Since $\alpha_{0i} < \pi$, $\beta_{0i} < \pi$, the denominator is always positive and nonnegativity requires that $\alpha_{0i} + \beta_{0i} \leq \pi$.

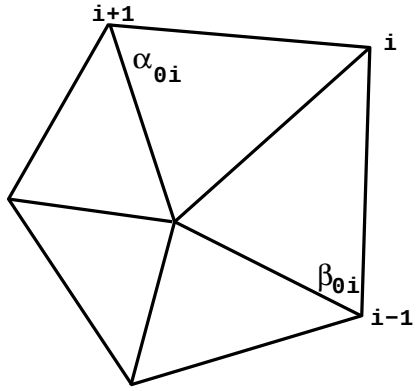

Figure 4.1.2 Circumcircle test for adjacent triangles.

Some trigonometry reveals that for the configuration of figure 4.1.2 with circumcircle passing through $\{v_0, v_i, v_{i+1}\}$ the sum $\alpha_{0i} + \beta_{0i}$ depends on the location of v_{i-1} with respect to the circumcircle in the following way:

$$\begin{aligned} &\alpha_{0i} + \beta_{0i} < \pi, \quad v_{i-1} \text{ exterior} \\ &\alpha_{0i} + \beta_{0i} > \pi, \quad v_{i-1} \text{ interior} \\ &\alpha_{0i} + \beta_{0i} = \pi, \quad v_{i-1} \text{ cocircular} \end{aligned} \tag{4.1.14}$$

Also note that we could have considered the circumcircle passing through $\{v_0, v_i, v_{i-1}\}$ with similar results for v_{i+1}. The condition of nonnegativity implies a circumcircle condition for all pairs of adjacent triangles whereby the circumcircle passing through either triangle cannot contain the fourth point. This is precisely the *unique* characterization of the Delaunay triangulation which completes the proof.

■

Keep in mind that from equation (4.1.13) we have that $\text{cotan}(\alpha) \geq 0$ and $\text{cotan}(\beta) \geq 0$ if $\alpha \leq \pi/2$. Therefore a sufficient but not necessary condition for nonnegativity of the Laplacian weights is that all angles of the mesh be less than or equal to $\pi/2$. This is a standard result in finite element theory [Ciarlet73] and applies in two or more space dimensions.

From Lemma 4.1.1 we have the following theorem concerning a discrete maximum principle.
Theorem 4.1.2: The discrete Laplacian operator obtained from the finite element discretization (4.1.11) with linear elements exhibits a discrete maximum principle for arbitrary point sets in two space dimensions if the triangulation of these points is a Delaunay triangulation.

Elements of the Proof: From Lemma 4.1.1 a one-to-one correspondence exists between nonnegativity of weights and Delaunay triangulation. Assume a Delaunay triangulation of the point set so that for some arbitrary interior vertex v_0 we have all $W_{0i} \geq 0$ and solve for U_0:

$$U_0 = \frac{\sum_{i \in \mathcal{N}_0} W_{0i} U_i}{\sum_{i \in \mathcal{N}_0} W_{0i}} = \sum_{i \in \mathcal{N}_0} \sigma_i U_i$$

with

$$\sigma_i = \frac{W_{0i}}{\sum_{i \in \mathcal{N}_0} W_{0i}}$$

which satisfies $\sigma_i \geq 0$ and $\sum_{i \in \mathcal{N}_0} \sigma_i = 1$. Since U_0 is a convex combination of the neighboring values we have

$$\min_{i \in \mathcal{N}_0} U_i \leq U_0 \leq \max_{i \in \mathcal{N}_0} U_i \tag{4.1.15}$$

If U_0 attains a maximum value M then all $U_i = M$. Repeated application of (4.1.15) to neighboring vertices in the triangulation establishes the discrete maximum principle.

■

We can ask if the result concerning Delaunay triangulation and the maximum principle extends to three space dimensions. As we will show, the answer is no. The resulting formula for the three dimensional Laplacian is

$$\int_{\Omega_0} w \, \Delta u \; d\, v = \sum_{i \in \mathcal{N}_0} W_{0i}(U_i - U_0) \tag{4.1.16}$$

where

$$W_{0i} = \frac{1}{6} \sum_{k=1}^{d(v_0, v_i)} |\Delta \mathbf{R}_{k+\frac{1}{2}}| \, \mathrm{cotan}(\alpha_{k+\frac{1}{2}}). \tag{4.1.17}$$

In this formula Ω_0 is the volume formed by the union of all tetrahedra that share vertex v_0. $\mathcal{N}_0$ is the set of indices of all adjacent neighbors of v_0 connected by incident edges, k a local cyclic index describing the associated vertices which form a polygon of degree $d(v_0, v_i)$ surrounding the edge $e(v_0, v_i)$, $\alpha_{k+\frac{1}{2}}$ is the face angle between the two faces associated with $\vec{\mathbf{S}}_{k+\frac{1}{2}}$ and $\vec{\mathbf{S}}'_{k+\frac{1}{2}}$ which share the edge $e(v_k, v_{k+1})$ and $|\Delta \mathbf{R}_{k+\frac{1}{2}}|$ is the magnitude of the edge (see figure below).

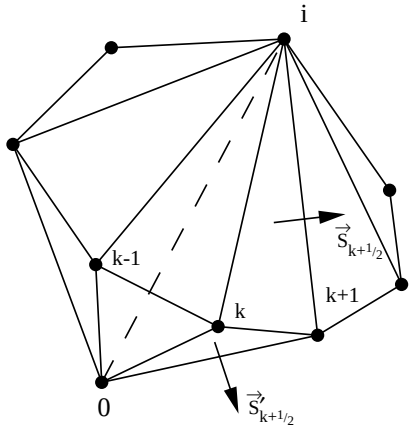

Figure 4.1.3 Set of tetrahedra sharing interior edge $e(v_0, v_i)$ with local cyclic index k.

A maximum principle is guaranteed if all $W_{0i} \geq 0$. We now will procede to describe a valid Delaunay triangulation with one or more $W_{0i} < 0$. It will suffice to consider the Delaunay triangulation of N points in which a single point v_0 lies interior to the triangulation and the remaining $N-1$ points describe vertices of boundary faces which completely cover the convex hull of the point set.

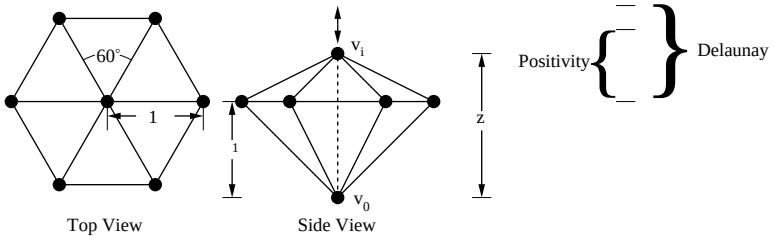

Figure 4.1.4 Subset of 3-D Delaunay Triangulation which does not maintain nonnegativity.

Consider a subset of the N vertices, in particular consider an interior edge incident to v_0 connecting to v_i as shown in figure 4.1.4 by the dashed line segment and all neighbors adjacent to v_i on the hull of the point set. In this experiment we consider the height of the interior edge, z, as a free parameter. Although it will not be proven here, the remaining $N-8$ points can be placed without conflicting with any of the conclusions obtained for looking at the subset.

It is known that a necessary and sufficient condition for the 3-D Delaunay triangulation is that the circumsphere passing through the vertices of any tetrahedron must be point free [Law86]; that is to say that no other point of the triangulation can lie interior to this sphere. Furthermore a property of locality exists so that we need only inspect adjacent tetrahedra for the satisfaction of the circumsphere test. For the configuration of points shown in figure 4.1.4, convexity of the point cloud constrains $z \geq 1$ and the satisfaction of the circumsphere test requires that $z \leq 2$.

$$1 \leq z \leq 2 \quad \text{(Delaunay Triangulation)}$$

From (4.1.17) we find that $W_{0i} \geq 0$ if and only if $z < 7/4$.

$$1 \leq z \leq \frac{7}{4}, \quad \text{(Nonnegativity)}$$

This indicates that for $7/4 < z \leq 2$ we have a valid Delaunay triangulation which does not satisfy a discrete maximum principle. In fact, the Delaunay triangulation of 400 points randomly distributed in the unit cube revealed that approximately 25% of the interior edge weights were of the wrong sign (negative).

Keep in mind that from (4.1.17) we can obtain a sufficient but not necessary condition for nonnegativity that all face angles be less than or equal to $\pi/2$. This is consistent with the known result from [Ciarlet73].

4.2 Discrete Variation and Maximum Principles for Hyperbolic Equations

In this section we examine discrete total variation and maximum principles for scalar conservation law equations. We first consider the nonlinear conservation law equation:

$$u_t + (f(u))_x = 0, u(x,0) = u_0(x) \tag{4.2.0}$$

which is discretized in the conservation form:

$$\begin{aligned} U_j^{n+1} =& U_j^n - \frac{\Delta t}{\Delta x}(h_{j+\frac{1}{2}} - h_{j-\frac{1}{2}}) \\ =& H(U_{j-l}^n, U_{j-l+1}^n, ..., U_{j+l}) \end{aligned} \tag{4.2.1}$$

where $h_{j+\frac{1}{2}} = h(U_{j-l+1}, ..., U_{j+l})$ is the numerical flux function satisfying the consistency condition

$$h(U, U, ..., U) = f(U).$$

A finite-difference scheme (4.2.1) is said to be *monotone* in the sense of Harten, Hyman, and Lax [HartHL76] if H is a monotone increasing function of each of its arguments.

$$\frac{\partial H}{\partial U_i}(U_{-k}, ..., U_k) \geq 0 \quad \forall \quad -k \leq i \leq k \qquad \text{(HHL monotonicity)}$$

This is a strong definition of monotonicity. In [HartHL76] it is proven that schemes satisfying this condition also satisfy the entropy inequality which distinguishes physically relevant discontinuities. Unfortunately, they also prove that HHL monotone schemes in conservation form are at most first order spatially accurate.

