AIAA Journal
Vol. 30, No. 8, August 1992

Compact Higher Order Characteristic-Based Euler Solver for Unstructured Grids

D. W. Halt* and R. K. Agarwal†
McDonnell Douglas Research Laboratories, St. Louis, Missouri 63166

Two new methods are presented for solving the Euler equations using a compact higher order polynomial reconstruction technique on unstructured grids. The methods use a characteristic-based approach with a cell-centered finite volume method. For transonic Ringleb flow, computations are performed for first-order to fourth-order accuracy and are compared with the hodograph solution. Results for a 10-deg ramp case are also presented. An analysis is performed that demonstrates that the higher order method is an order of magnitude more efficient than the lower order method in modeling the flow for moderate-to-fine error tolerances. Accuracy, speed, and memory requirements are evaluated in the efficiency study.

Introduction

A COMPACT higher order polynomial reconstruction technique is developed that allows for higher order characteristic-based numerical solutions to the Euler equations on unstructured grids. This reconstruction requires an extended set of Euler equations that includes as dependent variables not only mass, momentum, and energy but also their spatial derivatives. These additional dependent variables are used in a compact polynomial reconstruction process within a finite-volume framework. The development presented follows some of the ideas in the computational fluid dynamics (CFD) literature using continuous and discontinuous piecewise polynomial approximations.

Continuous piecewise linear polynomials have been used in several finite volume and finite element unstructured grid methods. Jameson and Mavriplis,[1] Lohner et al.,[2] and others have developed continuous piecewise linear elements using an added artificial dissipation and/or flux-corrected transport terms. More recently, Oden[3] has extended the linear elements by the use of higher order p elements. Oden's work represents a compact higher order approximation of the continuous element type.

Discontinuous piecewise linear polynomials have similarly been used on unstructured grids with a characteristic-based approach. Thareja et al.,[4] Batina,[5] Frink,[6] and others have developed discontinuous piecewise linear schemes with approximate Riemann solvers on unstructured grids. Barth and Frederickson[7] developed a technique where polynomials of degree k are reconstructed for each cell by maintaining average dependent variable properties over a neighboring set of cells in a least squares formulation. These polynomials are reconstructed using a large number of neighboring cells; therefore, the representation is not compact. This paper describes two methods that compactly reconstruct higher order polynomials over each cell in a discontinuous piecewise polynomial procedure.

Two methods are presented for extending the Euler equation set. The first method solves the spatial derivatives of the governing integral equations. The second method follows the work of Allmaras,[8] where the spatial moments of the governing equations are solved. Allmaras developed an approach with linear reconstructions on structured grids by using the first moment equations. This paper extends his approach to higher order and unstructured grids.

A compact reconstruction procedure is desirable for several reasons. First, the error for a compact polynomial reconstruction should be lower than that composed of many far-reaching neighbors. Second, the compact method should be much less sensitive to grid issues than a noncompact method. Third, only cells that contain flow singularities (e.g., shocks) are complicated by the compact reconstruction process, whereas for noncompact schemes, the discontinuities affect a much larger set of neighboring cells. Fourth, the compact method treats cells near boundaries like any other cells, but a noncompact scheme must disproportionately have more neighbors on one side than another and/or incorporate boundary values at ghost cells.

Numerical results are presented for the transonic shockless Ringleb flow.[9] A matrix of results is compared with the exact hodograph solution to establish accuracy levels for a two-dimensional transonic shockless flow. A study is also performed to ascertain the relative efficiencies of various orders of compact reconstruction. Future modifications to the compact scheme are anticipated for nonoscillatory shock capturing as discussed by Barth and Jesperson.[10] The second method has been demonstrated by Allmaras[8] to capture shocks using linear reconstructions when the dependent variables are cast in Roe's parametric vector form.

Governing Equations

The Euler equations for an unstructured two-dimensional grid are written in integral equation form as follows:

$$\frac{\partial}{\partial t}\int\int_V Q dV + \int_S F \cdot \hat{n} dS = 0 \tag{1}$$

where

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e_0 \end{bmatrix}, \qquad F = G\hat{\imath} + H\hat{\jmath} \tag{2}$$

$$G = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uh_0 \end{bmatrix}, \qquad H = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vh_0 \end{bmatrix} \tag{3}$$

$$p = \rho(\gamma - 1)\left[e_0 - \frac{(u^2 + v^2)}{2}\right] \tag{4}$$

Received Aug. 5, 1991; presented as Paper 91-3234 at the AIAA 9th Applied Aerodynamics Conference, Baltimore, MD, Sept. 23-26, 1991; revision received Dec. 6, 1991; accepted for publication Dec. 28, 1991. Copyright © 1992 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

*Scientist. Senior Member AIAA.

†McDonnell Douglas Corp. Fellow, Program Director, Computational Fluid Dynamics. Associate Fellow AIAA.

where ρ is the density, u and v are the velocity components, e_0 is the total energy per unit mass, h_0 is the total enthalpy per unit mass, and p is the static pressure. The dot product, $F \cdot \hat{n}$, represents the flux out of the control volume where $\hat{n}$ is the outward pointing normal to surface S. Pressure is determined from the equation of state, Eq. (4), for a perfect gas. The density and velocity are nondimensionalized by freestream density and freestream speed of sound, respectively. The energy and enthalpy are nondimensionalized by the freestream speed of sound squared, and the pressure is nondimensionalized by the freestream density times speed of sound squared.

