High order solution of the Euler equations on unstructured grids quadratic reconstruction

Article · February 1990

DOI: 10.2514/6.1990-13 · Source: NTRS

2 authors, including:

Paul Oliver Frederickson
Math Cube Associates, Inc.
33 PUBLICATIONS **3,533** CITATIONS

SEE PROFILE

AIAA'90

AIAA-90-0013

HIGHER ORDER SOLUTION OF THE EULER EQUATIONS ON UNSTRUCTURED GRIDS USING QUADRATIC RECONSTRUCTION

Timothy J. Barth
CFD Branch
NASA Ames Research Center
Moffett Field, CA

Paul O. Frederickson
RIACS
NASA Ames Research Center
Moffett Field, CA

28th Aerospace Sciences Meeting
January 8-11, 1990/Reno, Nevada

Higher Order Solution of the Euler Equations on Unstructured Grids using Quadratic Reconstruction†

Timothy J. Barth
CFD Branch
NASA Ames Research Center
Moffett Field, Ca

Paul O. Frederickson ††
RIACS
NASA Ames Research Center
Moffett Field, Ca

Abstract

High order accurate finite-volume schemes for solving the Euler equations of gasdynamics are developed. Central to the development of these methods are the construction of a k-exact reconstruction operator given cell-averaged quantities and the use of high order flux quadrature formulas. General polygonal control volumes (with curved boundary edges) are considered. The formulations presented make no explicit assumption as to complexity or convexity of control volumes. Numerical examples are presented for Ringleb flow to validate the methodology.

1. Introduction.

In this paper we develop a high order finite-volume algorithm for the Euler equations on arbitrary unstructured meshes. This algorithm is a finite-volume extension of Godunov's method and breaks naturally into three distinct operations: high order reconstruction of the solution in each cell, high order numerical flux quadrature on each edge, and time integration (evolution) in each cell. Particular attention is paid to solution reconstruction and flux quadrature since accuracy of these operations determines the spatial accuracy of the overall scheme for steady state problems.

The development presented here follows many of the ideas developed previously for structured meshes. For example, in the extension of Godunov's scheme [1] to second order accuracy in one space dimension, van Leer [2] developed an advection scheme based on discontinuous piecewise linear distributions together with Lagrangian hydrodynamics. Colella and Woodward [3] and Woodward and Colella [4] further extended these ideas to include discontinuous piecewise parabolic approximations with Eulerian or Lagrangian hydrodynamics. Harten et. al. [5,6] later extended related schemes to arbitrary order and clarified the entire process. These techniques have been applied to multi-dimensional structured meshes via individual one-dimensional application along coordinate lines. This has proven to be a highly successful approximation but does not directly extend to unstructured meshes. Nevertheless, we can still extract the basic ideas from this previous work and generalize them to unstructured meshes.

Several finite-volume schemes for unstructured meshes based on both continuous and discontinuous piecewise linear approximation have been presented in the literature. Note that when using continuous linear approximations on triangles, the distinction between finite-volume and Galerkin finite-element (with mass matrix lumping) is slight. Jameson and Mavriplis [7], Mavriplis and Jameson [8,9], Morgan and Peraire [10], Lohner, Morgan, and Peraire [11] and others have constructed schemes based on continuous piecewise linear elements with added artificial viscosity and/or flux corrected transport terms. In previous work by the first author (Barth and Jespersen [12]), a discontinuous piecewise linear reconstruction technique for arbitrary unstructured meshes was developed and incorporated into a finite-volume flux-split algorithm. Unlike finite-elements, we do *not* place restrictions on the shape of the cell, i.e. that they be triangles or quadrilaterals. This approach [12] also differs conceptually from finite-volume-like algorithms of Rostand and Stoufflet [13] and Desideri and Dervieux [14] which do not actually construct a unique piecewise approximation in each cell.

We have restricted the numerical calculations presented in this paper to smooth flows with analytical solutions for the purpose of assessing solution accuracy. The modifications for nonoscillatory shock capturing follow the ideas presented in [12] and will be presented elsewhere. We will use the Ringleb flow (a hodograph solution of the gasdynamic equations, see [15]) as a test case for measuring solution accuracy since absolute errors can be measured. Although we will show results indicating that quadratic reconstruction is adequate for Ringleb flow, we do not restrict our attention to quadratics but will present results for piecewise polynomials of higher order.

---

†† Work reported here was supported, in part, by Cooperative Agreement NCC 2-387 between the National Aeronautics and Space Administration (NASA) and the Universities Space Research Association (USRA).

