INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **31**: 121–147 (1999)

A COMPACT HIGH-ORDER UNSTRUCTURED GRIDS METHOD FOR THE SOLUTION OF EULER EQUATIONS

R.K. AGARWAL[a,*] AND D.W. HALT[b,1]

[a] *National Institute for Aviation Research, Wichita State University, Wichita, KS 67260-0093, USA*
[b] *Ford Motor Company, Dearborn, MI, USA*

SUMMARY

Two compact higher-order methods are presented for solving the Euler equations in two dimensions. The flow domain is discretized by triangles. The methods use a characteristic-based approach with a cell-centered finite volume method. Polynomials of order 0 through 3 are used in each cell to represent the conservation flow variables. Solutions are demonstrated to achieve up to fourth-order accuracy. Computations are presented for a variety of fluid flow applications. Numerical results demonstrate a substantial gain in efficiency using compact higher-order elements over the lower-order elements. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: high-order; unstructured grids method; Euler equations

1. 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 extended sets of Euler equations that include as dependent variables not only the associated physical variables 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

---

* Correspondence to: National Institute for Aviation Research, Wichita State University, Wichita, KS 67260-0093, USA. Tel.: +1 316 9785226; e-mail: agarwal@niar.twsu.edu
[1] Tel.: +1 313 3221036; e-mail: dhalt@ford.com

CCC 0271–2091/99/170121–27$17.50

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 equations. 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 non-compact method. Third, only cells that contain flow singularities (e.g. shocks) are complicated by the compact reconstruction process, whereas for non-compact 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 non-compact 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] and transonic flow past a ramp, a sinusoidal bump, NACA 0012 airfoil and NLR 7301 airfoil. For Ringleb flow, 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. Grid sensitivity study is also performed for transonic flow past a sinusoidal bump.

2. GOVERNING EQUATIONS

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

$$\frac{\partial}{\partial t}\iint_v Q\,\mathrm{d}V+\int_s F\cdot\hat{n}\,\mathrm{d}S=0, \tag{1}$$

$$F=G\hat{i}+H\hat{j}, \tag{2}$$

$$Q=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho e_0\end{pmatrix},\quad G=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho u h_0\end{pmatrix},\quad H=\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho v h_0\end{pmatrix} \tag{3}$$

and

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

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.

In Equation (1), 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.

3. METHOD 1—DERIVATIVE EQUATIONS

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

$$\frac{\partial^m}{\partial x^m}\frac{\partial^n}{\partial y^n}\left[\frac{\partial}{\partial t}\iint_v Q \,\mathrm{d}V + \int_s F\cdot\hat{n}\cdot\mathrm{d}S\right] = 0, \tag{5}$$

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:

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

The unknowns are the coefficients of the Taylor series expansion of the conservation variables.

3.1. 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 $Q_{mn} = (1/V)\iint_v (\partial^m/\partial x^m)(\partial^n/\partial y^n)Q \,\mathrm{d}V$ 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 [7]. The reconstruction polynomial is used to determine the left and right states at all Gaussian quadrature points on the shared cell edge as shown in Figure 1. Flux values are evaluated at

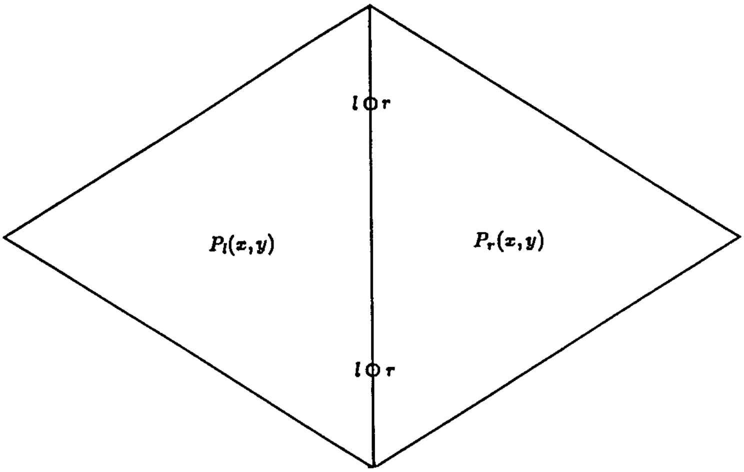


Figure 1. Flux integration on a shared edge by Gaussian quadrature.

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 3 reconstructions and are sufficient for third- and fourth-order spatial accuracy, respectively. Gaussian quadrature procedure is detailed below.

3.2. 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:

$$F_{mn}=\tfrac{1}{2}[F_{mnR}+F_{mnL}-S|\Delta|\Delta^{-1}S^{-1}(F_{mnR}-F_{mnL})], \tag{8}$$

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.

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

The flux formula resembles Roe's scheme [10] 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.

3.3. Flux integration by numerical quadrature

The integration of the fluxes through the cell faces is accomplished by the method of Gaussian quadrature [11]. Given the discrete flux information at certain positions (referred to as Gauss points) along each cell face, a high-order integration can be performed. The location of the Gauss points and their weight factors on a line segment between -1 and $+1$ are listed in Table I.

In Table I, NGP refers to the number of Gauss points and p refers to the highest degree polynomial that can be integrated exactly by this method. The weight parameters are used in the following formula:

$$\int_{-1}^{+1} P(x)\,\mathrm{d}x=\sum_{i=1}^{\mathrm{NGP}} W_i P(x_i). \tag{10}$$

For faces of arbitrary length, the x-locations and weights are linearly scaled by the length.

The Gauss–Lobatto scheme is not used in this study because the end point values must be recomputed for each face since a characteristic analysis is required for the normal direction associated with each face.

Table I. Gaussian quadrature parameters for the line segment

NGP	x-Location	Weight	p
1	0.0	2.0	1
2	−0.577350269189626	1.0	3
	0.577350269189626	1.0	
3	0.0	0.888888888888889	5
	−0.774596669241483	0.555555555555556	
	0.774596669241483	0.555555555555556	
4	−0.339981043584856	0.652145154862546	7
	0.339981043584856	0.652145154862546	
	−0.861136311594053	0.347854845137454	
	0.861136311594053	0.347854845137454	

3.4. 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:

$$\frac{Q_{mni}^{n+1}-Q_{mni}^{n}}{\Delta t}V_i+\sum_{j=1}^{n\text{faces}}(F_{mnj}\cdot\hat{n}_j)A_j=0, \tag{11}$$

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.

4. METHOD 2—MOMENT EQUATIONS

$$\iint_v x^m y^n\left[\frac{\partial}{\partial t}Q+\nabla\cdot F\right]\mathrm{d}V=0. \tag{12}$$

The second method requires the solution of the moment equations, 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 (5) also gives the number of simultaneous equations for the second method. The equation for

$m = n = 0$ is the same as Equation (1) for both methods. All higher-order equations are different for the two methods. The number of equations necessary for a level k compact reconstruction are the same as Equation (6).