Method 1—Derivative Equations

The first method considers the spatial derivatives of the governing equation to be an extended set of governing equations:

$$\frac{\partial^m}{\partial x^m}\frac{\partial^n}{\partial y^n}\left[\frac{\partial}{\partial t}\int\int_V QdV + \int_S F \cdot \hat{n}dS\right] = 0 \tag{5}$$

where $0 \le m \le k$, $0 \le n \le k$, and $0 \le m + n \le k$. They are solved simultaneously to level k to achieve the compact form of the reconstruction. The number of simultaneous equations necessary for a level k compact reconstruction in two dimensions is

$$neq_k = \frac{(k+1)(k+2)}{2} \tag{6}$$

Compact Reconstruction

A compact reconstruction polynomial is applied at each cell locally. The dependent variables of each cell represent the average values that the reconstructed polynomial must have over each cell. For example, given the average value of the dependent variables Q_{mn} of cell i, a polynomial of degree $k = 1$ is reconstructed as follows:

$$Q(x, y)_i = Q_{00} + (x - x_c)Q_{10} + (y - y_c)Q_{01} \tag{7}$$

where $Q_{mn} = (1/V)\int\int_V(\partial^m/\partial x^m)(\partial^n/\partial y^n)QdV$, and (x_c, y_c) is the centroid of cell i. Polynomials of higher order than $k = 1$ require the use of calculated moments of inertia for each cell as in Ref. 7. The reconstruction polynomial is used to determine the left and right states at all Gaussian quadrature points on the cell perimeter. Flux values are evaluated at each Gauss point by a characteristic analysis of the left and right states. Numerical integration of the fluxes through each edge is accomplished by Gaussian quadrature. One Gauss point on the center of each edge is used with $k = 0$ and $k = 1$ reconstructions. This is sufficient for integrations that provide first- and second-order spatial accuracy, respectively. Two Gauss points on each edge are used with $k = 2$ and $k = 3$ reconstructions and are sufficient for third- and fourth-order spatial accuracy, respectively.

Characteristic-Based Flux Model

An appropriate flux vector is determined by approximating a pseudo one-dimensional Riemann solution between the left and right states at any Gauss point as follows:

$$F_{mn} = ½[F_{mnR} + F_{mnL} - S|\Delta|\Delta^{-1}S^{-1}(F_{mnR} - F_{mnL})] \tag{8}$$

where S is the similarity matrix for diagonalizing the Jacobian $\partial F_{mn}/\partial Q_{mn}$, Δ is the eigenvalue diagonal matrix, and $|\Delta|$ refers to the absolute value of Δ obtained by taking the absolute values of each eigenvalue in the matrix. Note that the Jacobian is identical for all m and n, i.e.,

$$\partial F_{00}/\partial Q_{00} = \partial F_{mn}/\partial Q_{mn} \tag{9}$$

The flux formula resembles Roe's scheme[11] except that the flux jump ΔF drives the upwinding term (and not the primitive variable jump Δq). If Roe's scheme were attempted on the extended set of governing equations, one property does not extend to the higher derivatives, i.e., $\Delta F = (\partial F/\partial q)\Delta q$ is not extendable to $\Delta F_{mn} = (\partial F/\partial q)\Delta q_{mn}$. This implies that, if the exact values on either side of a shock were imposed as left and right states, an incorrect flux value would result. The previous flux formula does not suffer from this problem and will return the exact flux value when the left and right states represent the exact values on either side of a shock.

The last term in the flux formula is often referred to as the dissipation term in the numerical method. It is interesting to note that this term exists for each equation in the extended set; however, the order of accuracy is not diminished by this. It will be shown for the Ringleb flow problem that the order of accuracy of $k + 1$ results from reconstruction polynomials of order k.

Numerical Method

The computational domain is discretized by a novel cell-centered finite volume method using an approximate Riemann solver. The dependent variables Q_{mn} are updated by the Jacobi algorithm for each cell i as follows:

$$\frac{Q^{n+1}_{mni} - Q^n_{mni}}{\Delta t}V_i + \sum_{j=1}^{nfaces}(F_{mnj} \cdot \hat{n}_j)A_j = 0 \tag{10}$$

where A_j is the area of face j, V_i is the volume of cell i and Δt is the time step.

The indexing or pointer system is based on two primary pointer arrays for each face. The first array points to the two cells adjacent to each face. The second array identifies the vertices of each face.

The metrics can be efficiently computed from the pointer array defining the vertices of each cell face. The flow solver initially computes the left and right states for each face. The residuals are accumulated for each cell by looping through the faces and adding/subtracting flux contributions of the face from the associated left and right cells. The scheme then updates the dependent variables for every cell, and the iteration process is repeated for a desired number of iterations.

The grid points can be numbered randomly. The cell faces are numbered to alleviate recurrences when accumulating residuals for efficient vectorization. Boundary conditions are treated by separate pointer arrays which do not contribute significantly to the memory requirements.

Method 2—Moment Equations

The second method requires the solution of the moment equations,

$$\int\int_V x^m y^n\left[\frac{\partial}{\partial t}Q + \nabla \cdot F\right]dV = 0 \tag{11}$$

where $0 \le m \le k$, $0 \le n \le k$, $0 \le m + n \le k$, and the (x, y) origin is shifted to the centroid of each cell locally. Equation (6) also gives the number of simultaneous equations for the second method. The equation for $m = n = 0$ is the same as Eq. (1) for both methods. All higher order equations are different for the two methods.