To allow higher order accuracy, Harten introduced a weaker concept of monotonicity. A grid function U is called monotone if for all i

$$\min(U_{i-1}, U_{i+1}) \leq U_i \leq \max(U_{i-1}, U_{i+1}). \tag{4.2.2}$$

A scheme is called monotonicity preserving if monotonicity of U^{n+1} follows from monotonicity of U^n. Observe the close relationship between monotonicity preservation in time and the discrete maximum principle for Laplace's equation (4.1.15) in space. It follows immediately from the definition of monotonicity preservation that

(1) Local maxima are nonincreasing
(2) Local minima are nondecreasing

which is a property of the conservation law equation (discussed earlier). Using this weaker form of monotonicity Harten [Hart83] introduced the notion of total variation diminishing schemes. Define the total variation in one dimension:

$$TV(U) = \sum_{-\infty}^{\infty} |U_i - U_{i-1}|. \tag{4.2.3}$$

A scheme is said to be total variation diminishing (TVD) if

$$TV(U^{n+1}) \leq TV(U^n) \tag{4.2.4}$$

This is a discrete analog of the total variation statement (4.0.4) given for the conservation law equation. Harten has proven that schemes which are HHL monotone are TVD and schemes that are TVD are monotonicity preserving. Furthermore, it can be shown that all *linear* monotonicity preserving schemes are at most first order accurate. Thus high order accurate TVD schemes must necessarily be nonlinear in a fundamental way.

To understand the basic design principles for TVD schemes, assume a one-dimensional periodic grid together with the following numerical scheme in abstract matrix operator form

$$[I + \theta\Delta t\ \widetilde{M}\, D]U^{n+1} = [I - (1-\theta)\Delta t\ M\, D]U^n \tag{4.2.5}$$

where $\widetilde{M}$ and M are matrices which can be nonlinear functions of the solution U. The matrix D denotes the difference operator

$$D\,U = [I - E^{-1}]U = \begin{pmatrix} U_1 - U_J \\ U_2 - U_1 \\ U_3 - U_2 \\ \vdots \\ U_J - U_{J-1} \end{pmatrix}.$$

The scheme (4.2.5) represents a general family of explicit ($\theta = 0$) and implicit ($\theta = 1$) schemes with arbitrary support. More importantly we claim that schemes written in *conservative* form can be rewritten in this form using (exact) mean value constructions. Using this notation, we have a equivalent definition of the total variation in terms of the L_1 norm:

$$TV(U) = \|D\,U\|_1.$$

To analyze the scheme (4.2.5), multiply by D from the left and regroup.

$$[I + \theta\Delta t\ D\,\widetilde{M}]D\,U^{n+1} = [I - (1-\theta)\Delta t\ D\,M]D\,U^n \tag{4.2.6}$$

or in symbolic form

$$\mathcal{L}D\,U^{n+1} = \mathcal{R}D\,U^n \quad D\,U^{n+1} = \mathcal{L}^{-1}\mathcal{R}D\,U^n. \tag{4.2.7}$$

where we have assumed invertibility of $\mathcal{L}$. This invertibility will be guaranteed from the diagonal dominance required below. Next we take the L_1 norm of both sides and apply matrix-vector inequalities.

$$\begin{aligned} TV(U^{n+1}) = \|D\,U^{n+1}\|_1 &\leq \|\mathcal{L}^{-1}\mathcal{R}\|_1 \|D\,U^n\|_1 \\ &= \|\mathcal{L}^{-1}\mathcal{R}\|_1 TV(U^n) \end{aligned}$$

Thus we see that a sufficient condition is that $\|\mathcal{L}^{-1}\mathcal{R}\|_1 \leq 1$. Recall that the L_1 norm of a matrix is obtained by summing the absolute value of elements of columns of the matrix

and choosing the column whose sum is largest. Furthermore, we have the usual matrix inequality $\|\mathcal{L}^{-1}\mathcal{R}\|_1 \leq \|\mathcal{L}^{-1}\|_1 \|\mathcal{R}\|_1$ so that sufficient TVD conditions are that $\|\mathcal{L}^{-1}\|_1 \leq 1$ and $\|\mathcal{R}\|_1 \leq 1$. As we will see, these simple inequalities are enough to recover the TVD criteria of previous investigators, see [Hart83], [Jam86].

Theorem 4.2.0:[BarL87] A sufficient condition for the scheme (4.2.5) to be TVD with $\theta = 0$ is that $\mathcal{R}$ be bounded with $\mathcal{R} \geq 0$ (all elements are nonnegative).

Proof: Consider the explicit operator $\mathcal{R} = I - \Delta t\, D\, M$ and multiply it from the left by the summation vector $s^T = [1, 1, ..., 1]$. It is clear that $s^T D = 0$ so that $s^T \mathcal{R} = s^T$ (columns sum to unity). Because the L_1 norm of $\mathcal{R}$ is the maximum of the sum of absolute values of elements in columns of $\mathcal{R}$, we immediately obtain the desired result.

■

Next we consider the implicit scheme with $\theta = 1$ and sufficient conditions for $\|\mathcal{L}^{-1}\|_1 \leq 1$. From the previous development, one way to do this would be to show that $\mathcal{L}$ is monotone, i.e. $\mathcal{L}^{-1} \geq 0$ with columns that sum to unity.

Theorem 4.2.1:[BarL87] A sufficient condition for the scheme (4.2.5) to be TVD with $\theta = 1$ is that $\mathcal{L}$ be an M-type monotone matrix, i.e. diagonally dominant with positive diagonal entries and negative off-diagonal entries.

Proof: Consider the implicit operator $\mathcal{L} = I + \Delta t\, D\, \widetilde{M}$ and multiply it from the left by the summation vector. Again we have that $s^T D = 0$ so that

$$s^T \mathcal{L} = s^T \rightarrow s^T = s^T \mathcal{L}^{-1}$$

which implies that columns of $\mathcal{L}^{-1}$ sum to unity. Finally, we appeal to lemma 4.1.0 concerning M-type monotone matrix operators to obtain the final result.

■

Next we will demonstrate that this general theory reproduces some well known results by Harten [Hart83]. Consider the following explicit scheme in Harten's notation:

$$U_j^{n+1} = U_j^n + C^+_{j+1/2} \Delta_{j+1/2} U^n - C^-_{j-1/2} \Delta_{j-1/2} U^n$$

where $\Delta_{j+1/2} U = U_{j+1} - U_j$. The operator $\mathcal{R}$ in this case has the following banded structure

$$\begin{pmatrix} \ddots & \ddots & 0 & 0 & \ddots \\ \ddots & \ddots & C^+_{j+1/2} & 0 & 0 \\ 0 & C^-_{j-1/2} & 1 - C^+_{j+1/2} - C^-_{j+1/2} & C^+_{j+3/2} & 0 \\ 0 & 0 & C^-_{j+1/2} & \ddots & \ddots \\ \ddots & 0 & 0 & \ddots & \ddots \end{pmatrix}$$

We need only require that this matrix be nonnegative to arrive at Harten's criteria:

$$C^+_{j+1/2} \geq 0$$

$$C^-_{j+1/2} \geq 0$$

$$1 - C^+_{j+1/2} - C^-_{j+1/2} \geq 0$$

Next we consider Harten's implicit form:

$$U^{n+1}_j + D^+_{j+1/2}\Delta_{j+1/2}U^{n+1} - D^-_{j-1/2}\Delta_{j-1/2}U^{n+1} = U^n_j$$

In this case $\mathcal{L}$ has the general structure

$$\begin{pmatrix} \ddots & \ddots & 0 & 0 & \ddots \\ \ddots & \ddots & -D^+_{j+1/2} & 0 & 0 \\ 0 & -D^-_{j-1/2} & 1+ D^+_{j+1/2}+D^-_{j+1/2} & -D^+_{j+3/2} & 0 \\ 0 & 0 & -D^-_{j+1/2} & \ddots & \ddots \\ \ddots & 0 & 0 & \ddots & \ddots \end{pmatrix}$$

To obtain Harten's TVD criteria for the implicit scheme, we need only require that this operator be an M-matrix to obtain the following conditions as did Harten

$$D^+_{j+1/2} \geq 0$$

$$D^-_{j+1/2} \geq 0$$

Maximum Principles and Monotonicity Preserving Schemes on Multidimensional Structured Meshes

Unfortunately, we have two motivations for further weakening the concept of monotonicity. The first motivation concerns a negative result by Goodman and Le Veque [GoodLV85] in which they show that conservative TVD schemes on Cartesian meshes in two space dimensions are first order accurate. The second motivation is the apparent difficulty in extending the TVD concept to arbitrary unstructured meshes. The first motivation inspired Spekreijse [Spek87] to consider a new class of monotonicity preserving schemes based on positivity of coefficients. Consider the following conservation law equation in two space dimensions

$$u_t + (f(u))_x + (g(u))_y = 0. \tag{4.2.8}$$

Next construct a discretization of (4.2.8) on a logically rectangular mesh

$$\begin{aligned} \frac{U^{n+1}_{j,k} - U^n_{j,k}}{\Delta t} =& A^+_{j+\frac{1}{2},k}(U^n_{j+1,k} - U^n_{j,k}) + A^-_{j-\frac{1}{2},k}(U^n_{j-1,k} - U^n_{j,k}) \\ &+ B^+_{j,k+\frac{1}{2}}(U^n_{j,k+1} - U^n_{j,k}) + B^-_{j,k-\frac{1}{2}}(U^n_{j,k-1} - U^n_{j,k}) \end{aligned} \tag{4.2.9}$$

with *nonlinear* coefficients

$$A^{\pm}_{j+\frac{1}{2},k} = A(..., U^n_{j-1,k}, U^n_{j,k}, U^n_{j+1,k}, ...)$$

$$B^{\pm}_{j-\frac{1}{2},k} = B(..., U^n_{j-1,k}, U^n_{j,k}, U^n_{j+1,k}, ...)$$

Theorem 4.1.2: The scheme (4.2.9) exhibits a discrete maximum principle at steady-state if all coefficients are uniformly bounded and nonnegative

$$A^{\pm}_{j\pm\frac{1}{2},k} \geq 0 \quad B^{\pm}_{j\pm\frac{1}{2},k} \geq 0.$$

Furthermore, the scheme (4.2.9) is monotonicity preserving under a CFL-like condition if

$$\Delta t \leq \frac{1}{\sum_{\pm}(A^{\pm}_{j\pm\frac{1}{2},k} + B^{\pm}_{j,k\pm\frac{1}{2}})}.$$

Proof: We first prove a discrete maximum principle at steady-state by solving for the value at (j,k).

$$\begin{aligned} U_{j,k} =& \frac{\sum_{\pm}(A_{j\pm\frac{1}{2},k}U_{j\pm1,k} + B_{j,k\pm\frac{1}{2}}U_{j,k\pm1})}{\sum_{\pm}(A_{j\pm\frac{1}{2},k} + B_{j,k\pm\frac{1}{2}})} \\ =& \sum_{\pm}(\alpha_{j\pm\frac{1}{2},k}U_{j\pm1,k} + \beta_{j,k\pm\frac{1}{2}}U_{j,k\pm1}) \end{aligned} \tag{4.2.10}$$

with the constraints

$$\alpha_{j-\frac{1}{2},k} + \alpha_{j+\frac{1}{2},k} + \beta_{j,k-\frac{1}{2}} + \beta_{j,k+\frac{1}{2},k} = 1$$

and $\alpha_{j\pm\frac{1}{2},k} \geq 0$, and $\beta_{j,k\pm\frac{1}{2}} \geq 0$. From positivity of coefficients and convexity of (4.2.10) we have

$$\min(U_{j\pm1,k}, U_{j,k\pm1}) \leq U_{j,k} \leq \max(U_{j\pm1,k}, U_{j,k\pm1}). \tag{4.2.11}$$

If $U_{j,k}$ attains a maximum value M at (j,k) then

$$M = U_{j-1,k} = U_{j+1,k} = U_{j,k-1} = U_{j,k+1}.$$

Repeated application of (4.2.11) to neighboring mesh points establishes the maximum principle.