2. A Finite-Volume Algorithm for the Euler Equations.

In developing a finite-volume scheme, we first consider the integral form of the Euler equations for some domain, Ω and its tessellation $\mathcal{T}(\Omega)$ comprised of cells, c_i. In each cell we have

$$\frac{d}{dt}\int_{c_i} \mathbf{u}\, da + \int_{\partial c_i} \mathbf{F}(\mathbf{u},\mathbf{n})\, dl = 0 \qquad (2.0)$$

In this equation, u represents the vector of conserved variables for mass, momentum, and energy. The vector $\mathbf{F}(\mathbf{u},\mathbf{n})$ represents the Euler flux projected onto the local exterior normal of the cell. The higher order finite-volume extension of Godunov's scheme to arbitrary shaped cells begins by assuming distinct piecewise polynomial approximations in each cell (expanded about the centroid)

$$u^k(x,y) = \sum_{m+n\le k} \alpha_{(m,n)} P_{(m,n)}(x-x_c, y-y_c) \quad (2.1)$$

where u^k is a particular component of the vector $\mathbf{u}^k$ and $P_{(m,n)}(x-x_c, y-y_c) = (x-x_c)^m (y-y_c)^n$. Because the piecewise polynomials are discontinuous from cell to cell, along a cell boundary two distinct values of the solution can be obtained. To resolve this nonuniqueness, the Euler flux is supplanted by a "numerical flux function", $\overline{\mathbf{F}}(\mathbf{u}_+, \mathbf{u}_-, \mathbf{n})$, which when given two solution states produces a single unique flux. Approximating (2.0) by piecewise polynomials and a numerical flux function we obtain

$$\frac{d}{dt}\int_{c_i} \mathbf{u}^k\, da + \int_{\partial c_i} \overline{\mathbf{F}}(\mathbf{u}^k_+, \mathbf{u}^k_-, \mathbf{n})\, dl = 0 \qquad (2.2)$$

The solution algorithm for (2.2) is a relatively standard procedure for extensions of Godunov's scheme in Eulerian coordinates [6,12,16,17]. The process consists of three basic operations:

(1) **Reconstruction in Each Cell:** Given integral cell averages in all cells, reconstruct piecewise polynomial coefficients $\alpha_{(m,n)}$ for use in equation (2.1).

(2) **Flux Evaluation on Each Edge:** Consider each cell boundary, ∂c_i, to be a collection of edges from the mesh. Along each edge, perform a high order accurate flux quadrature.

(3) **Evolution in Each Cell:** Collect flux contributions in each cell and evolve in time using any time stepping scheme, i.e., Euler explicit, Euler implicit, Runge-Kutta, etc.

We describe the reconstruction operator in some detail in the following section, and present validating numerical results in section 5. Our algorithm for flux evaluation on an edge is described in detail in section 4, and numerical results validating the entire finite-volume algorithm, including numerical flux evaluation, are to be found in section 6.

3. Reconstruction.

The development of a reconstruction operator follows from the application of a few simple properties that in one dimension are relatively well known but often not explicitly stated. In multiple dimensions the same properties apply and are known by many but are almost never stated and sometimes violated. In this section we state the relevant properties and how they are used to design a reconstruction operator $\mathbf{R}^k$ which reconstructs k-th order piecewise polynomials. This reconstruction operator serves as a finite-dimensional pseudo inverse of the cell-averaging operator $\mathbf{A}$ whose i-th component $\mathbf{A}_i$ computes the cell average of the solution in c_i

$$\overline{u}_i = \mathbf{A}_i u = \int_{c_i} u(x,y)\, da$$

Reconstruction Design Criteria

(1)Conservation of the mean: Simply stated, given cell averages $\overline{u}$, we require that all polynomial reconstructions u^k have the correct cell average. That is to say,

$$\text{if } \; u^k = \mathbf{R}^k \overline{u} \; \text{ then } \; \overline{u} = \mathbf{A} u^k$$

More concisely,

$$\mathbf{A}\mathbf{R}^k = I \qquad (3.0)$$

(In words, $\mathbf{R}^k$ is a right inverse of the averaging operator $\mathbf{A}$.)

(2) k-exactness: We say that a reconstruction operator $\mathbf{R}^k$ is *k-exact* if $\mathbf{R}^k\mathbf{A}$ reconstructs polynomials of degree k or less exactly. Denoting by $\mathcal{P}_k$ the space of all piecewise polynomials of degree k, we require,

$$\text{if } \; u \in \mathcal{P}_k \; \text{ and } \; \overline{u} = \mathbf{A}u, \; \text{ then } \; u^k = \mathbf{R}^k \overline{u} = u$$

This can be written succinctly as

$$\mathbf{R}^k \mathbf{A}\Big|_{\mathcal{P}_k} = I \qquad (3.1)$$

(In words, $\mathbf{R}^k$ is a left-inverse of $\mathbf{A}$ restricted to the space of polynomials of degree at most k.)

(3)Compact Support: The third of our criteria in the selection of a reconstruction operator $\mathbf{R}^k$ involves the matter of locality: we insist that reconstructed polynomial u^k on the cell $\mathbf{c}_i$ be independent of the values $\overline{u}$ takes on outside of a relatively small neighborhood $\mathcal{N}_i$. See for example Fig. 1 which shows a

representative cell and its neighbors (distance 1 and 2 in dual graph).