To cast Equation (12) in a form where spatial derivatives can be updated in time, Equation (7) is substituted into Q in Equation (12). Using the fact that first moment of inertias about a centroid are zero, the first moment equations reduce to

$$A_{xx}Q_{xt} + A_{xy}Q_{yt} + \int_s xF\cdot\hat{n}\,\mathrm{d}S - \iint_v G\,\mathrm{d}V = 0 \tag{13}$$

and

$$A_{xx}Q_{xt} + A_{yy}Q_{yt} + \int_s xF\cdot\hat{n}\,\mathrm{d}S - \iint_v H\,\mathrm{d}V = 0, \tag{14}$$

where A_{xx}, A_{xy}, and A_{yy} are the second moments of inertia. The evaluation of line and area integrals in Equation (13) is performed as described below.

4.1. Evaluation of the line integral

The Gaussian quadrature described in method 1 is used for the line integration of the fluxes through each face in Equation (13). However, the moment method requires the use of more Gauss points than the derivative method. The moment method uses $k+1$ Gauss points along each face. When fewer than $k+1$ Gauss points are used, wiggles typically appear in the steady state solution indicating a poor numerical modeling of the physics. When more Gauss points are used than $k+1$, the solutions tend to change minimally indicating that more are used than necessary. The optimum number of Gauss points can be attributed to the moment equations where a polynomial of order $k+1$ is multiplied by x^k or y^k, requiring a quadrature of $k+1$ points. When the conservation variables are polynomials of order k, the flux function is really higher-order than $k+1$, but the amount of useful information appears to be limited to the order $k+1$.

4.2. Evaluation of the area integral

The integration of the flux G and H over the cell volume can be done numerically using the efficient quadrature formulas for the triangle by Dunavant [12]. Similar to the one-dimensional Gaussian quadrature formula given by Equation (10), there is a formula for integration of polynomials of high-order over a triangular domain. The formula uses the natural coordinates of an arbitrary triangular domain called the area coordinates. The A_1, A_2 and A_3 values listed in Table II are the area coordinates for the required Gauss points.

For any given point location in the triangular domain, the area coordinates are the ratio of the subtriangle opposite the vertex over the total area. This definition requires the sum of A_1, A_2 and A_3 to be 1.0. Given the area coordinates listed in Table II, the vertices are permuted to establish all possible unique locations of the Gauss points for the arbitrary triangle as shown in Figure 2 for the first four p levels. NGP refers to the number of Gauss points formed by permuting the area coordinates.

Similar to the line integration, the area integration uses the formula appropriate for $2k$ quadrature. Hence, when the dependent variable polynomials are of order $k = [0, 1, 2, 3]$ the quadrature rule used is for $p = [0, 2, 4, 6]$ respectively. This requires [1, 3, 6, 12] Gauss point locations respectively per triangle. The formula for numerical integration over the triangular region is:

$$\iint_A P(x, y)\,\mathrm{d}x\,\mathrm{d}y = \sum_{i=1}^{\mathrm{NGP}} W_i P(x_i, y_i)\,, \tag{15}$$

where W_i refers to the *Weight* parameter in Table II, A is the area of the triangle and (x_i, y_i) are the Gauss point coordinates. The Gauss point coordinates are computed using the summation of the area coordinates times their respective vertex coordinates as follows:

$$\begin{aligned} x_i &= A_1x_1 + A_2x_2 + A_3x_3 \\ y_i &= A_1y_1 + A_2y_2 + A_3y_3 \end{aligned}. \tag{16}$$

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 characteristic-based flux model for this method is simpler than the first method and can use

Table II. Gaussian quadrature parameters for the triangle

p	NGP	A_1	A_2	Weight
1	1	0.333333333333	0.333333333333	1.0
2	3	0.666666666667	0.166666666667	0.333333333333
3	1	0.333333333333	0.333333333333	−0.5625
	3	0.6	0.2	0.520833333333
4	3	0.108103018168	0.445948490916	0.223381589678
	3	0.816847572980	0.091576213510	0.109951743655
5	1	0.333333333333	0.333333333333	0.225
	3	0.059715871790	0.470142064105	0.132394152789
	3	0.797426985353	0.101286507323	0.125939180545
6	6	0.310352451034	0.053145049845	0.082851075618
	3	0.873821971017	0.063089014492	0.050844906370
	3	0.501426509658	0.249286745171	0.116786275726
7	1	0.333333333333	0.333333333333	−0.149570044468
	6	0.312865496005	0.065130102902	0.053347235609
	3	0.869739794196	0.063089014492	0.050844906370
	3	0.479308067842	0.260345966079	0.175615257433

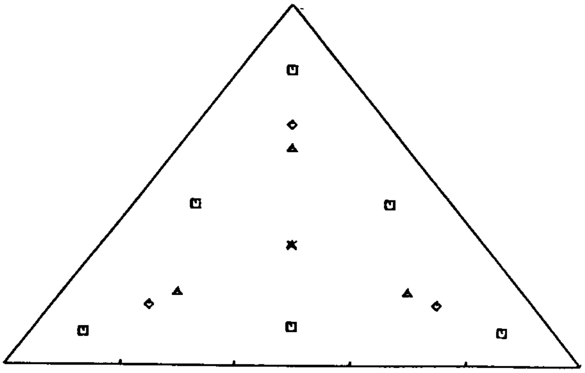


Figure 2. Gauss point locations for area integration of a triangle.

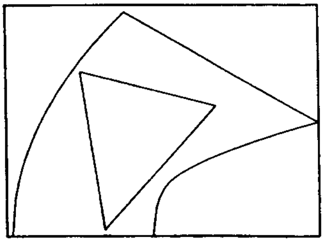


Figure 3. Schematic triangular Ringleb test region.

standard flux-split formulas such as those of Roe [10], van Leer [13], or Osher and Chakravarthy [14].

5. BOUNDARY CONDITIONS

The boundary conditions for the Ringleb flow problem come from the hodograph solution. The Ringleb boundary values are specified from the exact solution [9] 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 over specified.

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 velocity vector at 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.

6. COMPUTATIONAL TEST CASES

6.1. 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 Figure 3. 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 III obtained with the first method. In Table III, 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.