Next we determine a CFL-like condition for monotonicity preservation in time by again seeking positivity of coefficients and a convex local mapping from U^n to U^{n+1}.

$$\begin{aligned} U^{n+1}_{j,k} &= \left(1 - \Delta t \sum_{\pm}(A^{\pm}_{j\pm\frac{1}{2},k} + B^{\pm}_{j,k\pm\frac{1}{2}})\right) U^n_{j,k} + \Delta t \sum_{\pm}(A_{j\pm\frac{1}{2},k}U^n_{j\pm1,k} + B_{j,k\pm\frac{1}{2}}U^n_{j,k\pm1}) \\ &= \gamma_{j,k}U^n_{j,k} + \sum_{\pm}(\alpha_{j\pm\frac{1}{2},k}U^n_{j\pm1,k} + \beta_{j,k\pm\frac{1}{2}}U^n_{j,k\pm1}) \end{aligned} \tag{4.2.12}$$

with the derivable constraints

$$\gamma_{j,k} + \alpha_{j-\frac{1}{2},k} + \alpha_{j+\frac{1}{2},k} + \beta_{j,k-\frac{1}{2}} + \beta_{j,k+\frac{1}{2},k} = 1$$

and $\alpha_{j\pm\frac{1}{2},k} \geq 0$, and $\beta_{j,k\pm\frac{1}{2}} \geq 0$. To show that (4.2.12) is a local convex mapping we need only satisfy the CFL-like condition for nonnegativity of $\gamma_{j,k}$:

$$\Delta t \leq \frac{1}{\sum_{\pm}(A^{\pm}_{j\pm\frac{1}{2},k} + B^{\pm}_{j,k\pm\frac{1}{2}}).} \tag{4.2.13}$$

If (4.2.13) is satisfied then monotonicity preservation in time follows immediately:

$$\min(U^n_{j\pm1,k}, U^n_{j,k\pm1}, U^n_{j,k}) \leq U^{n+1}_{j,k} \leq \max(U^n_{j\pm1,k}, U^n_{j,k\pm1}, U^n_{j,k}) \tag{4.2.14}$$

■

Maximum Principles and Monotonicity Preserving Schemes on Unstructured Meshes

In this section examine the maximum principle theory for unstructured meshes. We restrict our attention to the analysis of Godunov-like schemes based on solution reconstruction and evolution. Thus the present analysis differs from maximum principle theory based on the "upwind triangle" scheme developed by Desideri and Dervieux [DesD88], Arminjon and Dervieux [ArmD89] and Jameson [Jam93].

Consider an integral form of (4.2.5) for some domain Ω comprised of nonoverlapping control volumes, Ω_i, such that $\Omega = \cup\Omega_i$ and $\Omega_i \cap \Omega_j = 0, i \neq j$. In each control volume we have

$$\frac{\partial}{\partial t}\int_{\Omega_i} u \; d\,\Omega + \int_{\partial\Omega_i} (F \cdot \mathbf{n}) \; d\,\Gamma = 0. \tag{4.2.15}$$

where $F(u) = f(u)\hat{i} + g(u)\hat{j}$. The situation is depicted for a control volume Ω_0 in fig. 4.2.0.

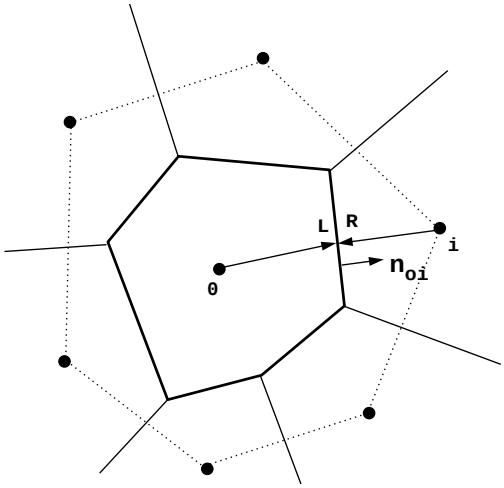

Figure 4.2.0 Local control volum configuration for unstructured mesh.

For two and three-dimensional triangulations, several control volume choices are available: the triangles themselves, Voronoi duals, median duals, etc. Although the actual choice of control volume tessellation is very important, the monotonicity analysis contained in the remainder of this section is largely independent of this choice. Therefore we assume a generic control volume Ω_0 with neighboring control volumes $\Omega_i,\ i \in \mathcal{N}_i$ as shown in fig. 4.2.0. In later sections we will examine the particular choice of control volume in detail.

Example: Analysis of an Upwind Advection Scheme with Piecewise Constant Reconstruction.

In this example we assume that the solution data $u(x,y)_i$ in each control volume Ω_i is constant with value equal to the integral average value, i.e.

$$u(x,y)_i = \overline{u}_i = \frac{1}{A_i}\int_{\Omega_i} u \; d\,\Omega, \;\; \forall \Omega_i \in \Omega. \tag{4.2.16}$$

where A_i is the area of Ω_i.. Next we define the unit exterior normal vector $\mathbf{n}_{0i}$ for the control volume boundary separating Ω_0 and Ω_i. It is also useful to define a variant of the unit normal vector $\vec{\mathbf{n}}_{0i}$ which is scaled by the length (area in 3-D) of that portion of the control volume boundary separating Ω_0 and Ω_i. Finally, to simplify the exposition we define

$$f(u;\vec{\mathbf{n}}) = F(u)\cdot\vec{\mathbf{n}}$$

and assume the existence of a mean value linearization such that

$$f(v;\vec{\mathbf{n}}) - f(u;\vec{\mathbf{n}}) = df(u,v;\vec{\mathbf{n}})(v-u). \tag{4.2.17}$$

Using this notation, we construct the following upwind scheme

$$\frac{d}{dt}(A_0\overline{u}_0) = -\sum_{i\in\mathcal{N}_0} h(\overline{u}_0,\overline{u}_i;\vec{\mathbf{n}}_{0i}) \tag{4.2.18}$$

with

$$h(\overline{u}_0,\overline{u}_i;\vec{\mathbf{n}}_{0i}) = \frac{1}{2}\left(f(\overline{u}_0;\vec{\mathbf{n}}_{0i}) + f(\overline{u},\vec{\mathbf{n}}_{0i})\right) - \frac{1}{2}|df(\overline{u}_0,\overline{u}_i;\vec{\mathbf{n}}_{0i})|(\overline{u}_i - \overline{u}_0) \tag{4.2.19}$$

In Barth and Jespersen [BarJ89] we proved a maximum principle and monotonicity preservation of the scheme (4.2.18) for scalar advection.

Theorem 4.2.2 The upwind algorithm with piecewise constant solution data (4.2.18) exhibits a discrete maximum principle for arbitrary unstructured meshes and is monotonicity preserving under the CFL-like condition:

$$\Delta t \le \frac{-A_j}{\sum_{i\in\mathcal{N}_j} df^-(\overline{u}_i,\overline{u}_j;\vec{\mathbf{n}}_{ji})}, \quad \forall \Omega_j \in \Omega$$

Proof: For simplicity consider the control volume surrounding v_0 as shown in fig.4.2.0. Recall that the flux function was constructed using a mean value linearization such that

$$f(\overline{u}_i;\vec{\mathbf{n}}_{0i}) - f(\overline{u}_0;\vec{\mathbf{n}}_{0i}) = df(\overline{u}_0,\overline{u}_i;\vec{\mathbf{n}}_{0i})\,(\overline{u}_i - \overline{u}_0) \tag{4.2.20}$$

This permits regrouping terms into the following form:

$$\frac{d}{dt}(\overline{u}_0 A_0) = -\sum_{i\in\mathcal{N}_0} f(\overline{u}_0;\vec{\mathbf{n}}_{0i}) - \sum_{i\in\mathcal{N}_0} df^-(\overline{u}_0,\overline{u}_i;\vec{\mathbf{n}}_{0i})(\overline{u}_i - \overline{u}_0) \tag{4.2.21}$$

where $(\cdot) = (\cdot)^+ + (\cdot)^-$ and $|(\cdot)| = (\cdot)^+ - (\cdot)^-$. For any closed control volume, we have that

$$\sum_{i \in \mathcal{N}_0} f(\overline{u}_0; \vec{\mathbf{n}}_{0i}) = 0.$$

Combining the remaining terms yields a final form for analysis.

$$\frac{d}{dt}(\overline{u}_0 A_0) = -\sum_{i \in \mathcal{N}_0} df^-(\overline{u}_0, \overline{u}_i; \vec{\mathbf{n}}_{0i})(\overline{u}_i - \overline{u}_0) \tag{4.2.22}$$

To verify a maximum principle at steady-state, set the time term to zero and solve for $\overline{u}_0$.

$$\overline{u}_0 = \frac{\sum_{i \in \mathcal{N}_0} df^-(\overline{u}_0, \overline{u}_i; \vec{\mathbf{n}}_{0i})\ \overline{u}_i}{\sum_{i \in \mathcal{N}_0} df^-(\overline{u}_0, \overline{u}_i; \vec{\mathbf{n}}_{0i})} = \sum_{i \in \mathcal{N}_0} \alpha_i \overline{u}_i \tag{4.2.23}$$

with $\sum_{i \in \mathcal{N}_0} \alpha_i = 1$ and $\alpha_i \geq 0$. Since $\overline{u}_0$ is a convex combination of all neighbors

$$\min_{i \in \mathcal{N}_0} \overline{u}_i \leq \overline{u}_0 \leq \max_{i \in \mathcal{N}_0} \overline{u}_i. \tag{4.2.24}$$

If $\overline{u}_0$ takes on a maximum value M in the interior, then $\overline{u}_i = M, \forall\ i \in \mathcal{N}_0$. Repeated application of (4.2.24) to neighboring control volumes establishes the maximum principle.

