) A_n [\Psi_i(x, y)]. \tag{6}$$

For a given mesh, the L_{ij} is only dependent upon the geometry, so it can be inverted beforehand. This preprocessing of the reconstruction makes it efficient in that only a simple sum is needed to compute the reconstructed solution. For the work considered here, the expansion is taken out only to $K = 1$, resulting in a linear expansion, with the meaningful expression for the reconstruction

$$u^{K=1} = \bar{u} + u_x(x - \bar{x}) + u_y(y - \bar{y}). \tag{7}$$

It should be pointed out that this reconstruction needs no special ordering of points, requires a minimum of two neighbors, and is obtained by a simple summation over the support cells. This results in a second-order accurate reconstruction of the local solution in a cell and reconstructs linear functions exactly. Unless noted otherwise, all calculations were computed using this reconstruction procedure.

Green–Gauss Reconstruction. Green–Gauss reconstruction is another type of linear reconstruction that is commonly used in unstructured grid solvers. Green's theorem applied to a scalar function relates the volumetric integral of the gradient of the function to its surface integral over the surface of the bounding volume. If the gradient is assumed to be constant over the cell the reconstructed gradient can be found as

$$\nabla u = \frac{1}{A} \oint_\Gamma u\hat{n}\, d\Gamma. \tag{8}$$

The line integral in (8) is ordered counterclockwise, which requires the neighbor cell averaged data also to be ordered counterclockwise. The line integral is computed using second-order Gaussian quadrature which results in a summation over the (ordered) neighbor cells. The reconstruction obtained is second-order accurate and reconstructs linear functions exactly. A truncation error analysis of this reconstruction on a uniform mesh using all order one neighbors indicates a slightly lower leading truncation error than the minimum-energy reconstruction.

Flux Computation

The Euler equations are solved using a standard, cell centered, finite-volume scheme. After converting the divergence of the fluxes into a surface integral, the time rate of change of the cell averages of the conserved variables is

$$\frac{\partial \bar{q}}{\partial t} = -\frac{1}{A} \oint_S F \cdot \hat{n}\, dS. \tag{9}$$

This surface integral is approximated using second-order Gaussian quadrature. It is pointed out in [17] that it is necessary to use a quadrature of at least the same order as the reconstruction order and that the accuracy achieved using a quadrature of higher order does not warrant the extra effort. Second-order Gaussian quadrature results in evaluating the kernel of (9) at the midpoints of all the edges representing the mesh. Replacing the right-hand side of (9) with the numerical quadrature, the semi-discrete form of the equations is reduced to

$$\frac{\partial \bar{q}}{\partial t} = -\frac{1}{A} \sum \Phi(U_L, U_R)\, \Delta S. \tag{10}$$

The sum is taken over all faces of the cell, and the left and right states of the Riemann solver are found using the reconstructed solutions in each cell evaluated at the Gauss points. The numerical flux, Φ, is formed in the face normal coordinate system in a standard upwind formulation, where the numerical flux difference is formed using Roe's linearized Riemann solver.

Evolution

Since obtaining steady-state solutions is the goal, a spatially varying time step is used by advancing all cells in the mesh at a constant Courant number. This is quite necessary since cell cutting and mesh adaptation can result in widely varying cell areas across the mesh. A three-stage explicit multi-stage scheme is used to converge the solution. For the semi-discrete form

$$\frac{\partial q}{\partial t} = R(q) \tag{11}$$

the update to the conserved variables is performed

$$\begin{aligned} q^{(0)} &= q^n \\ q^{(1)} &= q^{(0)} + \lambda_1 \Delta t R(q^{(0)}) \\ q^{(2)} &= q^{(0)} + \lambda_2 \Delta t R(q^{(1)}) \\ q^{(3)} &= q^{(0)} + \lambda_3 \Delta t R(q^{(2)}) \\ q^{n+1} &= q^{(3)} \end{aligned} \tag{12}$$

with stage coefficients $\lambda_1, \lambda_2, \lambda_3$ of (0.18, 0.5, 1.0) and a Courant number of 1.3.

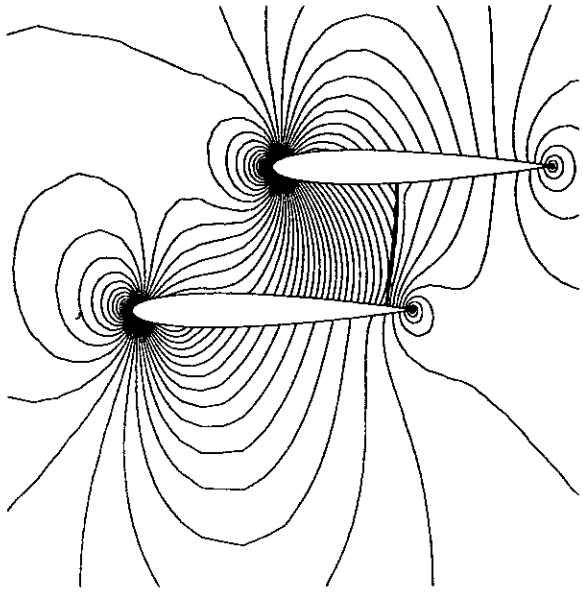

FIG. 4. Pressure contours: Staggered biplane configuration.