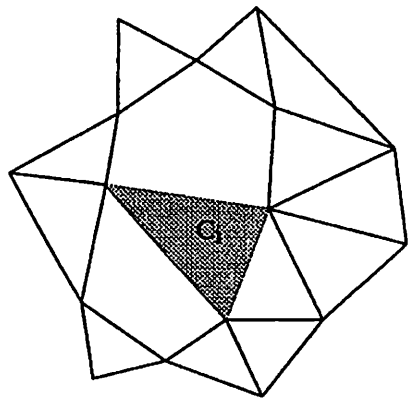

FIGURE 1. Typical Cell with distance 1 & 2 neighbors.

Recall that a polynomial of degree k in two dimensions contains $\binom{k+2}{2} = (k+1)(k+2)/2$ coefficients or degrees of freedom. We require that the support set be at least this large, but as the support set becomes larger not only does the computational cost increase, but eventually the accuracy decreases as less valid data from farther away is brought in.

(4)**Efficiency**: Since the numerical implementation of (2.2) will require repeated reconstruction in each cell, we seek to optimize efficiency by precomputing as a *one time* preprocessing step the set of weights $\mathbf{W}_i = W_{(m,n)j}$ in each cell c_i such that

$$\alpha_{(m,n)} = \sum_{j \in \mathcal{N}_{c_i}} W_{(m,n)j} \overline{u}_j \tag{3.2}$$

We have effectively reduced the problem of reconstruction to multiplication of predetermined weights and cell averages to obtain polynomial coefficients.

Normalized Local Coordinates

During the preprocessing to obtain the reconstruction weights $\mathbf{W}_i$ we assume a coordinate system with origin at the centroid of c_i to minimize roundoff errors. We then temporarily transform (rotate and scale) to another coordinate system $(\overline{x}, \overline{y})$ which is normalized to the cell $\mathbf{c}_i$

$$\begin{bmatrix} \overline{x} \\ \overline{y} \end{bmatrix} = \begin{bmatrix} \mathbf{D}_{1,1} & \mathbf{D}_{1,2} \\ \mathbf{D}_{2,1} & \mathbf{D}_{2,2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

with the matrix $\mathbf{D}$ chosen so that

$$\mathbf{A}_i(\overline{x}^2) = \mathbf{A}_i(\overline{y}^2) = 1$$

$$\mathbf{A}_i(\overline{x}\,\overline{y}) = \mathbf{A}_i(\overline{y}\,\overline{x}) = 0$$

We then temporarily represent polynomials on $\mathbf{c}_i$ using the polynomial basis functions $\overline{P} = [1, \overline{x}, \overline{y}, \overline{x}^2, \overline{x}\overline{y}, \overline{y}^2, \overline{x}^3, \ldots]$. Note that polynomials in this system are easily transformed to the standard cell-centroid basis

$$\overline{x}^m \overline{y}^n = \sum_{s+t \le k} \binom{m}{s} \binom{n}{t} \mathbf{D}_{1,1}^s \mathbf{D}_{1,2}^{m-s} \mathbf{D}_{2,1}^t \mathbf{D}_{2,2}^{n-t} x^{s+t} y^{m+n-s-t}$$

Since $0 \le s+t \le k$ and $0 \le m+n-s-t \le k$, we can reorder and rewrite in terms of the standard and transformed basis polynomials

$$\overline{P}_{(m,n)} = \sum_{s+t \le k} G_{m,n}^{s,t} P_{(s,t)} \tag{3.3}$$

We can guarantee satisfaction of design criterion (1) (conservation of the mean) by introducing into the transformed coordinate system *zero mean* basis polynomials $\overline{P}^0$ in which all but the first have zero cell average, i.e $\overline{P}^0 = [1, \overline{x}, \overline{y}, \overline{x}^2 - 1, \overline{x}\,\overline{y}, \overline{y}^2 - 1, \overline{x}^3 - A_i(\overline{x}^3), \ldots]$. Note that using these polynomials requires a minor modification of (3.3) but retains the same form:

$$\overline{P}^0_{(m,n)} = \sum_{s+t \le k} \overline{G}_{m,n}^{s,t} P_{(s,t)} \tag{3.4}$$

Given this preparatory work, we are now ready to describe the formulation of the reconstruction algorithm.

Minimum Energy Reconstruction

We have observed that the set of cell neighbors $\mathcal{N}_i$ must contain at least $(k+1)(k+2)/2$ cells $\mathbf{c}_j$ if the reconstruction operator $\mathbf{R}_i^k$ is to be k-exact. That $(k+1)(k+2)/2$ cells is not sufficient in all situations is easily observed. If, for example, the cell-centers all lie on a conic section, then there is a nonzero polynomial u such that $\mathbf{A}_j(u) = 0$ for every cell $\mathbf{c}_j$, which means that reconstruction of u is impossible. In other cases a k-exact reconstruction operator $\mathbf{R}_i^k$ may exist, but due to the geometry may be poorly conditioned.