For explicit time stepping, a CFL-like condition is obtained for monotonicity preservation. For Euler explicit time stepping, we have the time approximation,

$$\frac{d}{dt}(\overline{u}_0 A_0) \approx A_0 \frac{\overline{u}_0^{n+1} - \overline{u}_0^n}{\Delta t}$$

which results in the following scheme:

$$\begin{aligned} \overline{u}_0^{n+1} =& \overline{u}_0^n - \frac{\Delta t}{A_0} \sum_{i \in \mathcal{N}_0} df^-(\overline{u}_0^n, \overline{u}_i^n; \vec{\mathbf{n}}_{0i})(\overline{u}_i^n - \overline{u}_0^n) \\ =& \alpha_0 \overline{u}_0^n + \sum_{i \in \mathcal{N}_0} \alpha_i \overline{u}_i^n \end{aligned} \tag{4.2.25}$$

It should be clear that coefficients in (4.2.25) sum to unity. To prove monotonicity preservation, it is sufficient to show nonnegativity of coefficients. By inspection we have that $\alpha_i \geq 0 \quad \forall\ i > 0$. To achieve monotonicity preservation we require that $\alpha_0 \geq 0$.

$$\alpha_0 = 1 + \frac{\Delta t}{A_0} \sum_{i \in \mathcal{N}_0} df^-(\overline{u}_0^n, \overline{u}_i^n; \vec{\mathbf{n}}_{0i}) \ \geq 0 \tag{4.2.26}$$

A local convex mapping from $\overline{u}^n$ to $\overline{u}^{n+1}$ exists under the CFL-like condition

$$\Delta t \leq \frac{-A_j}{\sum_{i \in \mathcal{N}_j} df^-(\overline{u}_0^n, \overline{u}_i^n; \vec{\mathbf{n}}_{0i})}, \quad \forall \Omega_j \in \Omega. \tag{4.2.27}$$

which establishes monotonicity preservation in time.

■

Note that in one dimension, this number corresponds to the conventional CFL number. In multiple space dimensions, this inequality is sufficient but not necessary for stability. In practice somewhat larger timestep values may be used.

Example: Analysis of High Order Accurate Upwind Advection Schemes Using Arbitrary Order Reconstruction.

In this example we examine maximum principle properties for a general class of high order accurate schemes on unstructured meshes. The solution algorithm is a relatively standard procedure for extensions of Godunov's scheme in Eulerian coordinates, see for example [God59], [VanL79], [ColW84], [WoodC84], [HartO85], [HartEOC86]. The basic idea in Godunov's method is to treat the integral control volume averages, $\overline{u}$, as the basic unknowns. Using information from the control volume averages, $k-th$ order piecewise polynomials are *reconstructed* in each control volume Ω_i:

$$U^k(x,y)_i = \sum_{m+n\leq k} \alpha_{(m,n)} P_{(m,n)}(x-x_c, y-y_c) \tag{4.2.28}$$

where $P_{(m,n)}(x-x_c, y-y_c) = (x-x_c)^m (y-y_c)^n$ and (x_c, y_c) is the control volume centroid. The process of reconstruction amounts to finding the polynomial coefficients, $\alpha_{(m,n)}$. Near steep gradients and discontinuities, these polynomial coefficients maybe altered based on monotonicity arguments. Because the reconstructed polynomials vary discontinuously from control volume to control volume, a unique value of the solution does not exist at control volume interfaces. This nonuniqueness is resolved via exact or approximate solutions of the Riemann problem. In practice, this is accomplished by supplanting the true flux function with a numerical flux function (described below) which produces a single unique flux given two solution states. Once the flux integral is carried out (either exactly or by numerical quadrature), the control volume average of the solution can be evolved in time. In most cases, standard techniques for integrating ODE equations are used for the time evolution, i.e. Euler implicit, Euler explicit, Runge-Kutta. The result of the evolution process is a new collection of control volume averages. The process can then be repeated. The process can be summarized in the following steps:

(1) **Reconstruction in Each Control Volume**: Given integral solution averages in all Ω_j, reconstruct a $k-th$ order piecewise polynomial $U^k(x,y)_i$ in each Ω_i for use in equation (4.2.15). In faithful implementations of Godunov's method we require that

$$\int_{\Omega_i} U^k(x,y)_i \, d\,\Omega = (\overline{u}A)_i$$

during the reconstruction process (see discussion below). For solutions containing discontinuities and/or steep gradients, monotonicity enforcement may be required.

(2) **Flux Evaluation on Each Edge**: Supplant the true flux by a numerical flux function. Given two solution states the numerical flux function returns a single unique flux. Using

the notation of the previous section we define $f(u;\mathbf{n}) = (F(u)\cdot\mathbf{n})$ so that

$$\int_{\partial\Omega_i} f(u;\mathbf{n})\; d\,\Gamma \approx \int_{\partial\Omega_i} h(U^L,U^R;\mathbf{n}) d\,\Gamma. \tag{4.2.29}$$

Consider each control volume boundary $\partial\Omega_i$, to be a collection of polygonal edges (or dual edges) from the mesh. Along each edge (or dual edge), perform a high order accurate flux quadrature. When the reconstruction polynomial is piecewise linear, single (midpoint) quadrature is usually employed on both structured and unstructured meshes

$$\int_{\partial\Omega_i} h(U^L,U^R;\mathbf{n}) d\,\Gamma \approx \sum_{j\in\mathcal{N}_i} h(U^L,U^R;\vec{\mathbf{n}})_{ij} \tag{4.2.30}$$

where U^L and U^R are solution values evaluated at the midpoint of control volume edges as shown in fig. 4.2.0. When multi-point quadrature formulas are employed we assume that they are of the form:

$$\int_0^1 f(s) d\,s \;= \sum_{q\in Q} w_q f(\xi_q)$$

with $w_q > 0$ and $\xi_q \in [0,1]$. We denote the inclusion of multi-point quadrature formulas using the augmented notation

$$\int_{\partial\Omega_i} h(U^L,U^R;\mathbf{n}) d\,\Gamma \approx \sum_{j\in\mathcal{N}_i}\sum_{q\in Q} w_q h(U^L,U^R;\vec{\mathbf{n}})_{ijq} \tag{4.2.31}$$

(3) **Evolution in Each Control Volume**: Collect flux contributions in each control volume and evolve in time using any time stepping scheme, i.e., Euler explicit, Euler implicit, Runge-Kutta, etc. The result of this step is once again control volume averages and the process can be repeated.

In the present analysis we assume that the reconstruction polynomials $U^k(x,y)_i$ in each Ω_i are given. The result of the analysis will be conditions or constraints on the reconstruction so that maximum principles can be obtained. The topic of reconstruction and implementation of the constraints determined by this analysis will be examined in a later section. Using this notation, we construct the following upwind scheme for the configuration in fig. 4.2.0.

$$\frac{d}{dt}(A_0\overline{u}_0) = -\sum_{i\in\mathcal{N}_0}\sum_{q\in Q} w_q h(U^L,U^R;\vec{\mathbf{n}})_{0iq} \tag{4.2.32}$$

with a numerical flux function obtained from (4.2.17).

$$\begin{aligned} h(U^L,U^R;\vec{\mathbf{n}}_{0i}) = &\frac{1}{2}\left(f(U^L;\vec{\mathbf{n}}) + f(U^R;\vec{\mathbf{n}})\right)_{0i} \\ &- \frac{1}{2}|df(U^L,U^R;\vec{\mathbf{n}})|_{0i}(U^R - U^L)_{0i} \end{aligned} \tag{4.2.33}$$

To analyze the scheme, we use the fact that the flux function was constructed using a mean value linearization such that

$$f(U^R;\vec{\mathbf{n}}) - f(U^L;\vec{\mathbf{n}}) = df(U^L,U^R;\vec{\mathbf{n}})(U^R - U^L)$$

This permits regrouping terms into the following form:

$$\frac{d}{dt}(\overline{u}_0 A_0) = -\sum_{i\in\mathcal{N}_0}\sum_{q\in Q} w_q \left(f(U^L;\vec{\mathbf{n}}) + df^-(U^L,U^R;\vec{\mathbf{n}})(U^R - U^L)\right)_{0iq} \tag{4.2.34}$$

Next we rewrite the first term in the sum using a mean value construction

$$\sum_{i\in\mathcal{N}_0}\sum_{q\in Q} w_q f(\overline{u}_0;\vec{\mathbf{n}})_{0iq} + w_q \left(df(\overline{u}_0,U^L;\vec{\mathbf{n}})(U^L - \overline{u}_0)\right)_{0iq}$$

The first term vanishes when summed over a closed volume so that (4.2.34) reduces to

$$\frac{d}{dt}(\overline{u}_0 A_0) = -\sum_{i\in\mathcal{N}_0}\sum_{q\in Q} w_q \left(df(\overline{u}_0,U^L;\vec{\mathbf{n}})(U^L - \overline{u}_0) + df^-(U^L,U^R;\vec{\mathbf{n}})(U^R - U^L)\right)_{0iq} \tag{4.2.35}$$

By introducing difference ratios we rewrite the scheme in the following form:

$$\begin{aligned}\frac{d}{dt}(\overline{u}_0 A_0) = &-\sum_{i\in\mathcal{N}_0}\sum_{q\in Q} w_q \left(df^-(\overline{u}_0,U^L;\vec{\mathbf{n}})\Psi\right)_{0iq}(\overline{u}_i - \overline{u}_0)\\ &-\sum_{i\in\mathcal{N}_0}\sum_{q\in Q} w_q \left(df^+(\overline{u}_0,U^L;\vec{\mathbf{n}})\Phi\right)_{0iq}(\overline{u}_0 - \overline{u}_k)\\ &-\sum_{i\in\mathcal{N}_0}\sum_{q\in Q} w_q \left(df^-(U^L,U^R;\vec{\mathbf{n}})\Theta\right)_{0iq}(\overline{u}_i - \overline{u}_0)\end{aligned} \tag{4.2.36}$$

with

$$\Psi_{0iq} = \frac{U^L_{0iq} - \overline{u}_0}{\overline{u}_i - \overline{u}_0}, \quad \Phi_{0iq} = \frac{U^L_{0iq} - \overline{u}_0}{\overline{u}_0 - \overline{u}_k}, \quad \Theta_{0iq} = \frac{U^R_{0iq} - U^L_{0iq}}{\overline{u}_i - \overline{u}_0} \tag{4.2.37}$$

In this equation, the k subscript refers to some as yet unspecified index value such that $k \in \mathcal{N}_0$.