To cast Eq. (11) in a form where spatial derivatives can be updated in time, Eq. (7) is substituted into Q. Using the fact that first moment of inertias about a centroid are zero, the first moment equations reduce to

$$A_{xx}Q_{xt} + A_{xy}Q_{yt} + \int_S xF \cdot \hat{n}dS - \int\int_V GdV = 0 \tag{12}$$

$$A_{xy}Q_{xt} + A_{yy}Q_{yt} + \int_S yF \cdot \hat{n}dS - \int\int_V HdV = 0 \tag{13}$$

where A_{xx}, A_{xy}, and A_{yy} are the second moments of inertia. The integration of the flux G or H over the cell volume can be done numerically using the efficient quadrature formulas for the triangle by Dunavant.[12] Another approach proposed by Allmaras[8] is to implement Roe's parametric vector as the dependent variables, which allows direct integration of the fluxes over the cell volume given the moments of inertia. Results shown in this paper use the numerical quadrature formulas.

The second method requires $k+1$ Gauss points for the surface integrations. The higher number of Gauss points than the first method can be attributed to the moments where a polynomial of order k is multiplied by x^k or y^k, requiring a higher order quadrature. We will see this is not a limitation for the second method.

The characteristic-based flux model for this method is simpler than the first method and can use standard flux-split formulas such as those of Roe,[11] van Leer,[13] or Osher and Chakravarthy.[14] Each of these approximate Riemann schemes is implemented in the current flow solver.

Boundary Conditions

The boundary conditions for the Ringleb flow problem come from the hodograph solution. The Ringleb boundary values are specified as the outer state for the cell edges along the outer boundary. The characteristic-based flux formula retains the hyperbolic nature of the boundary value problem and keeps it from being overspecified.

For problems where the outer states are not known, a far-field, symmetry, and Neumann condition are imposed. In the first method, it is quite difficult to properly impose these conditions because derivatives of mass, momentum, and energy are involved in the implementation. The second method uses conventional boundary conditions and it is much simpler to implement proper boundary conditions. Flow tangency is imposed for the surface Neumann condition by subtracting the normal velocity component from the velocity vector at the Gauss points. The left and right states are set equal here so that zero mass and energy fluxes are insured through the surface boundary. Characteristic boundary conditions are applied to the far-field boundary by using the appropriate values of freestream and flowfield Riemann variables.

Ringleb Results—Method 1

The Ringleb flow is chosen as a model problem because it is a transonic flow and an exact solution exists for comparison. A triangular region is selected from the Ringleb flowfield as the model region shown in Fig. 1. A curved region is not selected here because a series of straight line segments along the curved boundary would introduce a truncation error from modeling the boundary shape. This method may be extended to cells with curved edges to alleviate this modeling error. A matrix of results for various grid sizes is compared with the exact solution so that the accuracy and efficiency of this compact higher order method can be demonstrated.

A set of computational results is shown in Table 1 obtained with the first method. In Table 1, k denotes the degree of polynomial reconstruction. For each k, successive grids are generated by subdividing each cell into four smaller cells of equal size. L2 error refers to the rms error between the reconstructed and exact density at the vertices of each cell. SP.RAD refers to the average spectral radius over the first six orders of magnitude of residual reduction. CPU refers to the number of microseconds of CPU time per iteration per cell used by a single CRAY YMP processor. The time steps used were 0.1, 0.05, 0.025, and 0.015, respectively, for the 16, 64, 256, and 1024 cell cases and were near optimal for the point Jacobi scheme. One Gauss point is used on each edge when $k = 0$ or 1, whereas two Gauss points are used when $k = 2$ or 3.

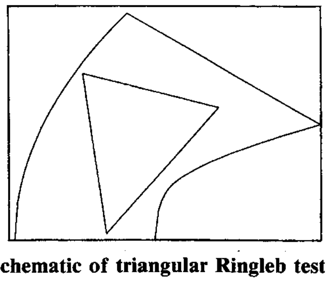

Fig. 1 Schematic of triangular Ringleb test region.

Table 1 Method 1—accuracy and efficiency results for Ringleb case

k	Cells	L2 error	SP.RAD	CPU	MEM
0	16	7.80×10^{-2}	0.917	11	1.0
0	64	4.08×10^{-2}	0.946	6	2.6
0	256	2.09×10^{-2}	0.977	4	9.0
0	1024	1.06×10^{-2}	0.988	4	34.4
1	16	2.15×10^{-2}	0.931	24	1.5
1	64	6.05×10^{-3}	0.964	13	4.6
1	256	1.65×10^{-3}	0.984	11	16.9
1	1024	4.46×10^{-4}	0.989	10	66.0
2	16	6.99×10^{-3}	0.937	84	2.3
2	64	2.17×10^{-3}	0.968	47	7.7
2	256	3.16×10^{-4}	0.979	38	28.7
2	1024	4.37×10^{-5}	0.990	35	112.7
3	16	2.06×10^{-2}	0.952	158	3.4
3	64	9.70×10^{-4}	0.974	90	11.7
3	256	1.63×10^{-4}	0.990	75	44.3
3	1024	8.71×10^{-6}	0.994	70	174.9

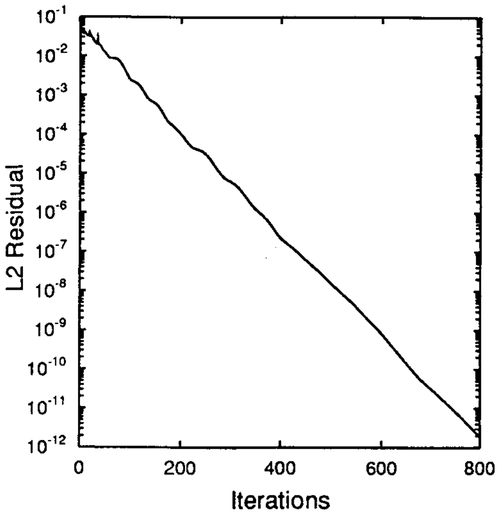

Fig. 2 Residual history for 64 cell, $k = 3$ case.