Our approach is to work with a slightly larger support containing more than the minimum number of cells. In this case the operator $\mathbf{R}_i^k$ is likely to be nonunique, because various subsets would be able to support reconstruction operators of degree k. Although all would reproduce a polynomial of degree k exactly, if we disregard roundoff, they would differ in their treatment of non-polynomials, or of polynomials of degree higher than k. Any k-exact reconstruction operator $\mathbf{R}_i^k$ is a weighted average of these basic ones. Our approach is to choose the one of minimum Frobenius norm. This operator is optimal, in a certain sense, when the function we are reconstructing is not exactly

a polynomial of degree k, but one that has been perturbed by the addition of Gaussian noise, for it minimises the expected deviation from the unperturbed polynomial in a certain rather natural norm.

As we begin the formulation of the reconstruction preprocessing algorithm, the reader is reminded that the task at hand is to calculate the weights $\mathbf{W}_i$ for each cell c_i which when applied via (3.2) produce piecewise polynomial approximations. We begin by first rewriting the piecewise polynomial (2.1) for cell c_i in terms of the reconstruction weights (3.2)

$$u^k(x,y) = \sum_{m+n\le k} P_{(m,n)} \sum_{j\in\mathcal{N}_{c_i}} W_{(m,n)j}\bar{u}_j \qquad (3.5a)$$

or equivalently

$$u^k(x,y) = \sum_{j\in\mathcal{N}_{c_i}} \bar{u}_j \sum_{m+n\le k} W_{(m,n)j} P_{(m,n)} \qquad (3.5b)$$

Polynomials of degree k or less are equivalently represented in the transformed coordinate system using zero mean polynomials

$$u^k(x,y) = \sum_{j\in\mathcal{N}_{c_i}} \bar{u}_j \sum_{m+n\le k} W'_{(m,n)j} \overline{P}^0_{(m,n)} \qquad (3.6)$$

Using (3.4) we can relate weights in the transformed system to weights in the original system

$$W_{(s,t)j} = \sum_{m+n\le k} \overline{G}^{s,t}_{m,n} W'_{(m,n),j} \qquad (3.7)$$

We satisfy k-exactness by requiring that (3.6) is satisfied for all linear combinations of $\overline{P}^0_{(s,t)}(x,y)$ such that $s+t\le k$. In particular, if $u^k(x,y) = \overline{P}^0_{(s,t)}(x,y)$ for some $s+t\le k$ then

$$\overline{P}^0_{(s,t)}(x,y) = \sum_{m+n\le k} \overline{P}^0_{(m,n)} \sum_{j\in\mathcal{N}_{c_i}} W'_{(m,n)j} A_i(\overline{P}^0_{(s,t)})$$

This is satisfied if for all $s+t, m+n \le k$

$$\sum_{j\in\mathcal{N}_{c_i}} W'_{(m,n)j} A_i(\overline{P}^0_{(s,t)}) = \delta^{st}_{mn}$$

Transforming basis polynomials back to the original coordinate system we have

$$\sum_{j\in\mathcal{N}_{c_i}} W'_{(m,n)j} \sum_{u+v\le k} \overline{G}^{u,v}_{s,t} A_i(P_{(u,v)}) = \delta^{st}_{mn} \qquad (3.8)$$

This can be locally rewritten in matrix form as

$$\mathbf{W}'_i \mathbf{A}'_i = \mathbf{I} \qquad (3.9a)$$

and transformed in terms of the standard basis weights via

$$\mathbf{W}_i = \mathbf{G}\mathbf{W}'_i \qquad (3.9b)$$

Note that $\mathbf{W}'_i$ is a $(k+1)(k+2)/2$ by $\mathcal{N}_i$ matrix and $\mathbf{A}'_i$ has dimensions $\mathcal{N}_i$ by $(k+1)(k+2)/2$. To solve (3.9a) in the optimum sense describe above, we perform an $\mathbf{L}_i\mathbf{Q}_i$ decomposition of $\mathbf{A}'_i$ where the orthogonal matrix $\mathbf{Q}_i$ and the lower triangular matrix $\mathbf{L}_i$ have been constructed using a modified Gram-Schmidt algorithm (or a sequence of Householder reflections) see ref. [22]. The weights $\mathbf{W}'_i$ are then given by

$$\mathbf{W}'_i = \mathbf{Q}^*_i \mathbf{L}_i^{-1}$$

Applying (3.7) we transform these weights to the standard centroid basis and the preprocessing step is complete.

We defer a precise error estimate of the reconstruction algorithm. Even so we can still mention some dominant effects to be addressed by a precise estimate. Because of k-exactness, the effect of mesh refinement results in a factor h^{k+1} in the error estimate. The bad news concerns fixed h and increasing k. Because those rows of $\mathbf{W}_i$ which compute highest order coefficients $\alpha_{(m,n)}$ are, in effect, order k difference operators, they must have coefficents of large magnitude and can produce large errors when applied to nonpolynomial data.