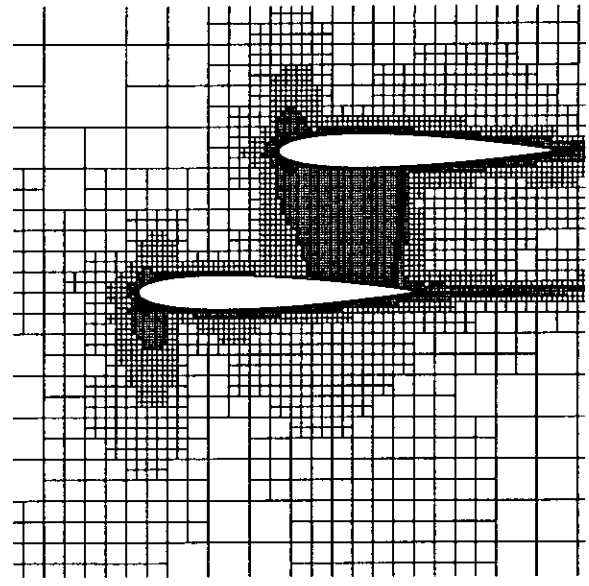

FIG. 5. Adapted mesh: Staggered biplane configuration.

II.c. Mesh Adaptation

Adaptive mesh refinement is achieved using the refinement criteria presented in [21, 15]. Briefly, refinement is based upon a statistical description of two refinement parameters that characterize the local compressibility and rotationality of the fluid. These serve as indicators of important fluid processes which need to be resolved and are not used to specifically reduce the truncation error of the discrete solution. These parameters are

$$\tau_r = |\nabla \times u| l^{3/2} \tag{13}$$

$$\tau_c = |\nabla \cdot u| l^{3/2}, \tag{14}$$

where the characteristic cell length is taken to be $l = \sqrt{A}$. Cells are refined that have cell lengths greater than some user-specified value and either refinement parameter is greater than its standard deviation from zero. Cells are a candidate for coarsening if both parameters are less than some fraction of the standard deviation from zero (taken here to be one-tenth).

The following figures illustrate the Cartesian-mesh approach for the staggered biplane configuration computed in [7]. The two elements are NACA 0012 airfoils, staggered $\frac{1}{2}$ chord length in the chordwise and pitch directions. The free stream Mach number is $M_\infty = 0.7$ and the configuration is at $\alpha = 0°$. The computation is made through four levels of refinement beyond the base mesh. Figures 4 and 5 show a portion of the adapted flow field and mesh.

All structured and Cartesian calculations reported in this study were converged to a drop in the L_2 norm of the residual of the continuity equation of at least six orders of magnitude. All of the calculations were performed on IBM RS6000 workstations. The aggregate processing rates for the Cartesian-mesh calculations on a Model 560 were approximately 200 μs/cell/iteration for the Ringleb's flow calculations.

III. ACCURACY ASSESSMENT: RINGLEB'S FLOW

Ringleb's flow is a hodograph solution to the Euler equations [22] and has been used to assess the accuracy of other structured- and unstructured-mesh approaches [17, 19, 23]. A variety of flows can be attained, depending upon the choice of parameters used. The solution is parameterized in terms of the total velocity, q, and streamline constant, k, as

$$x(q,k) = \frac{1}{2\rho}\left(\frac{2}{k^2} - \frac{1}{q^2}\right) - \frac{J}{2} \tag{15}$$

$$y(q,k) = \pm\frac{1}{k\rho q}\sqrt{1-(q/k)^2} \tag{16}$$

$$c = \sqrt{1-((\gamma-1)/2)\,q^2} \tag{17}$$

$$\rho = 2/c^{(\gamma-1)} \tag{18}$$

$$J = \frac{1}{c} + \frac{1}{3c^3} + \frac{1}{5c^5} - \frac{1}{2}\log\left(\frac{1+c}{1-c}\right), \tag{19}$$

where the density, ρ, is made non-dimensional by its stagnation value and all speeds are made non-dimensional by the stagnation sound speed. The flow angle θ is related to the streamline constant and total velocity by

$$\theta = 2\pi - \sin^{-1}(q/k). \tag{20}$$

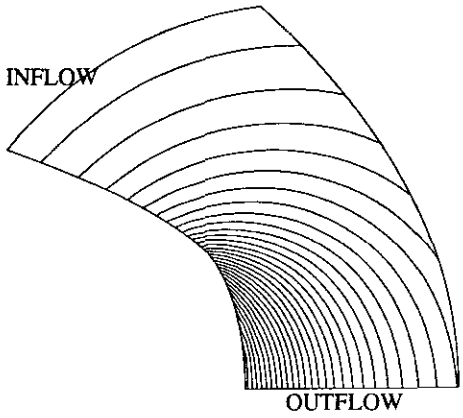

FIG. 6. Mach number contours of Ringleb's flow.