The order of accuracy is determined to be approximately $k+1$ for each k in Table 1. As expected, the spectral radius increases as the number of cells increases. There is also a slight increase in the spectral radius as k increases. The CPU time per cell per iteration varies considerably because the vector lengths are very small for the cases with fewer cells. The CPU time and memory requirement also increase as k increases because more equations are solved per cell. For all cases shown in Table 1, the residual decayed smoothly like the example shown in Fig. 2. However, when only one or four cells were used, the $k = 3$ cases did not converge.

The accuracy of the compact scheme is compared with the noncompact scheme of Barth[7] in Fig. 3. The first method is nearly a half of an order of magnitude more accurate than the noncompact method for $k = 3$ and furthermore shows a promising slope for extension to $k > 3$. The comparison with the second compact method will be discussed in the next section. Unfortunately, the domain used by Barth[7] includes a curved region between two streamlines, whereas the compact solution is limited to the triangular region shown in Fig. 1.

Mach contours are plotted in Figs. 4–6 for three cases where $k = 3$. Each cell reconstructs a local solution from a two-dimensional polynomial of order 3, and the Mach contours are

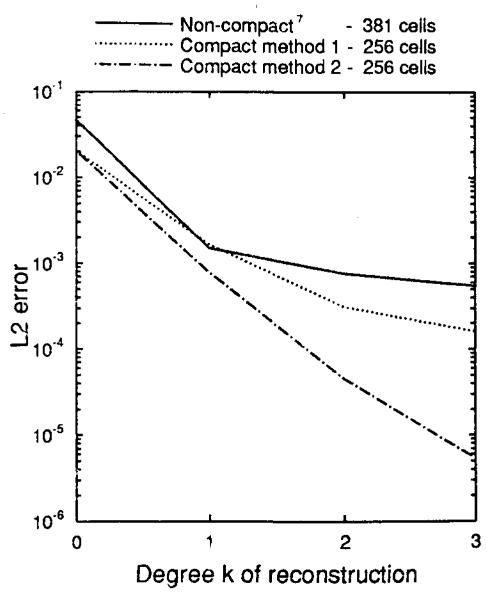

Fig. 3 Comparison of L2 errors between compact and noncompact schemes.

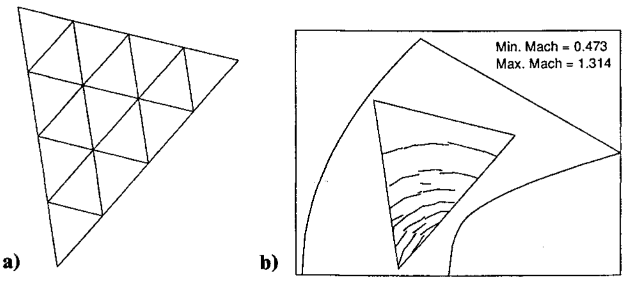

Fig. 4 Sixteen cell case: a) unstructured grid and b) cellwise continuous Mach contours for $k = 3$.

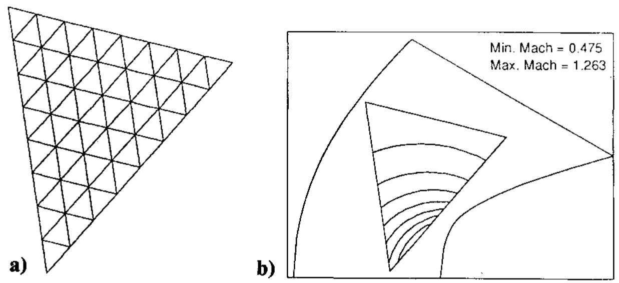

Fig. 5 Sixty-four cell case: a) unstructured grid and b) cellwise continuous Mach contours for $k = 3$.

plotted in a piecewise continuous manner. Significant contour level jumps can be seen in the 16 cell case; however, the error rapidly diminishes as the number of cells increases.

An efficiency study is made using the results in Table 1. Employing a six order of magnitude convergence criterion, the number of iterations needed to converge is determined from the spectral radius. The total CPU time is then determined and is plotted against L2 error in Fig. 7. The optimum choice of k is seen to vary depending on the desired error level. The more stringent the error requirement, the higher is the optimum k. The quadratic reconstruction ($k = 2$) is the optimum choice for a wide range of errors from 7×10^{-3} to 5×10^{-5}. Lower order reconstructions would require significantly more CPU time at modest error levels (below 1×10^{-3}) based on extrapolation of the curves.

The memory requirements of the compact scheme for the first method are compared for various error levels in Fig. 8. Here, the product of CPU time and the central memory needed is plotted against the L2 error. Once again, as the desired error tolerance becomes more stringent, the optimum use of memory requires a larger k value. The coupled effect of CPU and memory efficiency makes an even stronger case for the utilization of the higher k values for moderate to more stringent error tolerance requirements.

Perhaps of even greater significance is the drastic loss of efficiency for the lower k when a more stringent error tolerance is required. A lower k may require several orders of magnitude more CPU time, whereas a higher k may require

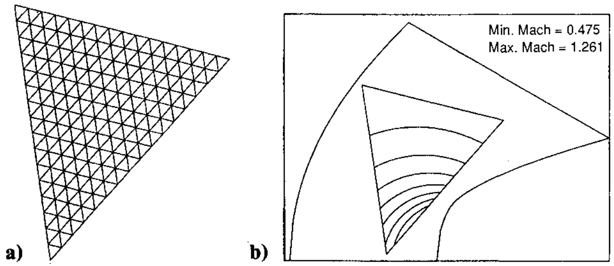

Fig. 6 Two hundred and fifty-six cell case: a) unstructured grid and b) cellwise continuous Mach contours for $k = 3$.