The order of accuracy is determined to be approximately $k + 1$ for each k in Table III. 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 increases as k increases because more equations are solved per cell. For all cases shown in Table III, the residual decayed smoothly like the example

Table III. Ringleb efficiency results using derivative method

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

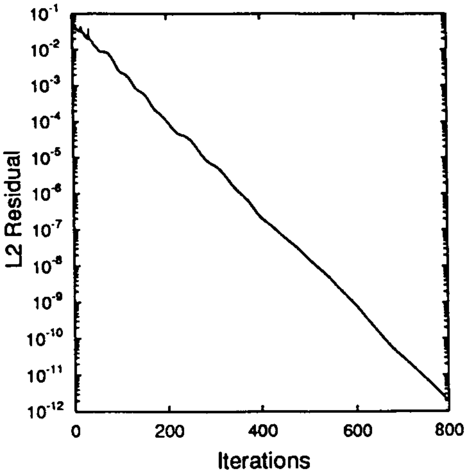


Figure 4. Residual history for 64 cell, $k = 3$ case.

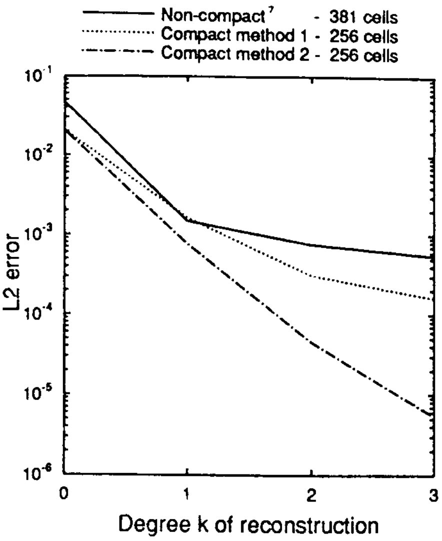


Figure 5. Comparison of L2 errors between compact and non-compact schemes.

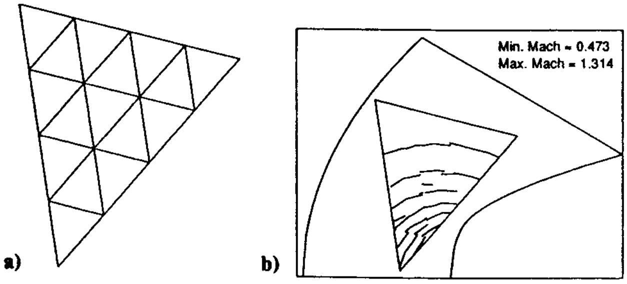


Figure 6. Sixteen cell case: (a) unstructured grid and (b) cellwise continuous Mach contours for $k = 3$.

shown in Figure 4. 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 non-compact scheme of Barth [7] in Figure 5. The first method is nearly a half of an order of magnitude more accurate than the non-compact method for $k = 3$ and furthermore shows a promising slope for extension to $k > 3$. The comparison with the second compact method is 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 Figure 3.

Mach contours are plotted in Figures 6–8 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 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 III. 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 Figure 9. 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.

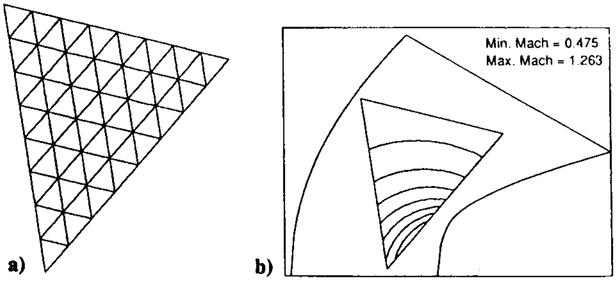


Figure 7. Sixty-four cell case: (a) unstructured grid and (b) cellwise continuous Mach contours for $k = 3$.

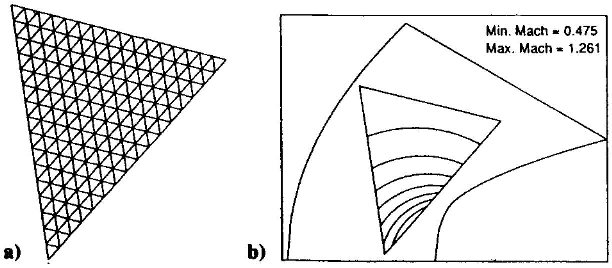


Figure 8. Two hundred and fifty-six cell case: (a) unstructured grid and (b) cellwise continuous Mach contours for $k = 3$.

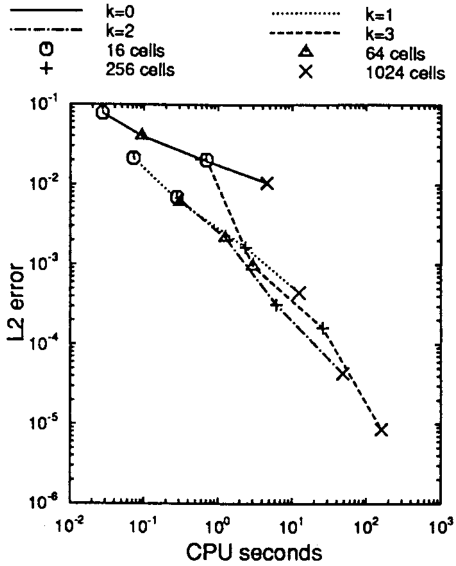


Figure 9. CPU efficiency comparison for method 1.

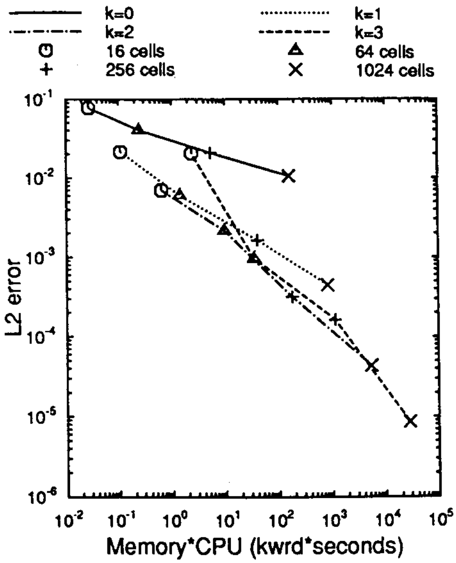


Figure 10. Memory CPU efficiency comparison for method 1.

The memory requirements of the compact scheme for the first method are compared for various error levels in Figure 10. 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 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.

6.2. Ringleb results—method 2

For the second method, a set of computational results is shown in Table IV. 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 III 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 III 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 non-compact method of Barth [7] in Figure 5. 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 non-compact result and an order and a half more accurate than the first compact method. Similar timings between the first compact method and the non-compact method are expected due to the similarity in degree k polynomial reconstructions and orders of Gaussian quadrature. However, the timings of the non-compact calculations are not known.

An efficiency comparison is also made for the second method. Total CPU times were calculated from the data in Table IV, again assuming six orders of magnitude residual reduction for the convergence criterion. Figure 11 shows total CPU time on 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.