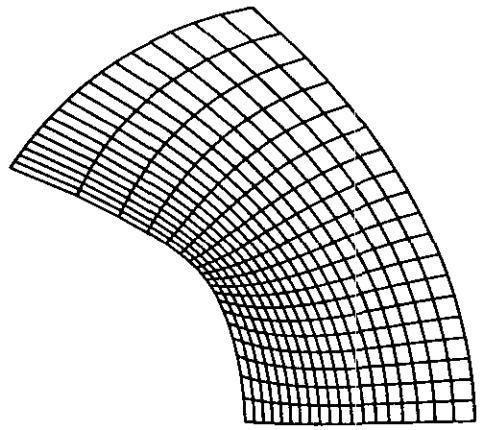

FIG. 7. Structured mesh: 400 cells.

The flow in the first quadrant is computed that is bounded by the streamlines $k = 0.75$ and $k = 1.5$. The ouflow boundary is situated along the $y = 0$ line of symmetry and the inflow boundary is along the iso-velocity line of $q = 0.5$. The resulting flow has a subsonic inflow and a mixed supersonic/subsonic outflow. Figure 6 shows contours of the Mach number of the flow field obtained with these parameters. As can be seen from the figure, the flow can be visualized as a transonic, accelerating flow contained between two curved streamlines.

If one defines the discrete error for the ith cell as

$$e_i = |\bar{\rho}_i - \rho_{\text{exact}}(\bar{x}_i, \bar{y}_i)|, \tag{21}$$

where $\bar{\rho}_i$ is the steady, discrete solution, then the L_p norms of this error are

$$L_p = \left(\frac{\sum e_i^p}{N}\right)^{1/p}. \tag{22}$$

First, an assessment of the order and magnitude of the discrete error using the Cartesian-mesh approach is made. The order is inferred by evaluating the behavior of the error norms with increasing mesh refinement, while the magnitude of the discrete error is assessed by comparing the error norms directly with those obtained from a structured mesh solver. This accuracy assessment provides a framework to quantitatively analyze other relevant procedures. The accuracy of the Green–Gauss and minimum-energy are compared to each other, and the process of cut cell data merging is evaluated.

The structured-grid flow solver uses Fromm's differencing of the primitive variables on a coordinate-by-coordinate basis. Roe's linearized Riemann solver is used to compute the fluxes through the cell interfaces. Care is taken in the formulation of the boundary procedures so that a direct comparison of the two codes yields meaningful results. Slip boundary conditions are applied by extrapolating the pressure to the Gauss points in a manner consistent with the interior scheme. At the subsonic inflow, a boundary procedure based upon constant total conditions and an extrapolated Riemann invariant (as in [24]) is used. Roe's approximate Riemann solver is used at the mixed supersonic/subsonic outflow boundary. The left and right states are supplied to the flux function from extrapolated and exact conditions evaluated at the Gauss points.

III.a. Structured Grid Results

The meshes used for the structured grid calculations have a family of coordinate lines lying along the exact solution streamlines. The other coordinate line family was generated using a sinusoidal blending of the streamline and iso-velocity constants. A sample structured mesh is shown in Fig. 7 with 400 cells. A sequence of successively finer meshes of 10×10, 20×20, 40×40, and 80×80 cells were used to compute Ringleb's flow, upon which the solution error norms were computed. The norms are tabulated in Table I.

By plotting the logarithm of the norms against the logarithm of the characteristic cell size, $1/\sqrt{N}$, one can infer the order of the truncation error from the slope of the plot. A least-squares curve fit of the data gives slopes of the L_1, L_2, and L_∞ norms of 2.08, 2.09, and 1.97, respectively, indicating that the structured scheme is uniformly second-order accurate.