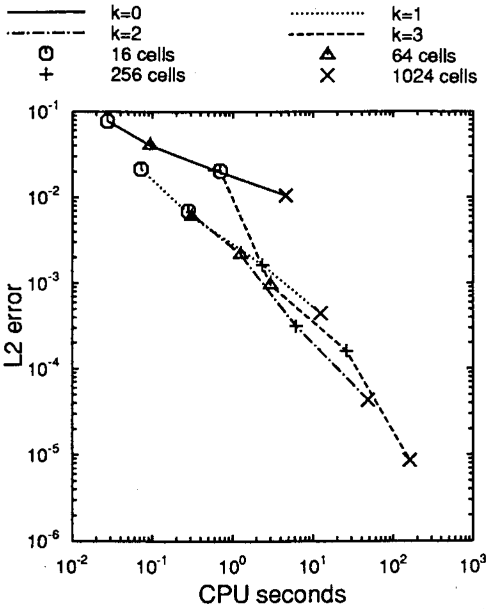

Fig. 7 CPU efficiency comparison for method 1.

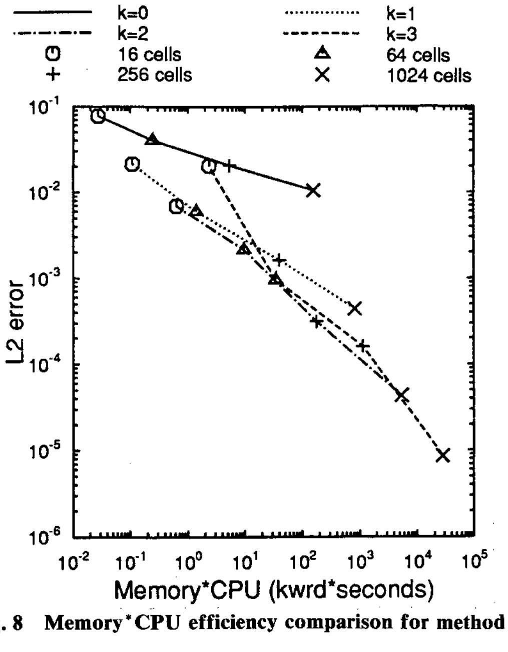

Fig. 8 Memory*CPU efficiency comparison for method 1.

only a few times more CPU time for a moderate error level. The optimum choice of k can be an important selection to conserve computer resources. A conservative approach may be to select a k value slightly higher than the estimated optimum k value for the accuracy level desired.

Another good approach might be to start calculations at lower values of k and increase k in sequential steps to improve convergence rates. The idea is much like grid sequencing except that only one grid is needed here. Furthermore, an estimate of the error level may be done by extrapolation of the various order k solutions.

Ringleb Results—Method 2

For the second method, a set of computational results is shown in Table 2. Solutions from 1 cell through 1024 cells are cross-tabulated for constant ($k = 0$) to cubic ($k = 3$) polynomial reconstructions. The errors are significantly lower than corresponding cases in Table 1 when $k > 0$. However, the spectral radii and CPU times per iteration per cell are higher. The CPU times are higher primarily because of the increased number of Gauss points needed for numerical integration. The CPU times are lower than those in Table 1 for the $k = 0$ cases because of more efficient programming techniques used for the second method. The memory needed is roughly the same for both methods.

The results for the cases with 256 cells are compared with the first method and the noncompact method of Barth[7] in Fig. 3. The L2 error is substantially smaller for the second compact method. For $k = 3$ solutions, the second method is two orders of magnitude more accurate than the noncompact result and an order and a half more accurate than the first compact method. Similar timings between the first compact method and the noncompact method are expected due to the similarity in degree k polynomial reconstructions and orders of Gaussian quadrature. However, the timings of the noncompact calculations are not known.

The spatial orders of accuracy are slightly degraded for the second method. Orders of accuracy for the results in Table 2 are approximately 0.98, 1.94, 2.84, and 3.60 for k values of 0, 1, 2, and 3, respectively. The area integrals used quadrature integrations corresponding to $k = 1$, 2, 4, and 6, respectively. The contour integrals used $k + 1$ Gauss points for these cases. It may be possible that higher order quadrature formulas are necessary for more accurate area integrations since the flux functions are higher order than the dependent variable functions.

Table 2 Method 2—accuracy and efficiency results for Ringleb case

k	Cells	L2 error	SP.RAD	CPU	MEM
0	1	1.97×10^{-1}	0.740	85	0.05
0	4	1.05×10^{-1}	0.855	22	0.16
0	16	7.80×10^{-2}	0.906	7	0.56
0	64	4.08×10^{-2}	0.946	3	2.1
0	256	2.09×10^{-2}	0.977	3	8.3
0	1024	1.06×10^{-2}	0.988	2	32.7
1	1	7.45×10^{-2}	0.923	248	0.08
1	4	4.86×10^{-2}	0.964	82	0.27
1	16	1.07×10^{-2}	0.965	33	1.0
1	64	2.99×10^{-3}	0.980	22	4.0
1	256	7.82×10^{-3}	0.986	20	15.7
1	1024	2.04×10^{-4}	0.992	20	62.4
2	1	4.39×10^{-4}	0.951	670	0.13
2	4	9.54×10^{-2}	0.958	202	0.5
2	16	1.91×10^{-3}	0.982	87	1.9
2	64	3.29×10^{-4}	0.989	63	7.6
2	256	4.62×10^{-5}	0.992	59	30.3
2	1024	6.46×10^{-6}	0.996	57	120.7
3	1	3.07×10^{-2}	0.971	1502	0.24
3	4	3.85×10^{-3}	0.983	444	0.9
3	16	5.25×10^{-4}	0.997	184	3.6
3	64	6.52×10^{-5}	0.996	130	14.4
3	256	5.49×10^{-6}	0.998	121	57.2

One approach to resolve this is to change to Roe's parametric vector as the dependent variable as has been done by Allmaras.[8] This causes the flux functions to be of the same polynomial order as the conservation variables. Allmaras states that this helped capture shock waves more robustly.