4. Numerical Flux Evaluation and Gaussian Quadratures.

Perhaps a dozen or more numerical fluxes have been devised for the Euler equations. (See the article by van Leer et. al. [18] for a critique of several of these.) One might infer that the large number of fluxes developed implies that the choice of numerical flux is of vital importance to solution accuracy. For smooth solutions and using the high order schemes described in this paper, this is not the case. To see this, consider the following argument. All numerical flux functions must have consistency, i.e. $\overline{\mathbf{F}}(\mathbf{u},\mathbf{u},\mathbf{n}) = \mathbf{F}(\mathbf{u},\mathbf{n})$. Furthermore, the k-exactness of the reconstructed piecewise polynomial insures that a measure of the reconstructed solution jump $[\mathbf{u}_+ - \mathbf{u}_-]$ diminishes with increasing polynomial order. In choosing a numerical flux, we require that the numerical flux approach the true Euler flux at this same rate. Beyond this we also require that the flux functions produce a dissipative and stable scheme. Many flux function satisfy these demands, in particular we have chosen Roe's flux [19] because of its simplicity:

$$\overline{\mathbf{F}}(\mathbf{u}_+,\mathbf{u}_-) = \frac{1}{2}(\mathbf{F}(\mathbf{u}_+,\mathbf{n}) + \mathbf{F}(\mathbf{u}_-,\mathbf{n})) - \frac{1}{2}|A(\mathbf{u}_+,\mathbf{u}_-,\mathbf{n})|(\mathbf{u}_+ - \mathbf{u}_-)$$

In this formula, $|A(u_+, u_-, n)|$ is a positive semidefinite matrix formed from the flux jacobian evaluated at the "Roe state", see [17] for details.

While the importance of numerical flux functions is reduced by higher order polynomial approximations, the choice of flux quadrature for evaluating

$$\oint_{\partial c_i} \overline{F}(u_+^k, u_-^k, n)\, dl$$

rises to a position of extreme importance. Of course this integration path ∂c_i is actually a collection of edges e_i of the mesh, i.e.

$$\sum_{e_i}^{deg(c_i)} \int_{e_i} \overline{F}(u_+^k, u_-^k, n)\, dl$$

and we need only consider integration on an edge. Two problems arise in this regard. The first difficulty involves interior edges. We seek approximate quadratures of sufficient order so as not to degrade the performance of the reconstruction while not being overly accurate (a waste of computer resources). The second difficulty involves boundary edges. We want the ability to perform a sequence of calculations about a geometry (with shape possibly described by transcendental functions) while increasing the order of polynomial order (k refinement) and/or refining the mesh (h refinement).

[illegible] converge to the exact solution unless we generalize our description of the boundary shape and allow for curved boundary edges in the overall scheme. Finite-element implementations typically resolve this problem by the use of isoparametric elements (elements which use the same interpolating function to describe the element shape as well as the field variable within the element).

We resolve both difficulties by using multi-point Gaussian quadrature formulas on both interior and boundary edges. Recall that N-point Gaussian quadrature on straight line intervals integrates polynomials of degree $2N-1$ exactly, and is $2N$ order accurate. On smooth curved boundary edges, Gauss points are distributed via arclength (see for example Ciarlet [20,21]) to yield consistent approximations. In the actual computer code, we reserve the right to use a different number of Guass quadrature points on boundary edges from that used on interior edges.

5. Numerical Validation of Reconstruction.

The accuracy of the reconstruction scheme is assessed in two numerical experiments involving the reconstruction of polynomial data and the Ringleb flow [15]. We first assess the accuracy of the reconstruction operator for polynomial data by generating a sequence of random meshes containing 89, 188, 381, and 777 cells, see Fig. 2 for a representative mesh.

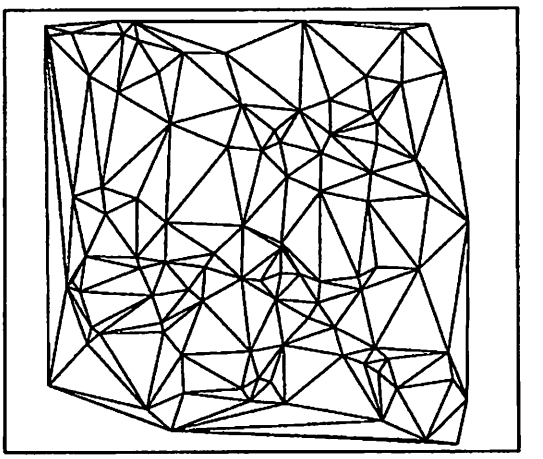

FIGURE 2. Random mesh (188 cells) for polynomial test function.

On each mesh a test polynomial (a sixth order polynomial with random normalized coefficients) is generated and cell-averaged onto the particular mesh. The data is then reconstructed and norms computed. Recall the standard norm measures, L_p

$$\|u - u^k\|_p = \left[\int (u - u^k)^p\, da\right]^{1/p}$$

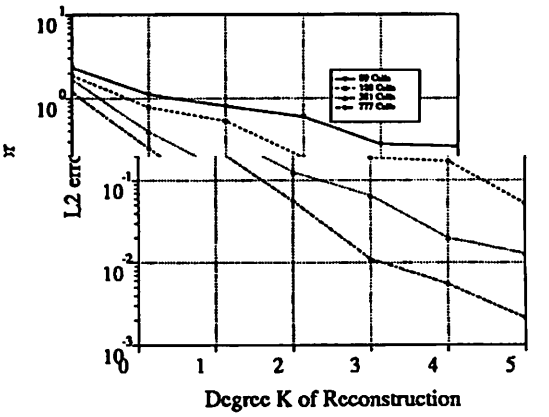

FIGURE 3. Reconstruction of Random Test Polynomial.