III.b. Uniformly Refined Cartesian Results

Next, the Cartesian-mesh approach is used to compute Ringleb's flow on a sequence of successively finer uniform

TABLE I

Structured Grid Error Norms

N	L_1	L_2	L_∞
100	2.368e-03	2.796e-03	1.066e-02
400	4.517e-04	5.352e-04	2.700e-03
1600	1.157e-04	1.337e-04	6.918e-04
6400	3.050e-05	3.514e-05	1.745e-04

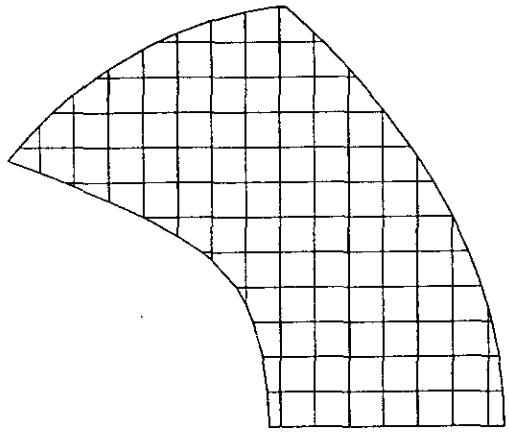

FIG. 8. Uniform Cartsian mesh, $L_0 = 4$.

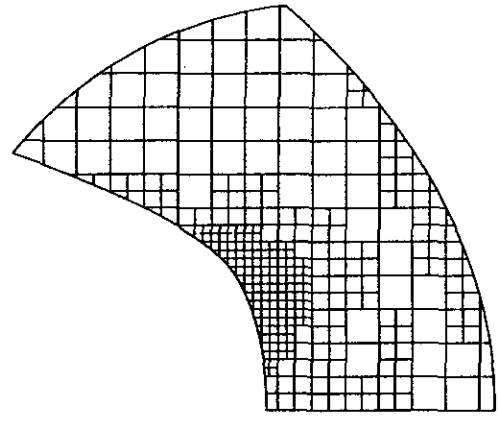

FIG. 9. Adapted Cartesian mesh.

Cartesian meshes. The uniform meshes are generated by recursively refining a set number of levels below the root of the tree, and then cutting the geometry out of the mesh. The number of levels below the root cell characterizes the fineness of the uniform meshes, which is referred to as the mesh base level, L_0.

Figure 8 shows the coarsest mesh, which is four levels below the root cell, hence, at a mesh base level of $L_0 = 4$. Uniformly refined calculations were made for base grid levels of 4, 5, 6, and 7, of which the error norms are tabulated in Table II.

A least-squares curve fit of the uniformly refined norm data yields slopes for the L_1, L_2, and L_∞ norms of 2.02, 1.91, and 1.40, respectively. Using the two finest meshes one obtains slopes for the L_1, L_2, and L_∞ norms of 2.16, 1.94, and 1.40, respectively. These slopes indicate that the Cartesian-cell based scheme is globally second-order accurate and that the local error is between first and second order. An analysis of the effect of the boundary cells upon the solution accuracy is estimated by computing error norms separately for the boundary cells and then computing the slopes as above. The computed slopes of the boundary cell L_1, L_2, and L_∞ norms were 1.68, 1.49, and 1.40. Although the local error is degraded by the irregularity in the mesh due to the cut cells/boundaries, the scheme remains globally second-order accurate.

III.c. Adaptively Refined Cartesian Results

The effect of adaptive refinement upon the solution quality is assessed next. Adaptive mesh refinement introduces refinement boundaries between cells where the cell area ratios can change by a factor of two or more. The study here investigates whether the accuracy of the scheme is maintained throughout the mesh refinement sequence and compares the accuracy with the uniformly refined scheme and structured grid schemes. Beginning at a base uniform mesh of level $L_0 = 4$, adaptation proceeds through four levels of refinement. The refinement is made according to the rotationality and compressibility parameters described above, although for this irrotational flow, the rotationality parameter is nearly zero and does not effect the refinement topology. Figure 9 shows the adapted mesh that corresponds to a mesh refinement of two levels below the base, uniform mesh.

The adaptively refined norms are tabulated in Table III. The error norms are compared with the characteristic cell size in Figs. 10, 11, and 12 for the structured, uniformly, and adaptively refined Cartesian calculations. As is shown in Figs. 10 through 12, the L_1 and L_2 norms continue to behave in a second-order accurate fashion throughout the refinement, and the L_∞ norm is appreciably reduced in the beginning stages of the refinement process.

These figures indicate that the adaptive refinement would require approximately twice the number of cells required by the structured solver for a given error magnitude. What is also indicated is that the adaptive mesh refinement requires approximately $\frac{2}{3}$ the number of cells for a given error norm that the uniformly refined (un-adapted) procedure would require.

To determine if the refinement strategy could be improved, the parameters that determine cell refinement and coarsening in Eqs. (11) and (12) were adjusted. First, to see if the length scale weighting of the refinement criterion are not tuned properly, the length scale weight powers were changed to 1 and to 2, but the effect was negligible. In addition, the cutoff parameters for coarsening and refining were adjusted. Cells were refined for refinement parameters greater than $\frac{1}{2}$ and $\frac{3}{2}$ times the standard deviation about zero, with no appreciable effect. Cells were coarsened for refinement parameters less than $\frac{1}{4}$ and $\frac{1}{2}$ the standard deviation about zero (the default level is $\frac{1}{10}$), also to no appreciable effect. These results indicate that the refinement procedure is tuned properly for this smooth flow.