The under-relaxation technique used by Allmaras[8] was used to advance the equations in time. Here, all equations for $k > 0$ are relaxed by a factor of 0.10 at each iteration in an explicit point Jacobi process. This allows quicker convergence rates compared with when all equations are advanced with the same time steps.

An efficiency comparison is also made for the second method. Total CPU times were calculated from the data in Table 2, again assuming six orders of magnitude residual reduction for the convergence criterion. Figure 9 shows total CPU time on a CRAY YMP single processor against L2 error. These results show that $k = 2$ reconstruction is most efficient for a wide range of errors from 7×10^{-3} to 1×10^{-5}. Again, as the error tolerance is lowered, the larger k values conserve CPU time.

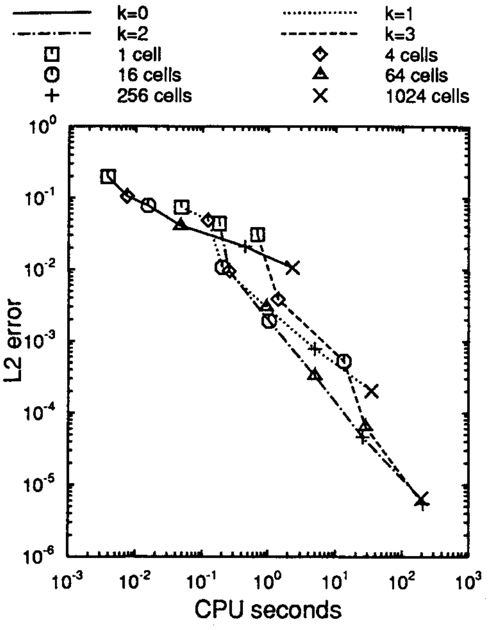

Fig. 9 CPU efficiency comparison for method 2.

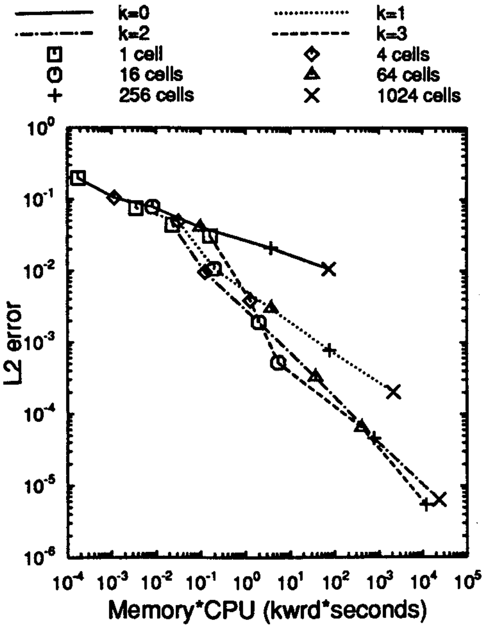

Fig. 10 Memory*CPU efficiency comparison for method 2.

Figure 10 shows the effect when central memory is coupled with CPU time. The higher order reconstructions become favorable at higher error levels relative to when only CPU time is considered as shown in Fig. 10. The $k = 2$ results are most efficient down to error levels of 2×10^{-3}, whereas $k = 3$ becomes marginally more efficient for smaller errors.

10-Deg Ramp Case

A 10-deg ramp case was analyzed at a freestream Mach number of 0.5 using the second method. A 10-deg ramp, one unit wide, is connected between two horizontal lines each 50 units wide. A far-field boundary was placed 50 units high and 101 units wide. An unstructured grid of 480 triangular cells was generated with Lohner's[2] two-dimensional advancing front grid generator. The grid is shown near the compression and expansion corners in Fig. 11. The far-field boundary conditions are specified as the freestream conditions since the flow is entirely subcritical.

Piecewise continuous Mach contours for a quadratic reconstruction solution ($k = 2$) are shown in Fig. 12. Relatively smooth and continuous contours near the compression and expansion corners indicate a well-behaved solution. Since there is no known analytical solution for this problem, one measure of the error in the numerical solution is the variation in total pressure throughout the flowfield. There should be no variation in the total pressure since the flow is subcritical.

Figure 13 shows the variation in total pressure from the freestream value along the surface and near the corners. A CFL2D[15] Euler solution for a 131×65 grid is shown for comparison with four k solutions. All solutions exhibit 1 or 2% fluctuations near the corners. After the compression corner, each solution nearly recovers the freestream total pressure. After the expansion corner, each solution shows a total pressure loss. All of the compact k solutions recover after about one unit in the x direction. They should recover since the far-field condition specifies the freestream variables as the outer states. The CFL2D solution allows the total pressure loss to convect all of the way to the downsteam boundary, which is what the far-field condition is supposed to do if an entropy jump occurs in the flowfield. The question, however, remains about the source of the total pressure loss at the corners.

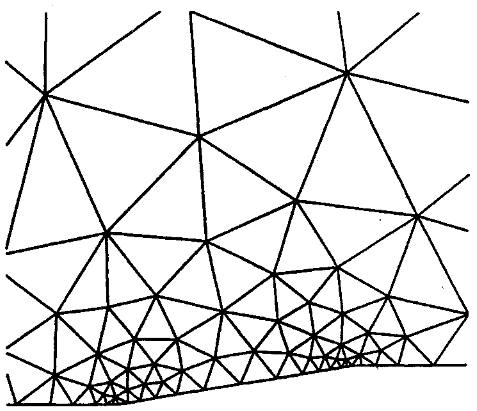

Fig. 11 Unstructured grid for 10-deg ramp case.