The spatial orders of accuracy are slightly degraded for the second method. Orders of accuracy for the results in Table IV 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 $k = 1, 2, 4, 6$, 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.

The underrelaxation 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.

Table IV. Ringleb efficiency results using moment method

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	284	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^{-4}	0.986	20	15.7
1	1024	2.04×10^{-4}	0.992	20	62.4
2	1	4.39×10^{-2}	0.951	670	0.13
2	4	9.54×10^{-3}	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

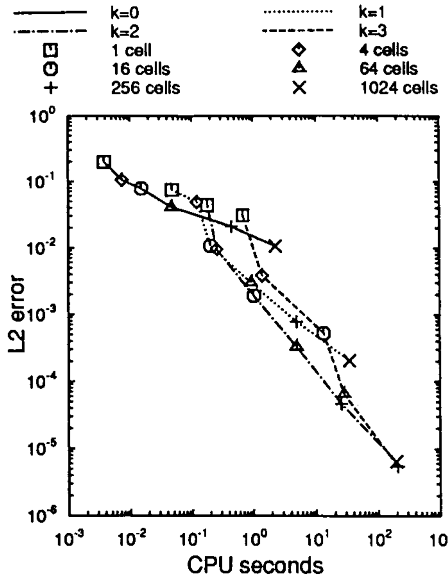


Figure 11. CPU efficiency comparison for method 2.

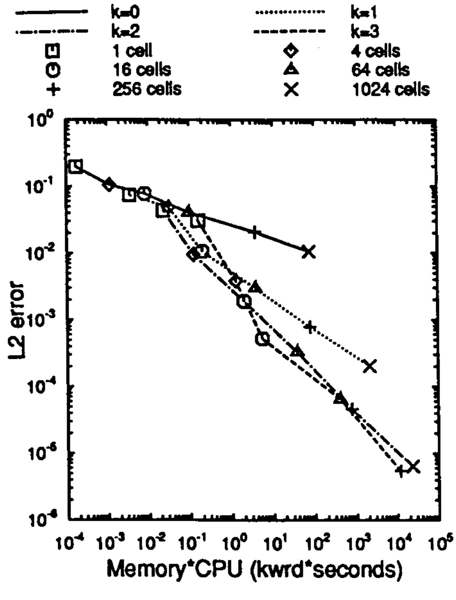


Figure 12. Memory CPU efficiency comparison for method 2.

Figure 12 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 Figure 12. 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. Because of robustness and ease in implementation of the boundary conditions, the rest of the calculations presented in this paper are based on method 2.

6.3. 10-Deg ramp

A 10-deg ramp case was analysed 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 Figure 13. 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 Figure 14. 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 15 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

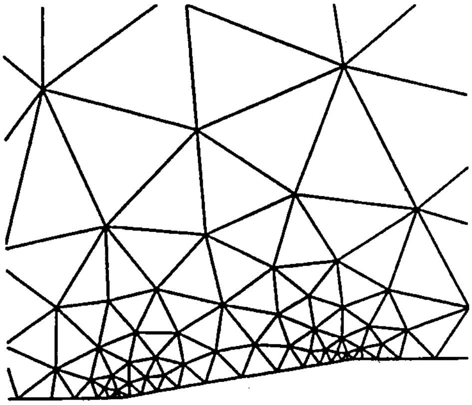


Figure 13. Unstructured grid for 10-deg ramp case.

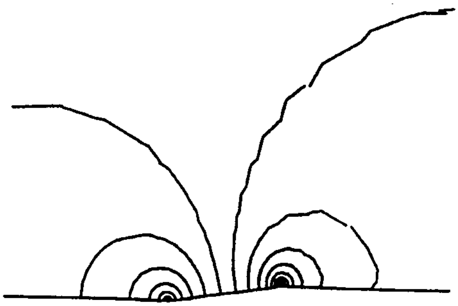


Figure 14. Piecewise continuous Mach contours.

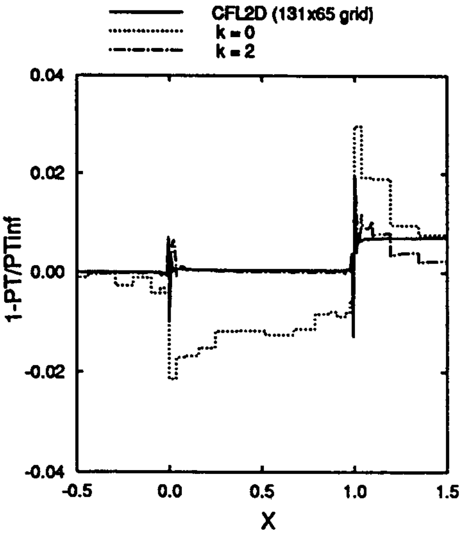


Figure 15. Surface total pressure variation.

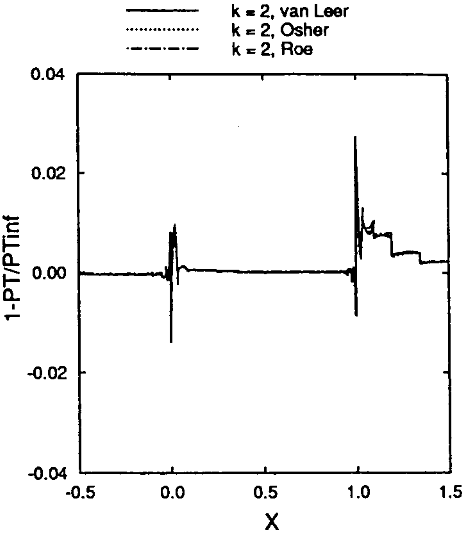


Figure 16. Effect of flux-splitting type.

total pressure loss to convect all the way to the downstream 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.

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 Figure 15. Solutions using van Leer [13] and Osher and Chakravarthy [14] approximations are shown in Figure 16. 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 multi-dimensional 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 17 shows the effect of grid density near the corners. The 480 cell grid in Figure 13 was refined twice by adding grid points near the corners and retriangulating, representing the finer and finest grid solutions shown in Figure 17. 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 geometries. 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.

6.4. Sinusoidal bump

The flow over a sinusoidal bump at freestream Mach number of 0.3 was selected as a test case to assess the accuracy of the curved edge model. The sinusoidal bump refers to a sine wave raised to the fourth power between 0 and pi. This maintains continuity of first, second and third derivatives at the ends of the sinusoidal function where it is extended by horizontal lines ten chordlengths to the far-field boundaries. An unstructured grid of 243 grid points and 425 cells was generated using Lohner's advancing front grid generator [2] and is shown in Figure 18 for $k = 0$ through $k = 3$ reconstruction. The pressure distribution converges quickly as the level of reconstruction is increased.