TABLE II

Uniformly Refined Cartesian Error Norms

N	L_1	L_2	L_∞
118	5.345e-03	8.658e-03	4.497e-02
417	1.571e-03	2.554e-03	1.458e-02
1578	4.236e-04	7.394e-04	6.928e-03
6134	9.793e-05	1.983e-04	2.674e-03

TABLE III

Adaptively Refined Cartesian Error Norms

N	L_1	L_2	L_∞
118	5.345e-03	8.658e-03	4.497e-02
165	2.691e-03	3.995e-03	2.146e-02
337	1.310e-03	1.693e-03	6.658e-03
754	5.570e-04	7.093e-04	2.846e-03
1846	2.263e-04	3.056e-04	1.599e-03

The fact that the Cartesian error is appreciably higher than the structured solver is attributable to the alignment of one family of structured grid lines with the exact solution fluid streamlines and to the clustering of the structured grid near the minimum radius of curvature on the leftmost streamline. In [28–31] it is shown that the quasi-one-dimensional decomposition of the numerical flux in a direction normal to the cell interfaces degrades the solution when the cell interfaces are not oriented along the primary flow direction. This is due to the one-dimensional nature of the flux decomposition for a multi-dimensional problem and has prompted research into the development of truly multi-dimensional flux functions. Since the Cartesian cell interfaces (disregarding the boundary faces) are always aligned with the base coordinate axes, there is nearly always misalignment of the faces with the primary flow direction, hence the increased error. In addition, Ringleb's flow is a very smooth flow and has essentially a single-length scale; it is surmised that, on even a very coarse grid, enough of the flow field is adequately resolved, so that refinement beyond this saturation will not yield much improvement over uniform refinement. Although this study reinforces the notion that adaptive mesh refinement works best for non-smooth flows, the adaptively refined results show that the Cartesian approach studied here maintains global second-order accuracy throughout the refinement process. This finding is important, since the introduction of many refinement boundaries into the flow (across which grid cell sizes change by a factor of two-to-one and greater) must be handled properly to ensure second-order accuracy.

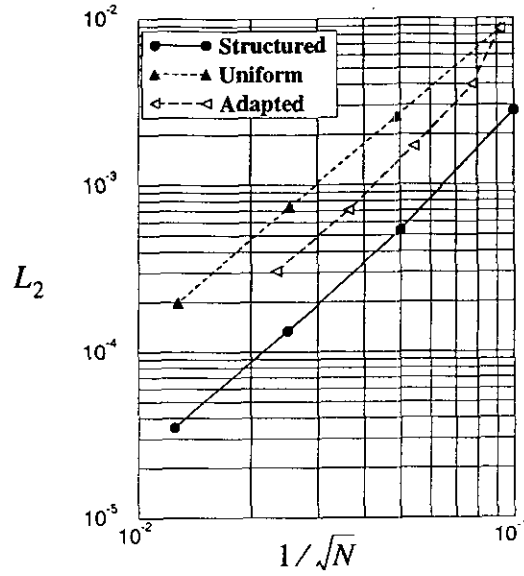

FIG. 11. L_2 norm of the error.

III.d. Green–Gauss/Minimum-Energy Reconstructions

The Green–Gauss and minimum-energy reconstructions are evaluated for uniformly refined Cartesian-mesh solutions to

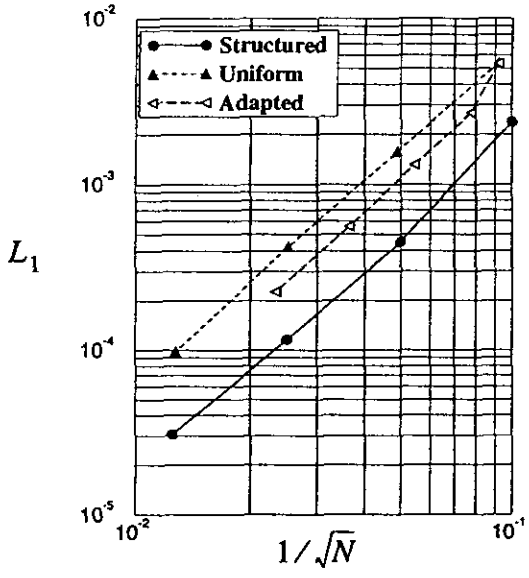

FIG. 10. L_1 norms of the error.