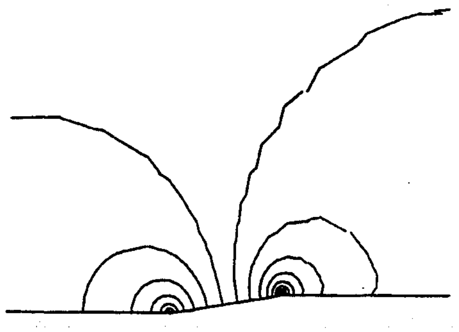

Fig. 12 Piecewise continuous Mach contours.

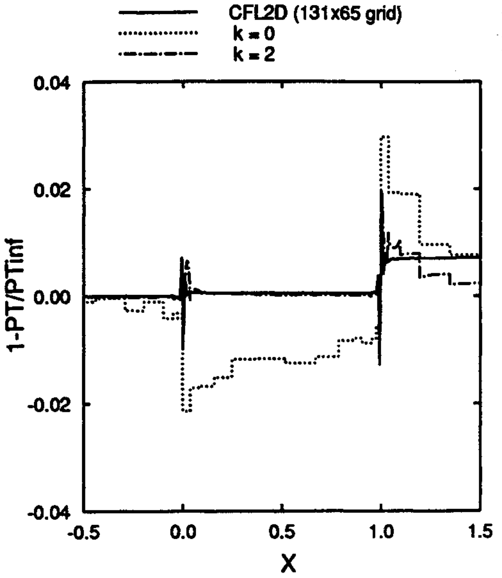

Fig. 13 Surface total pressure variation.

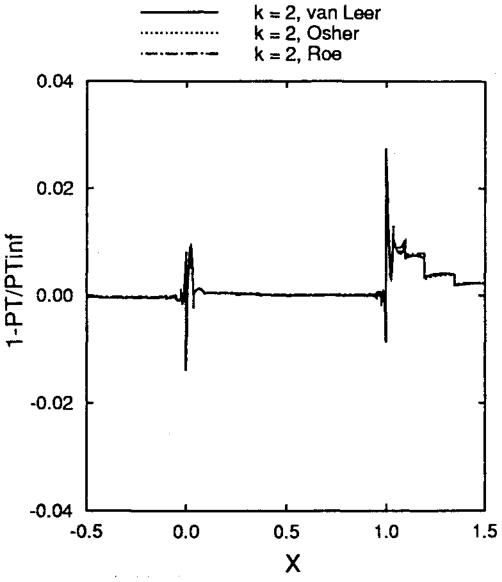

Fig. 14 Effect of flux-splitting type.

One possibility that was investigated is the type of flux splitting used in the numerical scheme. Roe's scheme was used in all five cases of Fig. 13. Solutions using van Leer[13] and Osher and Chakravarthy[14] approximations are shown in Fig. 14. Small variations can be seen near each corner, but all have the same order of total pressure variations. Each scheme is limited by a locally one-dimensional characteristic formulation; however, a multidimensional formulation is not likely to resolve this issue.

Another possible source of total pressure loss may be attributed to the truncation error near the corners. Figure 15 shows the effect of grid density near the corners. The 480 cell grid in Fig. 11 was refined twice by adding grid points near the corners and retriangulating, representing the finer and finest grid solutions shown in Fig. 15. The recovery is shown to be much quicker for the two finer grids; however, the jump in total pressure is of the same order at the corners for each grid. Hence, a zero-order truncation error appears to exist in the corners.

Van der Maarel and Koren[16] similarly observed zero-order entropy errors for subcritical Euler flows over kinked ge-

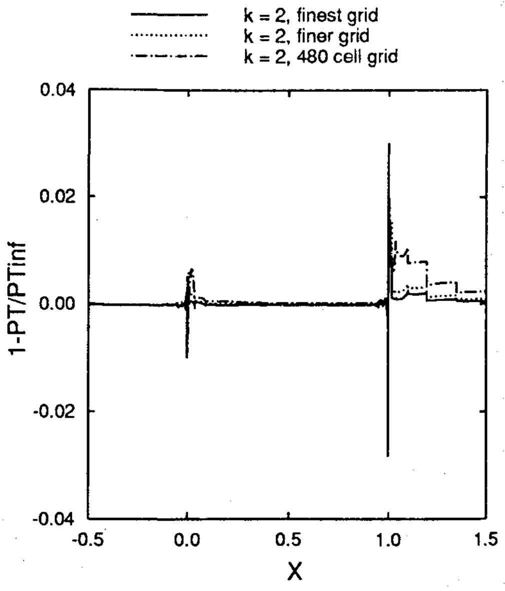

Fig. 15 Effect of grid density near corners.

ometries. They concluded that the source of the error was caused by improper modeling of an unknown type of singularity in the analytical solution at a corner. They proposed modeling the sharp corner by a smooth fillet that approaches the true kink shape in the limit of vanishing grid cell size. In this manner, they could control the error.

The analytical solution to this 10-deg ramp problem was attempted by Verhoff et al.[17] Their hodograph procedure with a small perturbation assumption gives a detailed look at the corner flows. Here, the flow must make an instantaneous change in direction at a nonzero, finite Mach number. Hence, the velocity vector is multivalued at the corners. The numerical solutions approach nearly the same Mach number in the corners. It is understandable that a polynomial approximation near the singularity can misbehave and a better model might be needed.