Theorem 4.2.3: The generalized Godunov scheme with arbitrary order reconstruction (4.2.32) exhibits a discrete maximum principle at steady-state if the following three conditions are fulfilled:

$$\Psi_{jiq} \geq 0, \quad \Phi_{jiq} \geq 0 \quad \Theta_{jiq} \geq 0 \quad \forall j,q,\ i \in \mathcal{N}_j. \tag{4.2.38}$$

Furthermore, the scheme is monotonicity preserving under a CFL-like condition if $\forall\ \Omega_j \in \Omega$:

$$\Delta t \leq \frac{-A_j}{\sum_{i\in\mathcal{N}_j}\sum_{q\in Q}\left(\overline{df}^-\Psi - \overline{df}^+\Phi + df^-\Theta\right)_{jiq}}$$

Proof: Assume that (4.2.38) holds. Define $\overline{df}_{0iq} = w_q df(\overline{u}_0, U^L; \vec{\mathbf{n}})_{0iq}$ and similarly $df_{0iq} = w_q df(U^L, U^R; \vec{\mathbf{n}})_{0iq}$. Setting the time term to zero and solving for $\overline{u}_0$ yields

$$\begin{aligned}\overline{u}_0 =& \frac{\sum_{i\in\mathcal{N}_0}\sum_{q\in Q}\left(\overline{df}^{+}\Phi\right)_{0iq}\overline{u}_k - \left(\overline{df}^{-}\Psi + df^{-}\Theta\right)_{0iq}\overline{u}_i}{\sum_{i\in\mathcal{N}_0}\sum_{q\in Q}\left(\overline{df}^{+}\Phi - \overline{df}^{-}\Psi - df^{-}\Theta\right)_{0iq}} \\ =& \sum_{i\in\mathcal{N}_0}\alpha_i\overline{u}_i.\end{aligned}$$

Examining the individual coefficients we see that $\sum_{i\in\mathcal{N}_0}\alpha_i = 1$ and $\alpha_i \geq 0$. Thus a convex local mapping exists and we obtain

$$\min_{i\in\mathcal{N}_0}\overline{u}_i \leq \overline{u}_0 \leq \max_{i\in\mathcal{N}_0}\overline{u}_i. \tag{4.2.40}$$

If $\overline{u}_0$ takes on a maximum value M in the interior, then $\overline{u}_i = M, \forall\ i \in \mathcal{N}_0$. Repeated application of (4.2.40) to neighboring control volumes establishes the maximum principle.

To establish monotonicity preservation in time we consider the Euler explicit timestepping scheme.

$$\frac{d}{dt}(\overline{u}_0 A_0) \approx A_0\frac{\overline{u}_0^{n+1} - \overline{u}_0^n}{\Delta t}$$

Inserting this formula into (4.2.36) yields

$$\begin{aligned}\overline{u}_0^{n+1} =& \overline{u}_0^n - \frac{\Delta t}{A_0}\sum_{i\in\mathcal{N}_0}\sum_{q\in Q}\overline{df}^{-}_{0iq}\Psi_{0iq}(\overline{u}_i^n - \overline{u}_0^n) \\ &-\frac{\Delta t}{A_0}\sum_{i\in\mathcal{N}_0}\sum_{q\in Q}\overline{df}^{+}_{0iq}\Phi_{0iq}(\overline{u}_0^n - \overline{u}_k^n) \\ &-\frac{\Delta t}{A_0}\sum_{i\in\mathcal{N}_0}\sum_{q\in Q}df^{-}_{0iq}\Theta_{0iq}(\overline{u}_i^n - \overline{u}_0^n) \\ =& \alpha_0\overline{u}_0^n + \sum_{i\in\mathcal{N}_0}\sum_{q\in Q}\alpha_i\overline{u}_i^n\end{aligned} \tag{4.2.41}$$

with $\alpha_0 + \sum_{i\in\mathcal{N}_0}\alpha_i = 1$ and $\alpha_i \geq 0, i > 0$. A locally convex mapping in time from U^n to U^{n+1} is achieved when $\alpha_0 \geq 0$. This assures monotonicity in time. Some algebra reveals the following formula for α_0

$$\alpha_0 = 1 + \frac{\Delta t}{A_0}\sum_{i\in\mathcal{N}_0}\sum_{q\in Q}\left(\overline{df}^{-}\Psi - \overline{df}^{+}\Phi + df^{-}\Theta\right)_{0iq}$$

From this we obatin the CFL-like condition for monotonicity preservation

$$\Delta t \leq \frac{-A_0}{\sum_{i\in\mathcal{N}_0}\sum_{q\in Q}\left(\overline{df}^{-}\Psi - \overline{df}^{+}\Phi + df^{-}\Theta\right)_{0iq}}$$

so that

$$\min_{i\in\mathcal{N}_0}(\overline{u}_i^n, \overline{u}_0^n) \le \overline{u}_0^{n+1} \le \max_{i\in\mathcal{N}_0}(\overline{u}_i^n, \overline{u}_0^n)$$

Applying this result to all control volumes establishes monotonicity preservation.
■

5.0 Finite-Volume Schemes for Scalar Conservation Law Equations

In this section we consider the specific application of the maximum principle theory of Section 4 to the design of numerical schemes for solving conservation law equations. Some portions of this section will repeat ideas introduced in Section 4 but now emphasis is placed on the details of implementation.

5.1 Conservation Laws

Definition: A conservation law asserts that the rate of change of the total amount of a substance with density z in a fixed region Ω is equal to the flux $\mathbf{F}$ of the substance through the boundary $\partial\Omega$.

$$\frac{d}{dt}\int_\Omega z \; d\,a + \int_{\partial\Omega} \mathbf{F}(z)\cdot\mathbf{n} \; d\,l = 0 \qquad \text{(integral form)}$$

The choice of a numerical algorithm used to solve a conservation law equation is often influenced by the form in which the conservation law is presented. A finite-difference practitioner would apply the divergence theorem to the integral form and let the area of Ω shrink to zero thus obtaining the divergence form of the equation.

$$\frac{d}{dt}z + \nabla\cdot\mathbf{F}(z) = 0 \qquad \text{(divergence form)}$$

The finite-element practitioner constructs the divergence form then multiplies by an arbitrary test function ϕ and integrates by parts.

$$\begin{aligned}\frac{d}{dt}\int_\Omega \phi z \; d\,a - \int_\Omega \nabla\phi\cdot\mathbf{F}(z) \; d\,a \\ + \int_{\partial\Omega} \phi\mathbf{F}(z)\cdot\mathbf{n} \; d\,l = 0\end{aligned} \qquad \text{(weak form)}$$

Algorithm developers starting from these three forms can produce seemingly different numerical schemes. In reality, the final discretizations are usually very similar. Some differences do appear in the handling of boundary conditions, solution discontinuities, and nonlinearities. When considering flows with discontinuities, the integral form appears advantageous since conservation of fluxes comes for free and the proper jump conditions are assured. Discretizations of the divergence form on structured meshes yield conventional finite-difference formulas. The divergence form of the equations is rarely used in the discretization of conservation law equations on unstructured meshes because of the difficulty in ensuring conservation and the proper jump conditions. The weak form of the equation guarantees satisfaction of the jump conditions over the extent of the support and is fully

compatible with unstructured meshes. For these reasons the integral and weak forms are both used extensively in numerical modeling of conservation laws on unstructured meshes. The remaining portion of these notes will be devoted to the finite-volume method which is a technique for obtaining discretizations based on the integral conservation law form of the equations.

5.2 Upwind Finite-Volume Schemes Via Godunov's Method

In this section, we consider upwind algorithms for scalar hyperbolic equations. In particular, we concentrate on upwind schemes based on Godunov's method [God59]. The development presented here follows many of the ideas developed previously for structured meshes. For example, in the extension of Godunov's scheme to second order accuracy in one space dimension, van Leer [VanL79] developed an advection scheme based on the *reconstruction* of discontinuous piecewise linear distributions together with Lagrangian hydrodynamics. Colella and Woodward [ColW84] and Woodward and Colella [WoodC84] further extended these ideas to include discontinuous piecewise parabolic approximations with Eulerian or Lagrangian hydrodynamics. Harten et. al. [HartO85], [HartEOC86] later extended related schemes to arbitrary order and clarified the entire process. These techniques have been applied to structured meshes in multiple space dimensions by applying one-dimensional-like schemes along individual coordinate lines. This has proven to be a highly successful approximation but does not directly extend to unstructured meshes. In reference [BarJ89], we proposed a scheme for multi-dimensional reconstruction on unstructured meshes using discontinuous piecewise linear distributions of the solution in each control volume. Monotonicity of the reconstruction was enforced using a limiting procedure similar to that proposed by van Leer for structured grids. In later papers ([BarF90], [Bar93]) we developed numerical schemes for unstructured meshes utilizing a reconstruction algorithm of arbitrary order. Recall that in section 4.2 the three basic steps in the generalized Godunov scheme were outlined:

(1) **Reconstruction in Each Control Volume**: Given integral cell averages in all control volumes, reconstruct piecewise polynomial approximations in each control volume, $U^k(x,y)_i$, for use in the integral conservation law equation (4.2.29). For solutions containing discontinuities or steep gradients, monotonicity enforcement may be required.

(2) **Flux Evaluation on Each Edge**: Consider each control volume boundary, $\partial\Omega_i$, to be a collection of edges (or dual edges) from the mesh. Along each edge (or dual edge), perform a high order accurate flux quadrature.

(3) **Evolution in Each Control Volume**: Collect flux contributions in each control volume and evolve in time using any time stepping scheme, i.e., Euler explicit, Euler implicit, Runge-Kutta, etc. The result of this process is once again cell averages.

By far, the most difficult of these steps is the polynomial reconstruction given cell averages. In the following paragraphs, we describe design criteria for a general reconstruction operator with fixed stencil size. In work by Vankeirsbilck and Deconinck [VankD92] and Michell [Mich94] ENO-like schemes have been constructed for both unstructured and structured meshes.