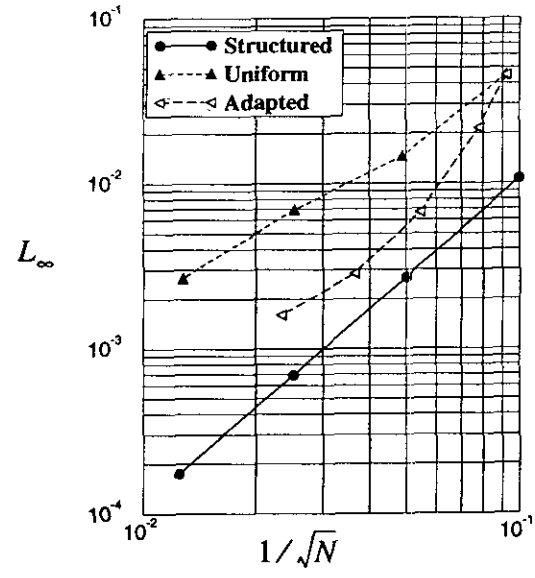

FIG. 12. L_∞ norm of the error.

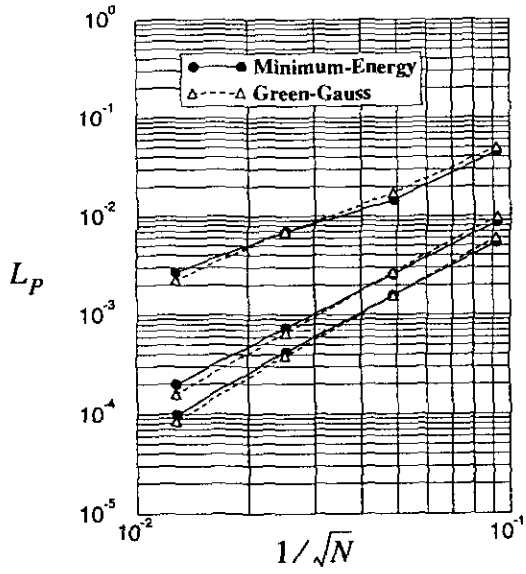

FIG. 13. Comparison of Green–Gauss and minimum-energy reconstructions.

Ringleb's flow. A truncation error analysis on uniform Cartesian grids indicates that the Green–Gauss has a slightly lower reconstruction error than the minimum-energy method and is still second-order accurate. Figure 13 shows that implementation of the two schemes results in a negligible difference.

III.e. Cut Cell Data Merging

Cell merging is a process where small, cut cells on the mesh are merged into larger, neighboring cells. The Cartesian-mesh approach routinely generates meshes which have a many-order variation of cell areas across the field. This wide area variation comes about through mesh refinement and from cell cutting. In the grid interior, smoothness can be imposed by only allowing a set number of tree level differences across the faces. Smoothness cannot be guaranteed at the boundaries.

Typically, there are only a small number of very small cut cells introduced along the boundaries, where their areas may vary greatly compared to their neighbors. For steady state calculations, the stiffness introduced from these cells is reduced by using a spatially varying time step, but for time-accurate computations, as in [26, 27], the efficiency can be greatly reduced. Cut cell merging is proposed to eliminate the stiffness caused by these few, very small cut cells that can be introduced into the mesh.

Cell merging creates a new cell, which is referred to as a merged cell, from a larger (mother) cell and a small cut (fragment) cell. This combination of the two, mother and fragment, cells into the single, larger merged cell can be achieved either geometrically or by a subtle change in the procedures used to compute the fluxes and flux balances on the fragment and mother cells. Since geometric merging can (except in fortunate circumstances) violate the hierarchy of the grid, cells are merged using the latter method. Theoretically, one can show that the global order of accuracy is unaffected by cell merging, although the error magnitude is increased. On a uniform mesh with an equal error distribution, one can relate the unmerged to merged L_1 norm of the error as

$$L_{1,\text{Merged}} = L_1\left(1 + \frac{3M}{N-M}\right), \tag{23}$$

where M cells have been merged on a mesh consisting of N cells prior to the cell merge. For small M/N, the magnitude of error is increased only negligibly, while the order is unaffected. The process of cell data merging proceeds as follows.

Identification

Fragment cells are chosen to be the minimum area cut cells in the mesh. Mother cells are chosen to be the maximum area cell that is a face neighbor to the fragment cell. For cells identified as merged cells, the merging changes its shape, and hence its centroid. So, the new merged cell centroid is computed from the fragment and mother cell centroids and areas,

$$\bar{x}_m = \frac{\bar{x}_M A_M + \bar{x}_F A_F}{A_M + A_F}, \tag{24}$$

where the subscripts m, M, and F refer to merged, mother, and fragment cells, respectively.

Reconstruction

After all the cell data has been reconstructed, the fragment cells data are altered so that at the interfaces between mother and fragment cells the reconstruction yields a unique value. That is, replace the fragment cell data as

$$\bar{u}_F = \bar{u}_M + \nabla u_M(\bar{x}_F - \bar{x}_M) \tag{25}$$

$$\nabla u_F = \nabla u_M. \tag{26}$$

Flux Construction/Evolution

The merged cell data is used as a matter of course in the flux computation for the mother and fragment cells. The flux balance on the larger, merged cell is obtained by summing the contributions of the mother and fragment cells to the merged cell

$$R_m = R_M + \sum R_F, \tag{27}$$

where R represents the flux balance of Eq. (10).