Figure 3 graphs the L_2 norm of the error in reconstruction for increasing k. The improvement with increased k and/or h is clear. The somewhat random behavior of the error is a result of the randomness of the mesh.

In the second numerical experiment we consider the reconstruction error obtained when reconstructing the cell-averaged form of the exact Ringleb flow density field for k and h refinement. A sequence of three random meshes were used for h refinement: coarse (210 cells), medium (506 cells) and fine (930 cells). The coarse mesh is shown in Fig. 4.

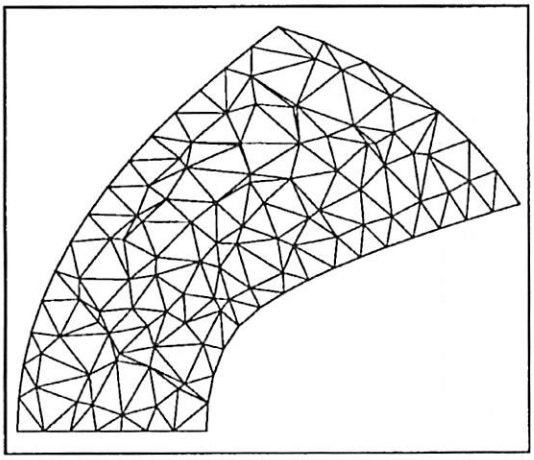

FIGURE 4. Coarse grid for Ringleb Flow.

In Fig. 5 we graph the L_2 error of the reconstructed density field. Again the reduction in error for k and h refinement is easily seen.

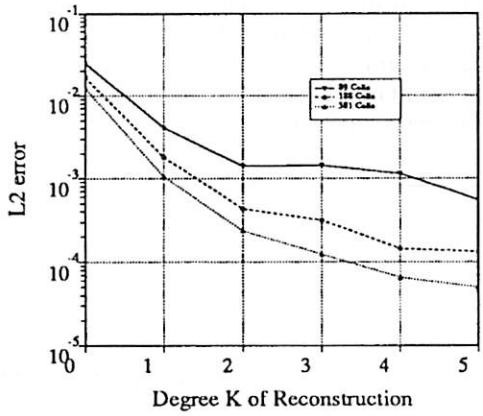

FIGURE 5. L_2 norm of error of reconstruction of density for Ringleb flow.

In Fig. 6A-B we show piecewise contouring of the density field on the coarse mesh for k=1 and 2. The improvement from linear to quadratic is very visible whereas improvement from quadratic to cubic (not shown) can only be visually detected in the corner region containing maximum solution gradients. Note that the use of piecewise plotting allows the observer a crude visual critique of accuracy. Clearly larger jumps in solution contours indicate poorer resolution by the approximating piecewise polynomials.

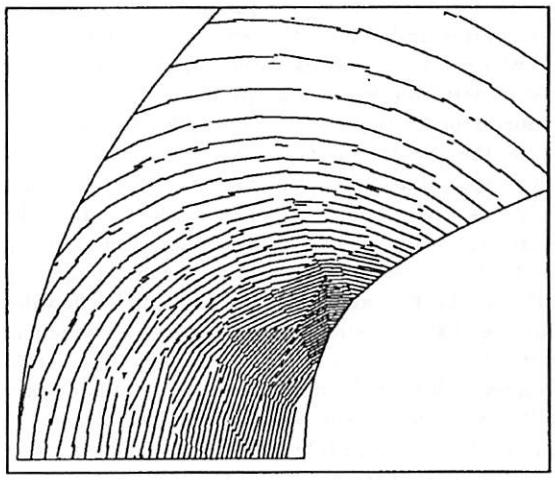

FIGURE 6A. Density contours for linear reconstruction of Ringleb flow (coarse mesh, k=1).

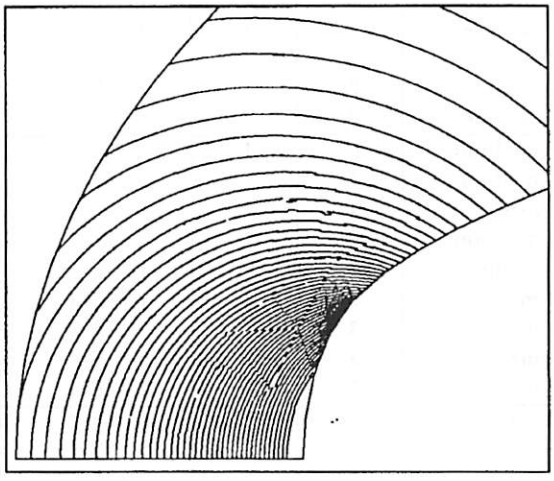

FIGURE 6B. Density contours for quadratic reconstruction of Ringleb flow (coarse mesh, k=2).