The reconstruction operator serves as a finite-dimensional (possibly pseudo) inverse of the cell-averaging operator $\mathbf{A}$ whose j-th component $\mathbf{A}_j$ computes the cell average of the solution in Ω_j.

$$\overline{u}_j = \mathbf{A}_j u = \frac{1}{A_j} \int_{\Omega_j} u(x,y)\, d\Omega \tag{5.2.0}$$

In addition, we place the following additional requirements:

(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.

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

This means that $\mathbf{R}^k$ is a right inverse of the averaging operator $\mathbf{A}$.

$$\mathbf{A}\mathbf{R}^k = I \tag{5.2.1}$$

Conservation of the mean has an important implication. Unlike finite-element schemes, *Godunov schemes have a diagonal mass matrix.*

(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.

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

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

$$\mathbf{R}^k \mathbf{A}\Big|_{\mathcal{P}_k} = I \tag{5.2.2}$$

This insures that exact solutions contained in $\mathcal{P}_k$ are in fact solutions of the discrete equations. For sufficiently smooth solutions, the property of k-exactness also insures that when piecewise polynomials are evaluated at control volume boundaries, the difference between solution states diminishes with increasing k at a rate proportional to h^{k+1} were h is a maximum diameter of the two control volumes. Figure 5.2.0 shows a global quartic polynomial $u \in \mathcal{P}_4$ which has been averaged in each interval.

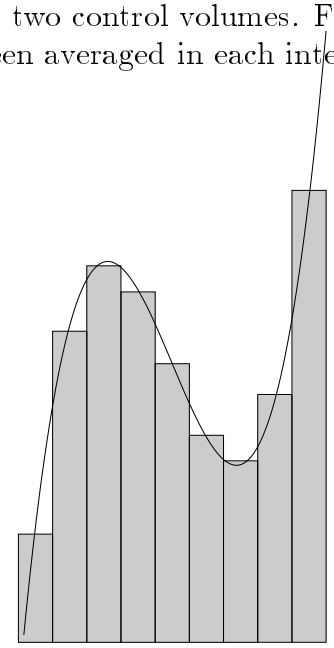

Figure 5.2.0 Cell averaging of quartic polynomial.

Figure 5.2.1 shows a quadratic reconstruction $u^k \in \mathcal{P}_2$ given the cell averages. Close inspection of fig. 5.2.1 reveals small jumps in the piecewise polynomials at interval boundaries.

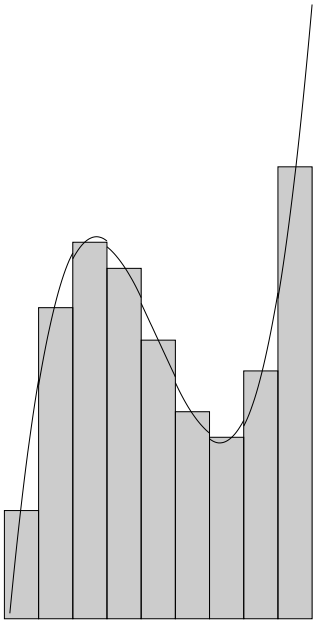

Figure 5.2.1 Piecewise quadratic reconstruction.

These jumps would decrease even more for cubics and vanish altogether for quartic reconstruction. Property (1) requires that the area under each piecewise polynomial is exactly equal to the cell average.

5.3 Linear Reconstruction

In this section, we consider algorithms for linear reconstruction. The process of linear reconstruction in one dimension is depicted in fig. 5.3.0.

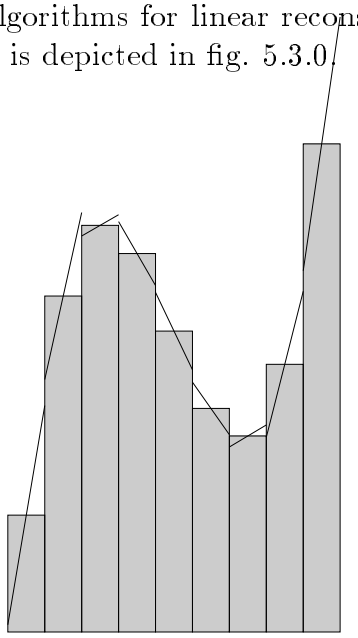

Figure 5.3.0 Linear Reconstruction of cell-averaged data.

One of the most important observations concerning linear reconstruction is that we can dispense with the notion of cell averages as unknowns by reinterpreting the unknowns as pointwise values of the solution sampled at the centroid (midpoint in 1-D) of the control volume. This well known result greatly simplifies schemes based on linear reconstruction. The linear reconstruction in each interval shown in figure 5.3.0 was obtained by a simple central-difference formula given point values of the solution at the midpoint of each interval.

In section 5.5, results for arbitrary order reconstruction will be presented. The reconstruction strategy presented there satisfies all the design requirements of the reconstruction operator. For linear reconstruction, simpler formulations are possible which exploit the edge data structure mentioned in Section 1. For each edge of a triangulation, we store the following two pieces of information:

(1) The two vertices which form the edge.

(2) The centroid location of the two cell which share that edge.

Several of these reconstruction schemes are given below. Note that for steady-state computations, conservation of the mean in the data reconstruction is not necessary. The implication of violating this conservation is that a *nondiagonal* mass matrix appears in the time integral. Since time derivatives vanish at steady-state, the effect of this mass matrix vanishes at steady-state. The reconstruction schemes presented below assume that solution variables are placed at the vertices of the mesh, which may not be at the precise centroid of the control volume, thus violating conservation of the mean. Even so, linear reconstruction methods given below can all be implemented using an edge data structure and satisfy k-exactness for linear functions.

5.3a Green-Gauss Reconstruction

Consider a domain Ω' consisting of all triangles incident to some vertex v_0, see fig. 5.3.0 and the exact integral relation

$$\int_{\Omega'} \nabla u \; d\,\Omega = \int_{\partial\Omega'} u\mathbf{n} \; d\,\Gamma.$$

In [BarJ89] we show that given function values at vertices of a triangulation, a discretization of this formula can be constructed which is exact whenever u varies linearly:

$$(\nabla u)_{v_0} = \frac{1}{A_0} \sum_{i \in \mathcal{N}_0} \frac{1}{2}(u_i + u_0)\vec{\mathbf{n}}_{0i}. \tag{5.3.0}$$

In this formula $\vec{\mathbf{n}}_{0i} = \int_a^b d\,\vec{\mathbf{n}}$ for any path which connects triangle centroids adjacent to the edge $e(v_0, v_i)$ and A_0 is the area of the *nonoverlapping* dual regions formed by this choice of path integration. Two typical choices are the median and centroid duals as shown below.

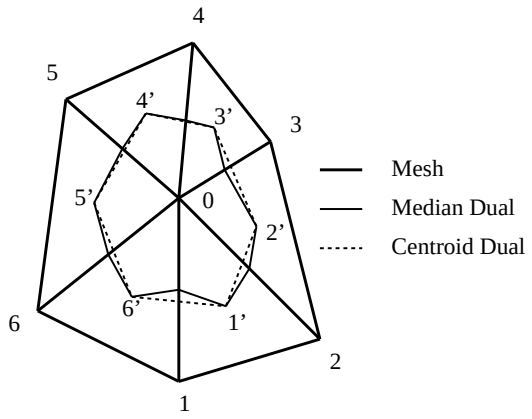

Figure 5.3.0 Local mesh with centroid and median duals.

This approximation extends naturally to three dimensions, see [Bar91b]. The formula (5.3.0) suggests a natural computer implementation using the edge data structure. Assume that the normals $\vec{\mathbf{n}}_{ij}$ for all edges $e(v_i, v_j)$ have been precomputed with the convention that the normal vector points from v_i to v_j. An edge implementation of 5.3.0 can be performed in the following way:

For $k = 1, n(e)$ *! Loop through edges of mesh*
  $j_1 = e^{-1}(k, 1)$ *!Pointer to edge origin*
  $j_2 = e^{-1}(k, 2)$ *!Pointer to edge destination*
  $uav = (u(j_1) + u(j_2))/2$ *!Gather*
  $ux(j_1) += normx(k) \cdot uav$ *!Scatter*
  $ux(j_2) -= normx(k) \cdot uav$
  $uy(j_1) += normy(k) \cdot uav$
  $uy(j_2) -= normy(k) \cdot uav$
Endfor

For $j = 1, n(v)$ *! Loop through vertices*
  $ux(j) = ux(j)/area(j)$ *! Scale by area*
  $uy(j) = uy(j)/area(j)$
Endfor

It can be shown that the use of edge formulas for the computation of vertex gradients is asymptotically optimal in terms of work done.

5.3b Linear Least-Squares Reconstruction

To derive this reconstruction technique, consider a vertex v_0 and suppose that the solution varies linearly over the support of adjacent neighbors of the mesh. In this case, the change in vertex values of the solution along an edge $e(v_i, v_0)$ can be calculated by

$$(\nabla u)_0 \cdot (\mathbf{R}_i - \mathbf{R}_0) = u_i - u_0 \tag{5.3.1}$$

This equation represents the scaled projection of the gradient along the edge $e(v_i, v_0)$. A similar equation could be written for all incident edges subject to an arbitrary weighting factor. The result is the following matrix equation, shown here in two dimensions:

$$\begin{bmatrix} w_1\Delta x_1 & w_1\Delta y_1 \\ \vdots & \vdots \\ w_n\Delta x_n & w_n\Delta y_n \end{bmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} w_1(u_1 - u_0) \\ \vdots \\ w_n(u_n - u_0) \end{pmatrix} \tag{5.3.2}$$

or in symbolic form $\mathcal{L}\ \nabla u = \mathbf{f}$ where

$$\mathcal{L} = \begin{bmatrix} \vec{L}_1 & \vec{L}_2 \end{bmatrix} \tag{5.3.3}$$

in two dimensions. Exact calculation of gradients for linearly varying u is guaranteed if any two row vectors $w_i(\mathbf{R}_i - \mathbf{R}_0)$ span all of 2 space. This implies linear independence of $\vec{L}_1$ and $\vec{L}_2$. The system can then be solved via a Gram-Schmidt process, i.e.,

$$\begin{bmatrix} \vec{V}_1 \\ \vec{V}_2 \end{bmatrix} \begin{bmatrix} \vec{L}_1 & \vec{L}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.3.4}$$

The row vectors $\vec{V}_i$ are given by

$$\begin{aligned} \vec{V}_1 &= \frac{l_{22}\vec{L}_1 - l_{12}\vec{L}_2}{l_{11}l_{22} - l_{12}^2} \\ \vec{V}_2 &= \frac{l_{11}\vec{L}_2 - l_{12}\vec{L}_1}{l_{11}l_{22} - l_{12}^2} \end{aligned}$$

with $l_{ij} = (\vec{L}_i \cdot \vec{L}_j)$.