A solution where only straight edges were used is shown for comparison in Figure 19. A large amount of error is generated when multiple Gauss points are used to model straight edges since the surface slopes are discontinuous. Similarly, cellwise continuous Mach contours are shown in Figure 20 for a compact cubic reconstruction solution using the boundary condition for a curved surface. The curved edge model is continuous in surface slope and produces very smooth contours which are nearly symmetric.

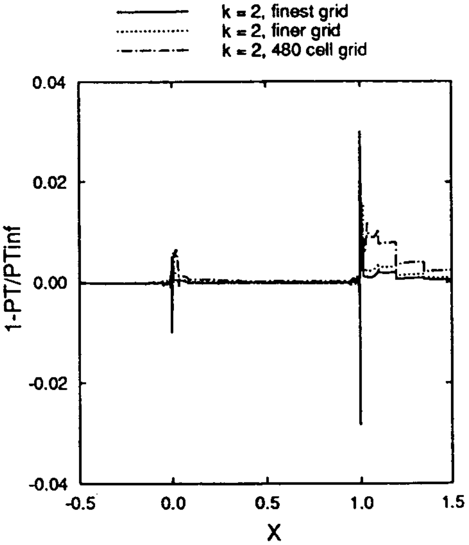


Figure 17. Effect of grid density near corners.

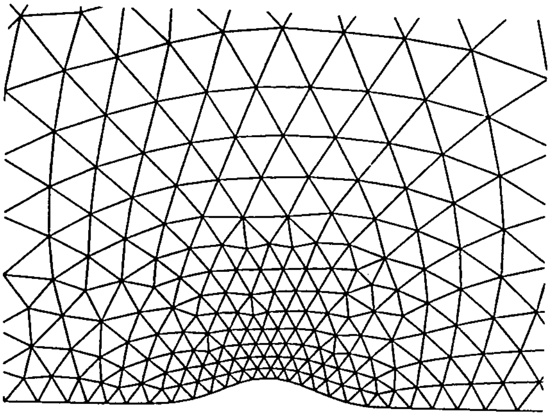


Figure 18. Unstructured grid for sinusoidal bump case.

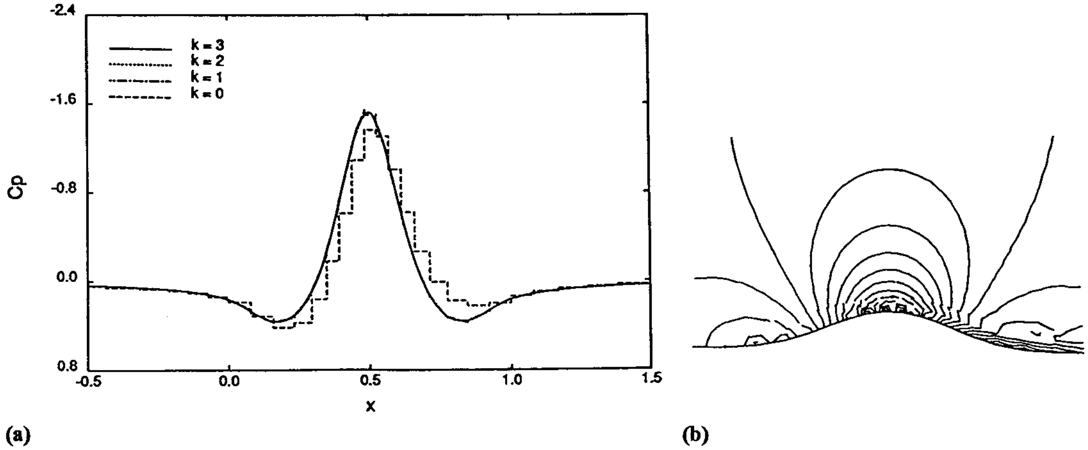


Figure 19. (a) Surface pressure coefficient distribution for bump case using straight edge model. (b) Cellwise continuous Mach contours for bump case using straight edge model.

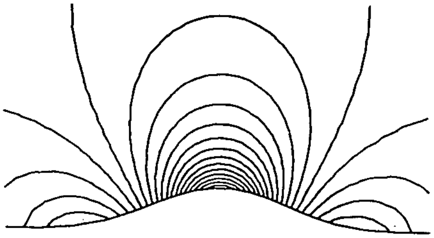


Figure 20. Cellwise continuous Mach contours for bump case.

A more quantifiable measure of solution error is the deviation in total pressure from freestream. The surface total pressure error decays rapidly as the order of the cell polynomial increases as shown in Figure 21. The total pressure errors are plotted against chord position for reconstructions of $k = 0$ through $k = 3$. The integrated total pressure RMS error levels are

4.5×10^{-3}, 4.4×10^{-4}, 4.3×10^{-5}, 2.5×10^{-5} for $k = 0, 1, 2, 3$, respectively. An increase in accuracy of a full order of magnitude results for each successive increment in order of reconstruction with the exception of $k = 3$. At this level of accuracy, the error in the far-field approximation at ten chordlengths prevents higher-order reconstructions from lowering the RMS error much further.

An efficiency comparison is performed for the sinusoidal bump case between CFL2D [15] and the compact higher-order method. CFL2D is used for three grid sizes of 61×21, 121×41 and 241×81. The compact higher-order method is used on the previous grid with the order of reconstruction ranging from $k = 1$ to 3. Figure 22 compares the efficiency of each result. Plotted along the x axis is the total CRAY-YMP CPU time used to converge the L2 norm of the continuity equation residual by three orders of magnitude. The deviation in the peak pressure coefficient is plotted for each Euler solution along the y axis. The results show the compact higher-order method to be much more efficient in reducing the peak pressure error. The $k = 2$ solution is about an order of magnitude more accurate than the CFL2D solution on the 241×81 grid and furthermore uses an order of magnitude less CPU time.

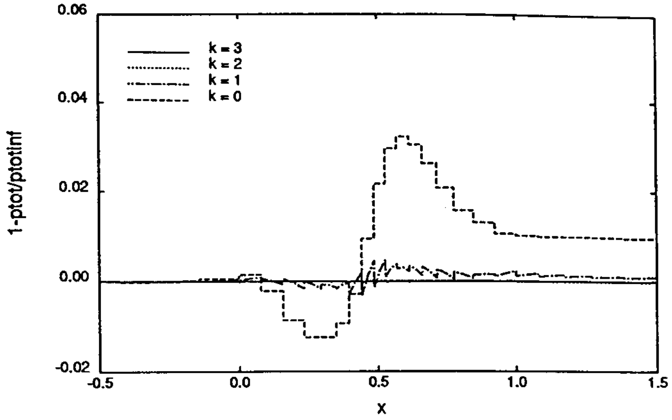


Figure 21. Surface total pressure deviation for bump case.