Here, the spatial accuracy of cell merging in the framework of Ringleb's flow is investigated. Adaptively refined calculations are made starting at a base grid level of $L_0 = 4$ and

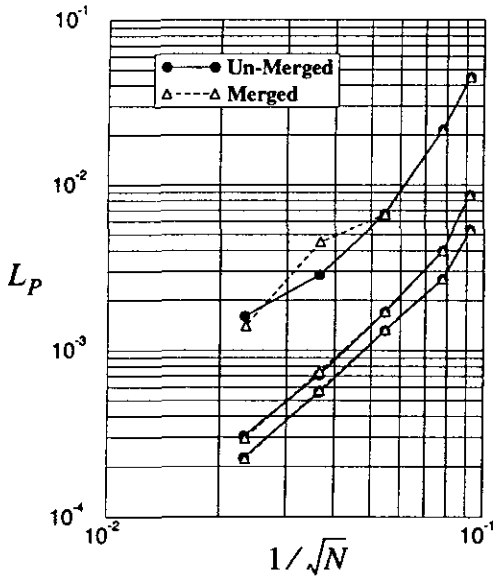

FIG. 14. Error norms: Merged and unmerged.

refining four levels. At each refinement level, the smallest cut cell is merged into the largest of its neighbors. Figure 14 shows all error norms compared with the unmerged results presented in Section III. In an attempt to gauge the increase in efficiency that could be achieved in a time accurate computation, the minimum time step is found for each grid level for the merged and unmerged results. The ratio of the minimum merged to minimum unmerged time step is shown in Fig. 15, where

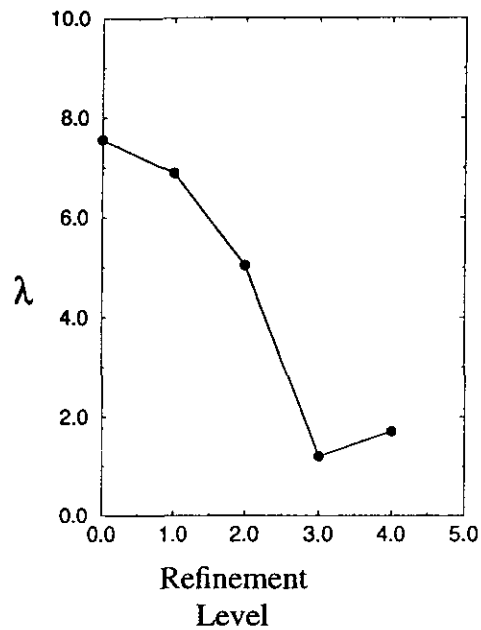

FIG. 15. Ratio of minimum time steps: Merged to unmerged.

$$\lambda = \frac{\Delta t_{\text{min,merged}}}{\Delta t_{\text{min,unmerged}}}. \tag{28}$$

One can see that the global error is practically unaffected by the cell merging, while there is a slight increase in the local error at the third refinement level. One can also see that the order of accuracy remains unchanged. The gain in efficiency through the increase of the time steps shown in Fig. 15 drops off with the refinement level. As the grid becomes populated with smaller and smaller cells from the adaptive mesh refinement, the smallest uncut cell sizes approach the sizes of the small cut cells, causing the ratio of minimum step sizes to decrease.

IV. CONCLUSIONS

In this paper a critical assessment of the accuracy of Cartesian-mesh approaches for computing steady, transonic solutions to the Euler equations of gas dynamics has been made. The basics behind the hierarchical data structure that can be used to store the mesh data and how arbitrary geometries can be represented with the Cartesian-mesh approach has been explained. The Cartesian-mesh approach presented used a cell centered, upwind-based finite-volume formulation coupled with a particular K-exact reconstruction of the primitive variables. Adaptive mesh refinement was achieved using cell size scaled refinement parameters based upon the local compressibility and rotationality of the fluid.

An exact, transonic, steady solution to the Euler equations (Ringleb's flow) was used to assess the order and magnitude of the solution error. The solution errors for both structured and Cartesian-mesh based calculations of Ringleb's flow were compared directly. Uniform refinement was performed for the structured-mesh calculations while both uniform and solution adaptive refinements were made for the Cartesian-mesh approach. Care was taken in the formulation of the structured solver, so that similar numerics are used by both approaches. The order of the discrete solution error was inferred by the norms of the errors through mesh refinement, and the magnitude of the error of the Cartesian approach were assessed by direct comparison to the errors of the structured grid approach. The assessment indicated that the Cartesian-mesh approach was globally second-order accurate and remained so through adaptive refinement, while the local error was between first- and second-order accuracy. In addition, adjustment of critical parameters which govern the mesh refinement indicators had little improvement of the adaptive mesh efficiency, which is surmised to be attributed to the smoothness of the particular flow problem being investigated. The magnitude of the Cartesian mesh discrete error is shown to be higher than that of the structured solver. This can be attributed to the fortunate alignment of one family of grid lines of the structured grid with the solution streamlines.