An accurate noncompact reconstruction near the corners would be difficult because of the solution singularity. All reconstructions that surrounded these points would be prone to having wiggles and errors. It may be more appropriate to reduce the order of accuracy near the corner for a noncompact scheme, but this would require more grid points to maintain lower error levels.

Conclusions

The quadratic and cubic polynomial reconstructions for a given number of cells are demonstrated to provide much more accurate solutions to the shockless transonic Ringleb flow problem than lower order reconstructions. A significant gain in efficiency is also demonstrated over a wide range of accuracy levels since fewer cells are needed by the higher order methods for maintaining the same level of accuracy. The larger cell size accommodates larger time steps, which in turn results in quicker convergence rates. This effect seems to outweigh the burden of extra work needed for the higher order methods. The higher order solutions were computed an order of magnitude faster (and with less memory) at moderate error levels than a lower order solution, which needs far more cells.

The compact reconstruction method is shown to give more accurate higher order results for the Ringleb flow compared with Barth's[7] noncompact method for equivalent orders of polynomial reconstruction. Two orders of magnitude improvement in L2 error is shown for the second method over the noncompact method on the Ringleb flow for cubic polynomials ($k = 3$). The compact approach has advantages over a noncompact scheme near boundaries, discontinuities, and areas of poor grid quality.

Method 2 (the moment method) appears more robust than the first method (the derivative method) primarily due to the simpler boundary conditions. The second method has no problem solving the one or four cell Ringleb flow problem, whereas the first method does not converge for cubic polynomial ($k = 3$) reconstruction. The second method was also more efficient in terms of CPU*memory than the first method for the Ringleb flow case. The first method requires that all cells use the same k level of polynomial reconstruction whereas the second method does not. Hence, the second method may be solution adapted by varying the order k as needed for each cell to spread out the truncation error more evenly. However, this does complicate the vectorization coding.

Acknowledgments

This effort was supported by the McDonnell Douglas Independent Research and Development program while the first author was employed at McDonnell Aircraft Company. Development of the compact higher order method was done in partial fulfillment of requirements for Doctor of Science degree for the first author in the Department of Mechanical Engineering at Washington University in St. Louis, Missouri. The authors would like to thank Rob Vermeland of CRAY Research for computer time on a CRAY YMP supercomputer in Mendota Heights, Minnesota.

References

[1] Jameson, A., and Mavriplis, D., "Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh," AIAA Paper 85-0435, Jan. 1985.

[2] Lohner, R., Morgan, K. and Peraire, J., "Finite Elements for Compressible Flow," *Numerical Methods for Fluid Dynamics,* Oxford Univ. Press, Oxford, England, UK, Vol. II, 1986, pp. 27–53.

[3] Oden, J. T., "A New h-p Adaptive Finite Element Method for High Speed Compressible Flows," AIAA Paper 91-0019, Jan. 1991.

[4] Thareja, R., Stewart, J., Hassan, O., Morgan, K., and Peraire, J., "A Point Implicit Unstructured Grid Solver for the Euler and Navier-Stokes Equations," AIAA Paper 88-0036, Jan. 1988.

[5] Batina, J. T., "Implicit Flux-Split Euler Schemes for Unsteady Aerodynamic Analysis Involving Unstructured Meshes," AIAA Paper 90-0936, April 1990.

[6] Frink, N. T., "A Fast Upwind Solver for the Euler Equations on Three-Dimensional Meshes," AIAA Paper 91-0102, Jan. 1991.

[7] Barth, T., and Frederickson, P., "Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction," AIAA Paper 90-0013, Jan. 1990.

[8] Allmaras, S. R., "A Coupled Euler/Navier-Stokes Algorithm for 2-D Unsteady Transonic Shock/Boundary-Layer Interaction," Ph.D. Thesis, Massachusetts Inst. of Technology, Dept. of Aeronautics and Astronautics, Cambridge, MA, March 1989.

[9] Chiocchia, G. (ed.), "Exact Solutions to Transonic and Supersonic Flows," AGARD Advisory Rept. AR-211, May 1985.

[10] Barth, T. J., and Jesperson, D. C., "The Design and Application of Upwind Schemes on Unstructured Meshes," AIAA Paper 89-0366, Jan. 1989.

[11] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," *Journal of Computational Physics,* Vol. 43, No. 2, 1981, pp. 357–372.

[12] Dunavant, D. A., "High Degree Efficient Symmetrical Gaussian Quadrature Rules for the Triangle," *International Journal for Numerical Methods in Engineering,* Vol. 21, No. 6, 1985, pp. 1129–1148.

[13] van Leer, B., "Flux Vector Splitting for the Euler Equations," *Lecture Notes in Physics,* Vol. 170, 1982, pp. 501–512.

[14] Osher, S., and Chakravarthy, S., "Upwind Schemes and Boundary Conditions with Applications to Euler Equations in General Coordinates," *Journal of Computational Physics,* Vol. 50, No. 3, 1983, pp. 447–481.

[15] Rumsey, C., Taylor, S., Thomas, J., and Anderson, W., "Application of an Upwind Navier-Stokes Code to Two-Dimensional Transonic Airfoil Flow," AIAA Paper 87-0413, Jan. 1987.

[16] van der Maarel, E., and Koren, B., "Spurious Entropy Generation in a Non-Smooth Geometry," Centre for Mathematics and Computer Sciences, Delft, The Netherlands, Rept. NM-R9009, May 1990.

[17] Verhoff, A., Stookesberry, D., and Michal, T., "Hodograph Solution for a Compressible Flow Past a Corner and Comparison with Euler Numerical Predictions," AIAA Paper 91-1547, June 1991.