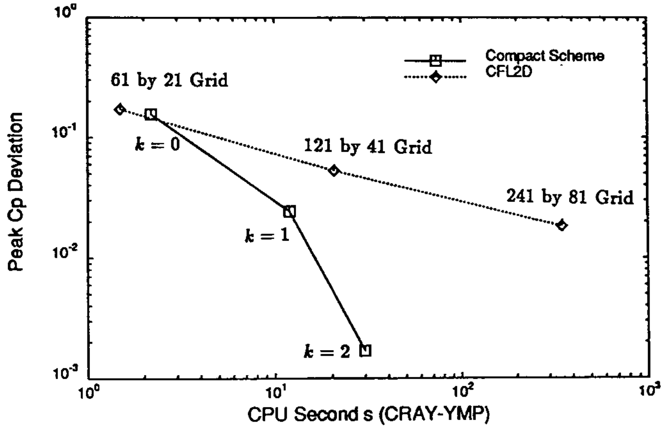


Figure 22. Code efficiency comparison for bump case.

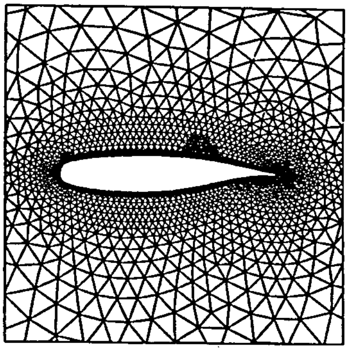


Figure 23. Unstructured grid for NLR 7301 airfoil case.

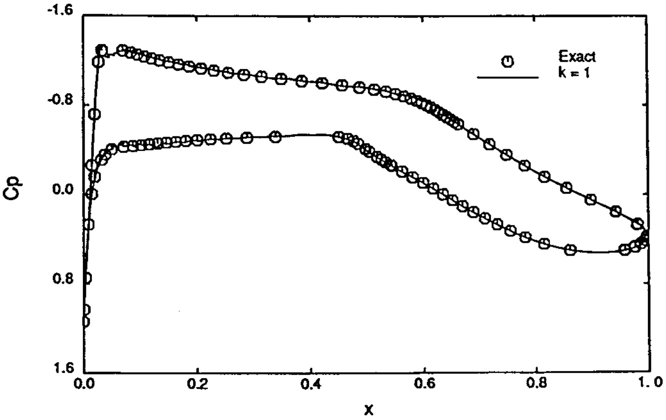


Figure 24. Surface pressure coefficient distribution for NLR airfoil case.

6.5. *NLR 7301 airfoil*

The NLR 7301 airfoil was run at Mach 0.721 and − 0.194° angle of attack. The unstructured grid shown in Figure 23 is much coarser in comparison to those in [17], yet accurate results are obtained. The grid was generated from the 2D advancing front grid generator of Lohner [2]. The grid is composed of 2379 grid points and 4496 triangles with 234 points on the airfoil surface. The far-field is a rectangular box which extends ten chordlengths from the airfoil surface. A point vortex was added to the far-field boundary condition to better model the circulation in the far-field. A compact higher-order solution is shown in Figure 24 in comparison to the hodograph computed pressures [17]. The pressures match the hodograph solution very well with a small exception near the point on the upper surface where the flow turns from supersonic to subsonic.

6.6. *NACA 0012 airfoil*

The standard AGARD [17] 2D Euler test case of flow over a NACA 0012 airfoil at a freestream Mach number of 0.8 and angle of attack of 1.25° was chosen to test the shock capturing ability of the compact high-order method. A high-order solution was computed on the unstructured grid of 848 grid points and 1586 triangles with 134 points on the airfoil surface shown in Figure 25. The results of Jameson and Schmidt [17] using a structured grid of 320 × 64 quadrilaterals are shown for comparison in Figure 26. A close comparison in surface pressures can be seen everywhere except near the shocks on both upper and lower surfaces. The stronger shock on the upper surface is captured extremely well with the current compact high-order scheme except for a small pressure overshoot after the shock. The weaker shock on the lower surface exhibits pressure overshoots on both sides of the shock. There is no TVD limiter or added dissipation (beyond that of the upwind scheme) in the high-order method. It appears that a small amount of added dissipation or a TVD limiter could eliminate the overshoots.

7. GRID SENSITIVITY STUDY

Grid sensitivity can effect the accuracy of the numerical solution from a particular code. Smoothness is often a requirement for grids in order to obtain high quality numerical results.

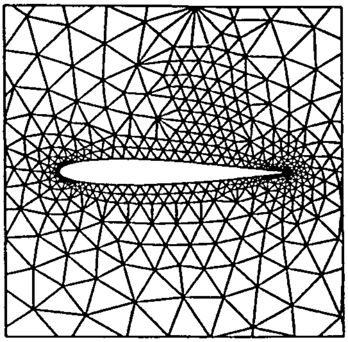


Figure 25. Unstructured grid for NACA 0012 airfoil case.

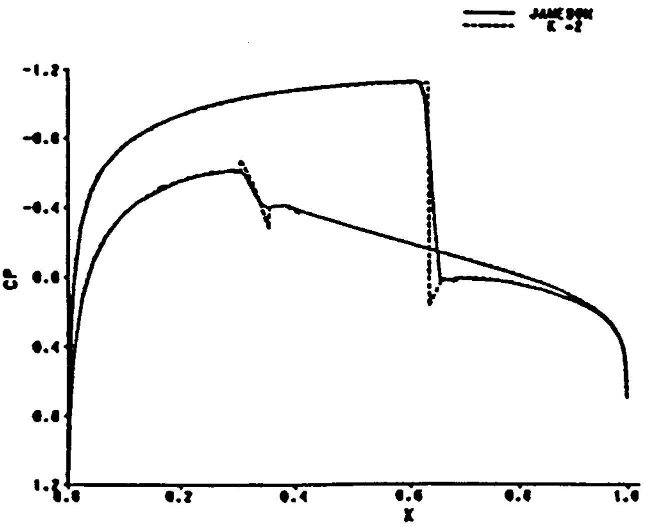
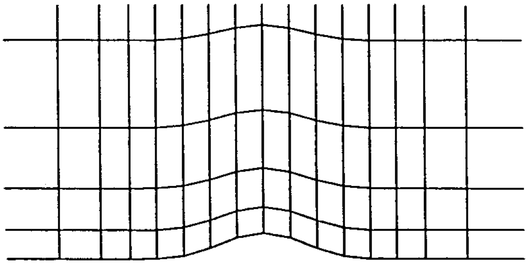
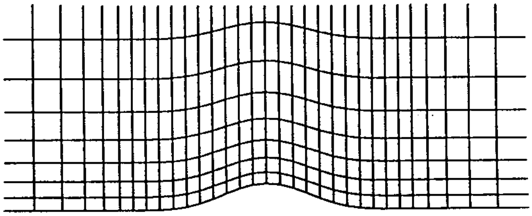


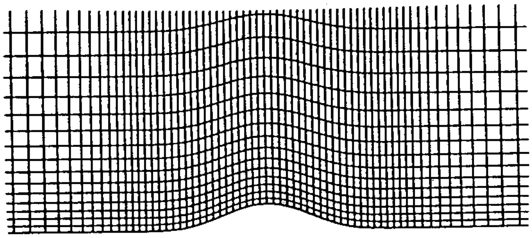
Figure 26. Surface pressure coefficient distribution for NACA airfoil case.