6. Numerical Validation of Solution Algorithm.

In this section use our flow solver to compute numerical solutions of the Ringleb flow. Unlike the previous study of section 5, we now have the added complexity of choosing the order N of the Gauss quadrature rule for flux evaluations (as discussed in section 4). To better understand this effect tabulated results are first presented and discussed. Once this effect is better understood, we then graph the norm of absolute solution error for sequences of k and h refinement.

In the present study we are particularly interested in spatial accuracy and have therefore chosen a steady flow with an analytic solution. In obtaining the numerical solution we employ a variable timestepping explict scheme. Although we impose exact boundary conditions at inflow and outflow of the Ringleb flow domain, the characteristic nature of the Roe numerical flux function effectively sorts out ingoing and outgoing waves such that the number and type of boundary conditions are consistent with hyperbolic Euler flow.

In table 1 we give tabulated results of L_2 solution error using the same three meshes of the previous section, k-exact reconstruction ($0 \le k \le 3$), and various orders of Gauss quadrature N.

Mesh	k	N	L_2 norm
coarse	0	1	9.66E-2
coarse	1	1	8.88E-3
coarse	1	2	7.59E-3
coarse	1	3	7.60E-3
coarse	2	1	7.73E-3
coarse	2	2	4.05E-3
coarse	2	3	4.03E-3
coarse	3	1	6.80E-3
coarse	3	2	2.73E-3
coarse	3	3	2.72E-3
medium	0	1	7.31E-2
medium	1	1	4.37E-3
medium	1	2	4.21E-3
medium	2	1	2.54E-3
medium	2	2	1.78E-3
medium	2	3	1.78E-3
medium	3	1	2.42E-3
medium	3	2	9.45E-4
medium	3	3	9.22E-4
fine	0	1	5.62E-2
fine	1	1	2.43E-3
fine	1	2	2.49E-3
fine	2	1	1.23E-3
fine	2	2	8.22E-4
fine	2	3	8.22E-4
fine	3	1	1.61E-3
fine	3	2	5.46E-4
fine	3	3	5.50E-4

Table 1. Flow solver accuracy for Ringleb Flow

From this table we observe a consistent loss of accuracy if the order N of the Gauss quadrature flux evaluation is less than $(k+1)/2$. Moreover we do not find a significant advantage in using N greater than $\lceil (k+1)/2 \rceil$ (the least integer greater than or equal to $(k+1)/2$). In Fig. 7 we graph the absolute error of the computed Ringleb flow for various h and k with $N = \lceil (k+1)/2 \rceil$.

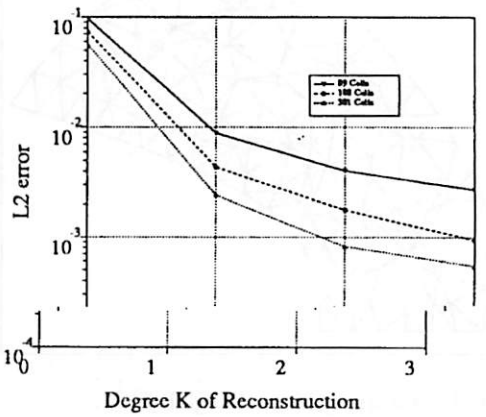

FIGURE 7. Accuracy of flow solver for Ringleb flow.

In Fig8a we show density contours of the Ringleb flow computed using quadratic reconstruction and two point Gauss quadrature of numerical fluxes. The loss of solution accuracy resulting from inadequate Gauss quadrature is graphically illustrated in Fig8b which shows the result of using quadratic reconstruction with one point Gauss quadrature.

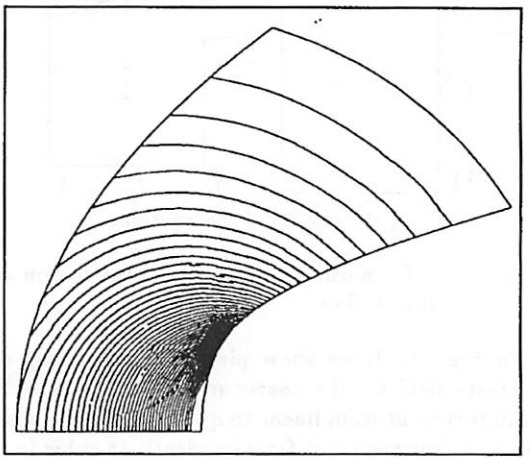

FIGURE 8A. Density contours for numerical solution of Ringleb flow (coarse mesh, k=2, N=2)