Note that reconstruction of N independent variables in $\mathbf{R}^d$ implies $\binom{d+1}{2} + d\,N$ inner product sums. Since only $d\,N$ of these sums involves the solution variables themselves, the remaining sums could be precalculated and stored in computer memory. This makes the present scheme competitive with the Green-Gauss reconstruction. Using the edge data structure, the calculation of inner product sums can be calculated for *arbitrary* combinations of polyhedral cells. In all cases linear functions are reconstructed exactly. We demonstrate this idea by example:

For $k = 1, n(e)$ *!Loop through edges of mesh*
  $j_1 = e^{-1}(k, 1)$ *!Pointer to edge origin*
  $j_2 = e^{-1}(k, 2)$ *!Pointer to edge destination*
  $dx = w(k) \cdot (x(j_2) - x(j_1))$ *!Weighted* Δx
  $dy = w(k) \cdot (y(j_2) - y(j_1))$ *!Weighted* Δy
  $l_{11}(j_1) = l_{11}(j_1) + dx \cdot dx$ *! l_{11} orig sum*
  $l_{11}(j_2) = l_{11}(j_2) + dx \cdot dx$ *! l_{11} dest sum*
  $l_{12}(j_1) = l_{12}(j_1) + dx \cdot dy$ *! l_{12} orig sum*
  $l_{12}(j_2) = l_{12}(j_2) + dx \cdot dy$ *! l_{12} dest sum*

$du = w(k) \cdot (u(j_2) - u(j_1))$ *!Weighted* Δu
$lf_1(j_1) += dx \cdot du$ *!*$\vec{L}_1 f$ *sum*
$lf_1(j_2) += dx \cdot du$
$lf_2(j_1) += dy \cdot du$ *!*$\vec{L}_2 f$ *sum*
$lf_2(j_2) += dy \cdot du$
Endfor

For $j = 1, n(v)$ *! Loop through vertices*
$det = l_{11}(j) \cdot l_{22}(j) - l_{12}^2$
$ux(j) = (l_{22}(j) \cdot lf_1(j) - l_{12} \cdot lf_2)/det$
$uy(j) = (l_{11}(j) \cdot lf_2(j) - l_{12} \cdot lf_1)/det$
Endfor

This formulation provides freedom in the choice of weighting coefficients, w_i. These weighting coefficients can be a function of the geometry and/or solution. Classical approximations in one dimension can be recovered by choosing geometrical weights of the form $w_i = 1./|\mathbf{\Delta r}_i - \mathbf{\Delta r}_0|^t$ for values of $t = 0, 1, 2$. The L_2 gradient calculation technique is optimal in a weighted least squares sense and determines gradient coefficients with least sensitivity to Gaussian noise. This is an important property when dealing with highly distorted (stretched) meshes which typically arise in the computation of viscous flow.

5.3c Data Dependent Reconstruction

Both the Green-Gauss and L_2 gradient calculation techniques can be generalized to include data dependent (i.e. solution dependent) weights. In the case of Green-Gauss formulation, the sum

$$\sum_{i \in \mathcal{I}_0} \frac{1}{2}(u_0 + u_i)\vec{\mathbf{n}}_{0i}$$

is replaced by

$$\begin{aligned} \sum_{i \in \mathcal{I}_0} p_{0i}^- \frac{1}{2}(u_0 + u_i)\vec{\mathbf{n}}_{0i} + \\ p_{0i}^+ \frac{1}{2}\left((\nabla u)_0 \cdot (\mathbf{R}_i - \mathbf{R}_0)\right)\vec{\mathbf{n}}_{0i} \end{aligned} \tag{5.3.5}$$

If the $p_{0i}^{\pm}$ are chosen such that $p_{0i}^- + p_{0i}^+ = 1$ then the gradient calculation is exact whenever the solution varies linearly over the support. In two space dimensions, equation (5.3.5) implies the solution of a linear 2×2 system of the form

$$\begin{bmatrix} A_0 - m_{xx} & -m_{xy} \\ -m_{yx} & A_0 - m_{yy} \end{bmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \sum_{i \in \mathcal{I}_0} p_{0i}^- \frac{1}{2}(u_0 + u_i)\vec{\mathbf{n}}_{0i}$$

where

$$m_{xx} = \sum_{i \in \mathcal{I}_0} p_{0i}^+ \Delta x_i n_{x_i}, \quad m_{yy} = \sum_{i \in \mathcal{I}_0} p_{0i}^+ \Delta y_i n_{y_i}$$

$$m_{xy} = \sum_{i \in \mathcal{I}_0} p_{0i}^+ \Delta x_i n_{y_i}, \quad m_{yx} = \sum_{i \in \mathcal{I}_0} p_{0i}^+ \Delta y_i n_{x_i}$$

Care must be exercised in the selection of $p^{\pm}$ in order that the system be invertible. This is similar to the spanning space requirement of the least-squares gradient calculation technique.

5.3d Monotonicity Enforcement

When solution discontinuites and steep gradients as present, additional steps must be taken to prevent oscillations from developing in the numerical solution. One way to do this was pioneered by van Leer [VanL79] in the late 1970's. The basic idea is to take the reconstructed piecewise polynomials and enforce strict monotonicity in the reconstruction. Monotonicity in this context means that the value of the reconstructed polynomial does not exceed the minimum and maximum of neighboring cell averages. The final reconstruction must guarantee that no new extrema have been created. When a new extremum is produced, the slope of the reconstruction in that interval is reduced until monotonicity is restored. This implies that at a local minimum or maximum in the cell averaged data the slope in 1-D is *always* reduced to zero, see for example fig. 5.3.2.

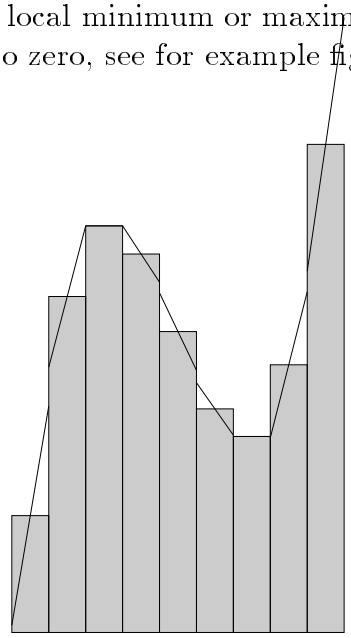

Figure 5.3.2 Linear Data Reconstruction with monotone limiting.

Theorem 4.2.3 provides sufficient conditions for a discrete maximum principle using arbitrary order reconstruction. Consider the control volume interface separating Ω_i and Ω_j as shown in fig. 5.3.3.

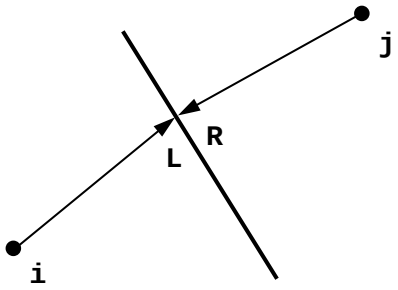

Figure 5.3.3 Control volume interface between Ω_i and Ω_j showing left and right extrapolations.

From Theorem 4.2.3 we can guarantee a maximum principle if for all quadrature points on the interface separating Ω_i and Ω_j

$$\Psi \geq 0, \quad \Phi \geq 0 \quad \Theta \geq 0.$$

By performing the analysis for the control volume Ω_i we determined the following conditions:

$$\begin{aligned} 0 \leq & \frac{U^L - \overline{u}_i}{\overline{u}_j - \overline{u}_i} \quad (a) \\ 0 \leq & \frac{U^L - \overline{u}_i}{\overline{u}_i - \overline{u}_k} \quad (b) \\ 0 \leq & \frac{U^R - U^L}{\overline{u}_j - \overline{u}_i} \quad (c) \end{aligned} \tag{5.3.6}$$

where constraint (b) should be interpreted as find *any* $k \in \mathcal{N}_i$ such that the inequality exists. By considering the neighboring control volumes, constraint (c) is reproduced at shared interfaces and the remaining new constraints do not involve the reconstruction polynomial in the other control volume. In other words, constraints (a) and (b) are purely local to a control volume and can be enforced on an individual control volume basis. Constraint (c) is not local and must be enforced pairwise at common interface boundaries.

The entire picture becomes clearer by considering a one-dimensional mesh with points contiguously numbered $i = 1, 2, ...N$. In this scenario, we would have that $j = i + 1$ and k would necessarily be equal to $i - 1$ so that previous result (5.3.6) reduces to

$$\begin{aligned} 0 \leq & \frac{U^L - \overline{u}_i}{\overline{u}_{i+1} - \overline{u}_i} \\ 0 \leq & \frac{U^L - \overline{u}_i}{\overline{u}_i - \overline{u}_{i-1}} \\ 0 \leq & \frac{U^R - U^L}{\overline{u}_{i+1} - \overline{u}_i} \end{aligned} \tag{5.3.7}$$

for the interface separating i and $i+1$. Looking from Ω_{i+1} at the same interface from $i+1$ to i yields the additional constraints:

$$\begin{aligned} 0 \leq & \frac{U^R - \overline{u}_{i+1}}{\overline{u}_i - \overline{u}_{i+1}} \\ 0 \leq & \frac{U^R - \overline{u}_{i+1}}{\overline{u}_{i+1} - \overline{u}_{i+2}} \\ 0 \leq & \frac{U^R - U^L}{\overline{u}_{i+1} - \overline{u}_i}. \end{aligned} \tag{5.3.8}$$

The local constraints (a) and (b) in (5.3.7) and (5.3.8) are satisfied if the reconstructed U^L and U^R are bounded between the minimum and maximum of neighboring cell averages. The last inequality appearing in (5.3.7-5.3.8) requires that the difference in the extrapolated

states at a cell interface must be of the same sign as the difference in the cell average values. For example in fig. 5.3.4(a) this condition is violated but can be remedied either by a symmetric reduction of slopes or by replacing the larger slope by the minimum value of the two slopes.