(a) 31 by 11 Grid



(b) 61 by 21 Grid



(c) 121 by 41 Grid

Figure 27. Smooth structured grids for grid sensitivity study.

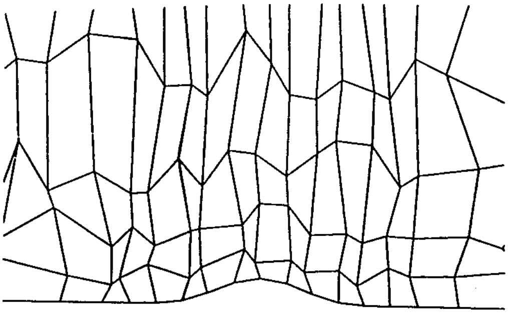
Some codes are much more sensitive to grid smoothness than others. The objective of this study is to ascertain the sensitivity of the compact high-order solver with respect to grid smoothness.

A matrix of flow solutions for the sinusoidal bump case is computed for an array of grids. A sequence of three smooth grids is generated for the structured grid flow solver, CFL2D [15], as shown in Figure 27. Each successive grid has twice the resolution in each coordinate direction. The first grid has 30×10 cells, followed by 60×20 cells and finally 120×40 cells. An order of accuracy can be computed by monitoring the numerical error for each case.

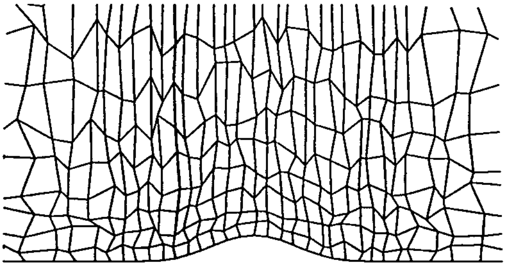
Each of these three grids is randomized as shown in Figure 28 by allowing the internal grid points to float randomly within a specified space. Each internal grid point is allowed to float a random distance up to 40% of the distance to the neighboring point. The process uses a random number in each direction for each point between -0.40 and 0.40. Then a bilinear element fit is applied to the appropriate quadrant the point is moving toward. This assures that no grid crossover or singularity will occur in this process. The smooth and randomized structured grids are used with CFL2D to generate numerical solutions.

These sets of grid points are used to generate triangular unstructured grids using the Delaunay approach. However, since the compact high-order solutions have been shown above to be much more accurate than the CFL2D solutions, the largest set of unstructured grid points is replaced with a set of points with half the resolution of the coarsest structured grid case. The unstructured grids for the smooth set of points are shown in Figure 29. The randomized points are used to generate the unstructured grids shown in Figure 30.

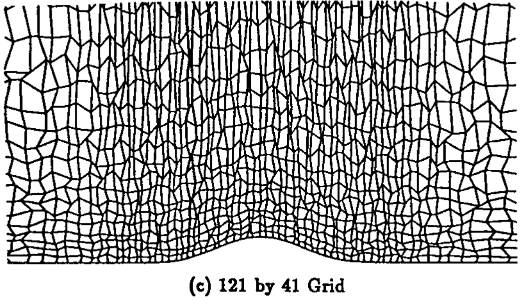
A matrix of steady state numerical solutions are generated and compared in Figures 31 and 32. Two representative error levels are plotted against the number of x faces, which is equivalent to the number of structured grid cells in the x-direction. The peak pressure coefficient error is estimated by assuming the best high-order solution ($k = 3$) has the correct result for peak pressure coefficient. Unfortunately, this assumption causes the accuracy of the $k = 1$ and 2 reported errors to behave a little erratically, leaving some question as to the validity of the measurement of grid sensitivity. This question can be alleviated by simultaneously monitoring the mean total pressure error in Figure 32. The series of results indicate that the compact high-order scheme is not very sensitive to the random scattering process used. Accuracy levels tend to drop off slightly with the randomized grids, however, the order of accuracy is often unchanged. It is interesting to note that the solutions converged much faster



(a) 31 by 11 Grid

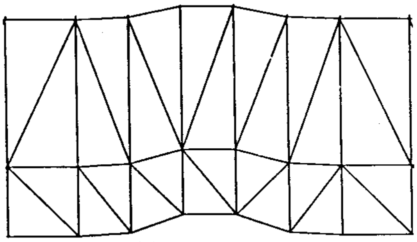


(b) 61 by 21 Grid

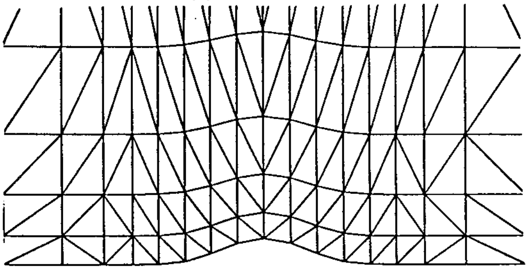


(c) 121 by 41 Grid

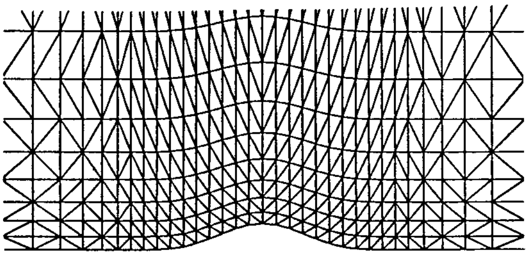
Figure 28. Randomized structured grids for grid sensitivity study.



(a) 16 by 6 Grid



(b) 31 by 11 Grid



(c) 61 by 21 Grid

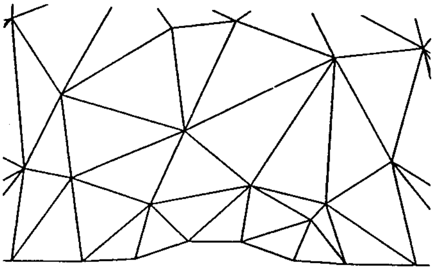
Figure 29. Smooth unstructured grids for grid sensitivity study.

on the randomized grids than the smooth grids. CFL2D results also show little effect from the randomization process.