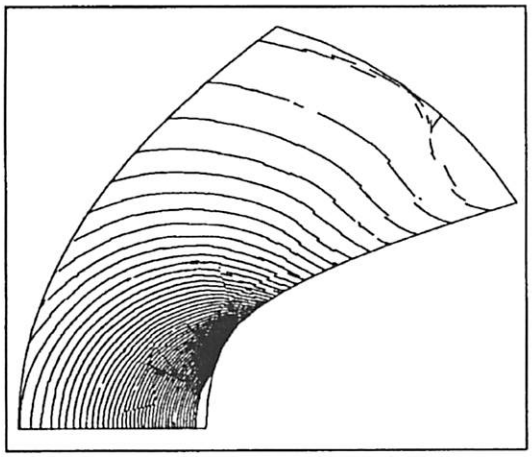

FIGURE 8B. Density contours for numerical solution of Ringleb flow (coarse mesh, k=2, N=1)

8. Concluding Remarks.

We have constructed and tested a class of higher order accurate finite-volume schemes for the Euler equations of gasdynamics for arbitrary unstructured meshes. The development of these flow solvers necessitated the development of high order (k-exact) reconstruction and the use of high order Gaussian quadratures in the flux evaluations. Numerical experiments have verified that Gauss quadratures of order $N \geq (k+1)/2$ are essential to obtain the full accuracy of the k-exact reconstruction.

9. References.

1. Godunov, S. K., "A Finite Difference Method for the Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics", Mat. Sb., Vol. 47, 1959.

2. Van Leer, B., "Towards the Ultimate Conservative Difference Schemes V. A Second Order Sequel to Godunov's Method", J. Comp. Phys., Vol. 32, 1979.

3. Colella, P., Woodward, P., "The Piecewise Parabolic Method for Gas-Dynamical Simulations", J. Comp. Phys., Vol. 54, 1984.

4. Woodward, P., Colella, P., "The Numerical Simulation of Two-Dimensioal Fluid Flow with Strong Shocks" J. Comp. Phys., Vol. 54, 1984.

5. Harten, A. , Osher, S., "Uniformly High-Order Accurate Non-oscillatory Schemes, I.," MRC Technical Summary Report 2823, 1985.

6. Harten, A., Engquist, B., Osher, S.,Chakravarthy, "Uniformly High Order Accurate Essentially Non - Oscillatory Schemes III, ICASE report 86-22, 1986.

7. Jameson, A. and Mavriplis, D., "Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh", AIAA paper 85-0435, January 1985.

8. Mavriplis, D., "Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes", ICASE Report No. 87-53.

9. Mavriplis, D. and Jameson, A., "Multigrid Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes', AIAA paper 87-0353, January 1987.

10. Morgan K., Peraire J.,, "Finite Elements for Compressible Flows", VKI Lecture Notes, 1988.

11. Lohner, R., Morgan K., and Peraire J., "Finite Elements for Compressible Flow", Numerical Methods for Fluid Dynamics (K.W. Morton and M.J. Baines eds.) Oxford University Press, 1986.

12. Barth, T. J., and Jespersen, D. C., "The Design and Application of Upwind Scehemes on Unstructured Meshes", AIAA-89-0366, Jan. 9-12, 1989.

13. Rostand, P. and Stoufflet, B., "A Numerical Scheme for Computing Hypersonic Viscous Flows on Unstructured Meshes", Second International Conference on Hyperbolic Problems, Aachen, W. Germany, 1988.

14. Desideri, J. A., and Dervieux, A., "Compressible Flow Solvers Using Unstructured Grids", VKI Lecture Series 1988-05, March 7-11, 1988, pp. 1—115.

15. Chiocchia, G., " Exact Solutions to Transonic and Supersonic Flows", AGARD Advisory Report AR-211,1985.

16. Van Leer, B., " Upwind-Difference Methods for Aerodynamic Problems Governed by the Euler Equations", Lecture Notes in Applied Mathematics, Vol. 22, 1985.

17. Thomas, J. L., van Leer, B., and Walters, R. W., "Implicit Flux-Split Schemes for the Euler Equations", AIAA paper 85-1680, July 1985.

18. Van Leer, B., Thomas, J. L., Roe, P. L., Newsome, R. W., "A Comparison of Numerical Flux Formulas for the Euler and Navier-Stokes Equations", AIAA Paper 87-1104, AIAA 8th CFD Conference, Honolulu, Hawaii, 1987.

19. Roe, P.L.,"Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes", J. Comput. Phys., Vol 43, 1981.

20. Ciarlet, P.G., **The Finite Element Method for Elliptic Problems**, (Ed. Lions, Papanicolaou, and Rockafellar), North Holland/Elsevier, 1978.

21. Ciarlet, P.G., Raviart, P.-A., "The Combined Effect of Curved Boundaries and Numerical Integration in Isoparametric Finite Element Methods", **The Mathematical Foundations of the Finite Element Method with Application to Partial Differential Equations**, (Ed., A.K. Aziz), Academic Press, New York, 1972, pp. 409-474.

22. Börk, Å, "Least Squares Methods", To appear in **Handbook of Numerical Analysis**, Vol. 1: Solutions of Eqns. in R^n, (Ed. Ciarlet, Lions), Elsevier.