The accuracy of two different reconstruction procedures,

minimum-energy and Green–Gauss, were compared directly and shown to give no appreciable difference. Cell data merging was investigated and was shown to increase the efficiency for time accurate computations by increasing the allowable time step with a negligible cost in accuracy.

ACKNOWLEDGMENTS

The authors express their gratitude to Professor M. J. Berger for her insightful comments regarding this work. Thanks also go to Mr. B. P. Curlett for use of his workstation "hornet" for some of the Ringleb flow calculations and to Dr. D. Rigby, Dr. P. Jorgenson, Dr. J. Lee, and Mr. C. Steffen, Jr. for many interesting discussions.

REFERENCES

1. J. F. Dannenhoffer III, *Int. J. Numer. Methods Eng.* **32,** 653 (1991).
2. M. J. Berger, *J. Comput. Phys.* **53,** 484 (1984).
3. M. J. Berger and A. Jameson, *AIAA J.* **23**(4), 561 (1985).
4. M. J. Berger and P. Colella, *J. Comput. Phys.* **82,** 64, (1989).
5. J. J. Quirk, Ph.D. thesis, Cranfield Institute of Technology, College of Aeronautics, 1991 (unpublished).
6. D. P. Young, R. G. Melvin, M. B. Bieterman, F. T. Johnson, S. S. Samant, and J. E. Bussoletti, *J. Comput. Phys.* **92,** 1 (1991).
7. D. K. Clarke, M. D. Salas, and H. A. Hassan, *AIAA J.* **24**(3), 353 (1986).
8. J. J. Quirk, ICASE Report No. 92-7, 1992 (unpublished).
9. M. J. Berger and R. J. LeVeque, AIAA Paper 89-1930-CP (unpublished).
10. M. J. Berger and R. J. LeVeque, *Comput. Systems Eng.* **1**(2–4), 305 (1990).
11. M. J. Berger and R. J. LeVeque, AIAA Paper CP-91-1602 (unpublished).
12. K. Morinishi, *Comput. Fluids* **29**(3), 331 (1992).
13. B. Epstein, A. L. Luntz, and A. Nachson, AIAA Paper 89-1960-CP (unpublished).
14. D. De Zeeuw and K. G. Powell, *J. Comput. Phys.* **104,** 56 (1991).
15. D. De Zeeuw and K. G. Powell, AIAA Paper 93-0321 (unpublished).
16. T. J. Barth, Computational Fluid Dynamics Lecture Series, 1990-03, Von Karman Institute for Fluid Dynamics (unpublished).
17. T. J. Barth and P. O. Frederickson, AIAA Paper 90-0013 (unpublished).
18. K. G. Powell, P. L. Roe, and J. J. Quirk, *Algorithmic Trends in Computational Fluid Dynamics in the 1990's, Sept. 1991;* to appear.
19. A. G. Godfrey, C. R. Mitchell, and R. W. Walters, AIAA Paper 92-0054 (unpublished).
20. P. L. Roe, *J. Comput. Phys.* **43,** 357 (1981).
21. P. H. Paillere, K. G. Powell, and D. De Zeeuw, AIAA Paper 92-0322 (unpublished).
22. AGARD Report No. AR-211 (unpublished).
23. D. W. Halt and R. K. Agarwal, *AIAA J.* **30**(8), 1993 (1992).
24. R. V. Chima, NASA TM 103800 (unpublished).
25. W. R. Hingst and D. F. Johnson, NASA TM X-2902 (unpublished).
26. S. A. Bayyuk, K. G. Powell, and B. van Leer, "An Algorithm For the Simulation of 2-D Unsteady Inviscid Flows Around Arbitrarily Moving and Deforming Bodies of Arbitrary Geometry," AIAA 11th Computational Fluid Dynamics Proceedings (unpublished).
27. Y.-L. Chiang, Ph.D. thesis, Aerospace Engineering Department, University of Michigan, 1992 (unpublished).
28. R. L. Rumsey, B. van Leer, and P. L. Roe, AIAA-91-1530-CP (unpublished).
29. D. Levy, K. G. Powell, and B. van Leer, AIAA-89-1931-CP (unpublished).
30. H. Deconinck, K. G. Powell, P. L. Roe, and R. Struijs, AIAA-91-1532-CP (unpublished).
31. P. L. Roe, *J. Comput. Phys.* **63,** 458 (1986).