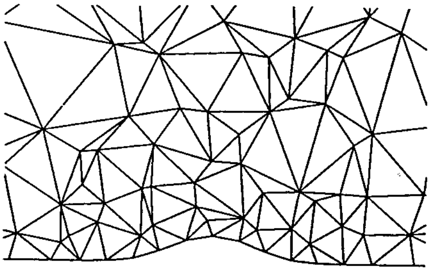
The order of accuracy can be approximated by the slope of the lines in the figures. In each case, the order of accuracy of the compact high-order scheme is a little less than the order $k+1$ as expected. CFL2D requires roughly twice the grid resolution in each direction as the compact unstructured scheme using only $k=1$. The higher k values are shown to provide much more accurate solutions.

8. CONCLUSIONS

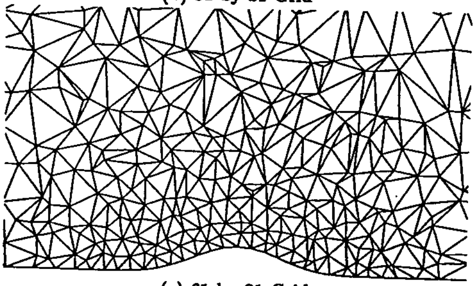
The quadratic and cubic polynomial reconstructions for a given number of cells are demonstrated to provide much more accurate solutions to the shockless transonic Ringleb flow problem than lower-order reconstructions. A significant gain in efficiency is also demonstrated over a wide range of accuracy 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



(a) 16 by 6 Grid



(b) 31 by 11 Grid



(c) 61 by 21 Grid

Figure 30. Randomized unstructured grids for grid sensitivity study.

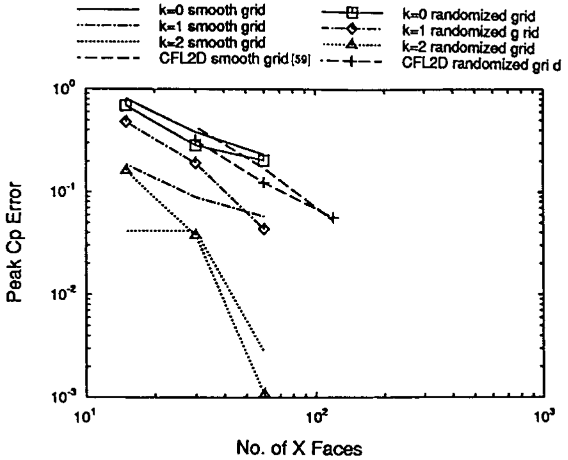


Figure 31. Peak C_p error for randomized grid study.

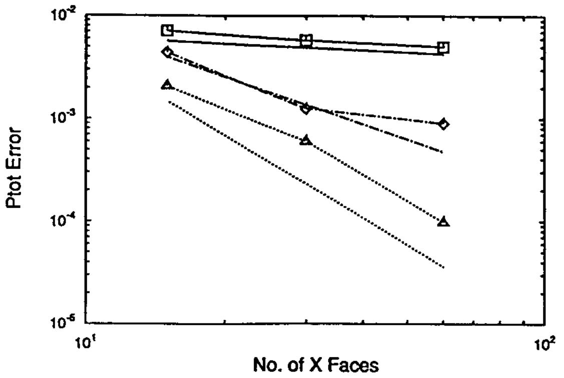


Figure 32. Total pressure error for randomized grid study.

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] non-compact method for equivalent orders of polynomial reconstruction. Two orders of magnitude improvement in L2 error is shown for the second method over the non-compact method on the Ringleb flow for cubic polynomials ($k = 3$). The compact approach has advantages over a non-compact 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.

REFERENCES

1. A. Jameson and D. Mavriplis, 'Finite volume solution of the two-dimensional Euler equations on a regular triangular mesh', *AIAA Paper 85-0435*, January, 1985.
2. R. Lohner, K. Morgan and J. Peraire, 'Finite elements for compressible flow', in *Numerical Methods for Fluid Dynamics*, vol. 2, Oxford University Press, Oxford, UK, 1986, pp. 27–53.
3. J.T. Oden, 'A new *h–p* adaptive finite element method for high speed compressible flows', *AIAA Paper 91-0019*, January, 1991.
4. R. Thareja, J. Stewart, O. Hassan, K. Morgan and J. Peraire, 'A point implicit unstructured grid solver for the Euler and Navier–Stokes equations', *AIAA Paper 88-0036*, January, 1988.
5. J.T. Batina, 'Implicit flux-split Euler schemes for unsteady aerodynamic analysis involving unstructured meshes', *AIAA Paper 90-0936*, April, 1990.
6. N.T. Frink, 'A fast upwind solver for the Euler equations on three-dimensional meshes', *AIAA Paper 91-0102*, January, 1991.
7. T. Barth and P. Frederickson, 'Higher-order solution of the Euler equations on unstructured grids using quadratic reconstruction', *AIAA Paper 90-0013*, January, 1990.
8. S.R. Allmaras, 'A coupled Euler/Navier–Stokes algorithm for 2D unsteady transonic shock/boundary-layer interaction', *Ph.D. Thesis*, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, MA, March, 1989.
9. Chiocchia, G. (ed.), 'Exact solutions to transonic and supersonic flows', *AGARD Advisory Report AR-211*, May, 1985.

10. P.L. Roe, 'Approximate Riemann solvers, parameter vectors and difference schemes', *J. Comput. Phys.*, **43**(2), 357–372 (1981).
11. A.H. Stroud, *Numerical Quadrature and Solution of Ordinary Differential Equations*, Springer-Verlag, New York Inc., 1974.
12. D.A. Dunavant, 'High degree efficient symmetrical Gaussian quadrature rules for the triangle', *Int. J. Numer. Methods Eng.*, **21**(6), 1129–1148 (1985).
13. B. van Leer, 'Flux vector splitting for the Euler equations', *Lecture Notes in Physics*, **170**, 501–512 (1982).
14. S. Osher and S. Chakravarthy, 'Upwind schemes and boundary conditions with applications to Euler equations in general coordinates', *J. Comput. Phys.*, **50**(3), 447–481 (1983).
15. C. Rumsey, S. Taylor, J. Thomas and W. Anderson, 'Application of an upwind Navier–Stokes code to two-dimensional transonic airfoil flow', *AIAA Paper 87-0413*, January, 1987.
16. E. Van der Maarel and B. Koren, 'Spurious entropy generation in a non-smooth geometry', *Report NM-R9009*, Centre for Mathematics and Computer Sciences, Delft, The Netherlands, May, 1990.
17. G. Chiocchia, 'Test cases for inviscid flow field methods', *AGARD Advisory Report AR-211*, May 1985.