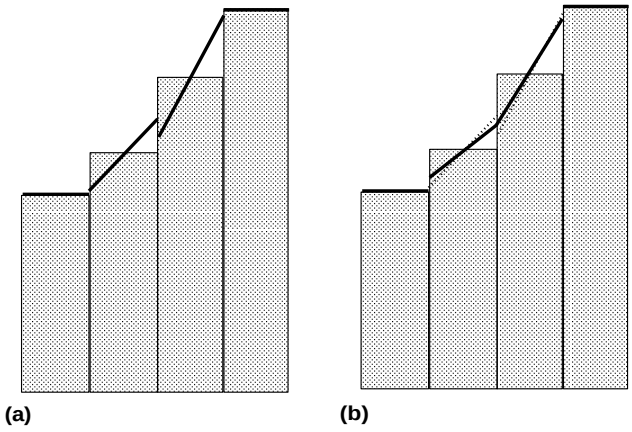

Figure 5.3.4 (a) Reconstruction profile with increased variation violating monotonicity constraints. (b) Profile after modification to satisfy monotonicity constraints.

Observe that in one space dimension the net effect of the slope limiting in the reconstruction process is to ensure that the total variation of the reconstructed function does not exceed the total variation of the cell averaged data.

Next we consider the implementation of slope limiting procedures on unstructured meshes. In Barth and Jespersen [BarJ89], we gave a simple recipe for slope limiting on arbitrary unstructured meshes. Extensive testing has shown that the limiting characteristics of our limiter can damage the overall accuracy of computations, especially for flows on coarse meshes. In addition the limiter behaves poorly when the solution is nearly constant unless additional heuristic parameters are added. This has prompted other researchers (c.f. [Ven93], [AftGT94]) to proposed alternative limiter functions, but no serious attempt is made to appeal to the rigors of maximum principle theory. The design of accurate limiters satisfying the maximum principle constraints is a topic of current research.

In the following paragraphs the limiter function developed in [BarJ89] is discussed. Consider writing the linearly reconstructed data in the following form for Ω_0:

$$U(x,y)_0 = \overline{u}_0 + \nabla u_0 \cdot (\mathbf{r} - \mathbf{r}_0) \tag{5.3.9}$$

Now consider a "limited" form of this piecewise linear distribution.

$$U(x,y)_0 = \overline{u}_0 + \Phi_0 \nabla u_0 \cdot (\mathbf{r} - \mathbf{r}_0) \tag{5.3.10}$$

Next compute the minimum and maximum of all adjacent neighbors

$$u_j^{min} = \min_{i \in \mathcal{N}_j} (\overline{u}_0, \overline{u}_i)$$

$$u_j^{max} = \max_{i \in \mathcal{N}_j}(\overline{u}_0, \overline{u}_i)$$

and require that

$$u_j^{min} \leq U(x,y)_0 \leq u_j^{max} \tag{5.3.11}$$

when evaluated at the quadrature points used in the flux integral computation. For each quadrature point location in the flux integral compute the extrapolated state U_{0i}^L and determine the smallest Φ_0 so that

$$\Phi_0 = \begin{cases} \min(1, \frac{u_j^{max} - \overline{u}_0}{U_{0i}^L - \overline{u}_0}), & \text{if } U_{0i}^L - \overline{u}_0 > 0 \\ \min(1, \frac{u_j^{min} - \overline{u}_0}{U_{0i}^L - \overline{u}_0}), & \text{if } U_{0i}^L - \overline{u}_0 < 0 \\ 1 & \text{if } U_{0i}^L - \overline{u}_0 = 0 \end{cases} .$$

When the above procedures are combined with the flux function given earlier (4.2.23),

$$\begin{aligned} h(U^L, U^R; \mathbf{n}) = & \frac{1}{2}\left(f(U^L; \mathbf{n}) + f(U^R; \mathbf{n})\right) \\ & - \frac{1}{2}|df(U^L, U^R; \mathbf{n})|\left(U^R - U^L\right) \end{aligned} \tag{5.3.12}$$

the resulting scheme has good shock resolving characteristics. To demonstrate this, we consider the scalar nonlinear hyperbolic problem suggested by Struijs, Deconinck, *et al* [StruVD89]. The equation is a multidimensional form of Burger's equation.

$$u_t + (u^2/2)_x + u_y = 0$$

This equation is solved in a square region $[0, 1.5] \times [0, 1.5]$ with boundary conditions: $u(x,0) = 1.5 - 2x,\ x \leq 1$, $u(x,0) = -.5,\ x > 1$, $u(0,y) = 1.5$, and $u(1.5,y) = -.5$. Figures 5.3.5 and 5.3.6 show carpet plots and contours of the solution on regular and irregular meshes.

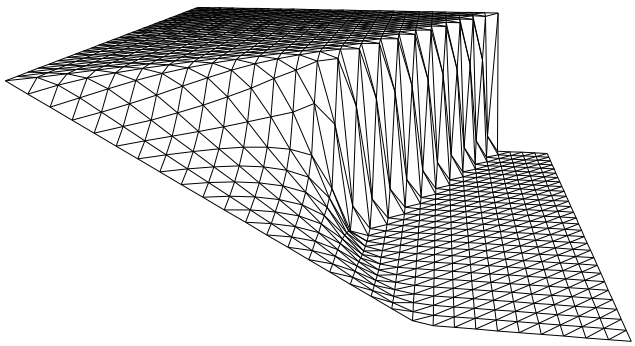

Figure 5.3.5a Carpet plot of Burger's equation solution on regular mesh.

The exact solution to this problem consists of converging straightline characteristics which eventually form a shock which propagates to the upper boundary.

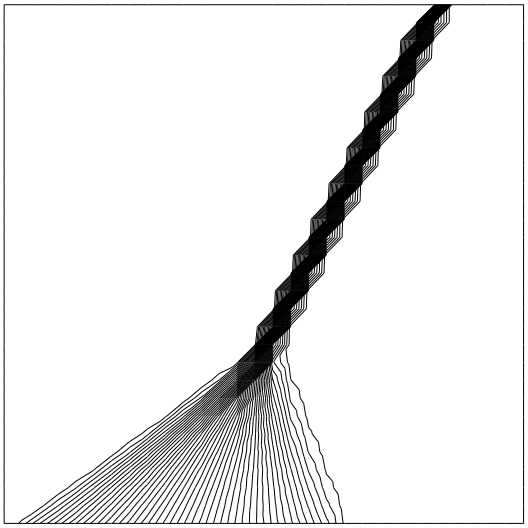

Figure 5.3.5b Solution Contours on regular mesh.

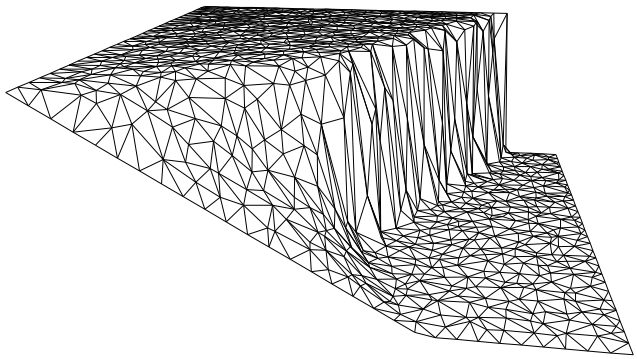

Figure 5.3.6a Carpet plot of Burger's equation solution on irregular mesh.

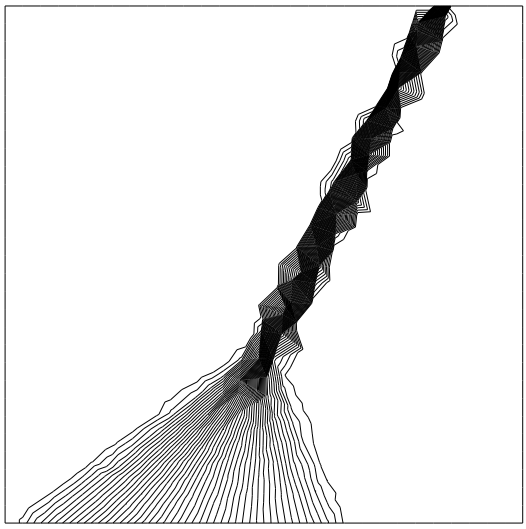

Figure 5.3.6b Solution contours.

The carpet plots indicate that the numerical solution on both meshes is monotone. Even so, most people would prefer the solution on the regular mesh. This is an unavoidable consequence of irregular meshes. The only remedy appears to be mesh adaptation. Similar results for the Euler equations will be shown on irregular meshes in the next section.

5.4 Simplified Quadratic Reconstruction

Several possible strategies exist for piecewise quadratic reconstruction. As a general design criterion we require that the reconstruction exhibit quadratic precision so that the reconstruction will be exact whenever the solution varies quadratically. Recall that a quadratic polynomial contains six degrees of freedom in two dimensions. This means that the support (stencil) of the reconstruction operator must contain at least six members. In [BarF] and [VankD92] this is accomplished by increasing the physical support on the mesh until six or more members are included. This approach has several pitfalls. Increasing the physical support may not always be possible due to the presence of boundaries. In this situation either the data support must be shifted in an unnatural way or the order of polynomial reconstruction must be lowered. Increasing the physical support also has the undesirable effect of bringing less relevant data into the reconstruction. One example would be data reconstruction in a flow field containing two shock waves in close proximity.

Another approach to higher order reconstruction is to add additional degrees of freedom into the solution representation. To simplify the approach, pointwise values of the solution are assumed rather than cell averages. Thus the resulting scheme has a nondiagonal mass matrix multiplying time derivates. We presume that this matrix can be safely ignored for steady-state computations.

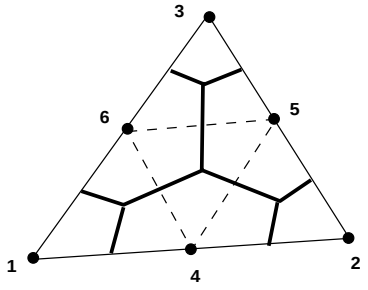

Figure 5.4.0 Six noded quadratic element with control volume tessellation (bold lines).

In our approach a quadratic element approximation is used, see Fig. 5.4.0. The solution unknowns are the six nodal values. These values uniquely describe a quadratic function within the element. The control volumes for the scheme are then formed from a tessellation of the elements. The particular tessellation which we prefer is obtained by connecting centroids and edges as shown in Fig. 5.4.0.

For each control volume surrounding a vertex or midside node, v_0, a quadratic polynomial of the form

$$u(x,y)_0 = u_0 + \mathbf{\Delta r}^T \nabla u_0 + \frac{1}{2}\mathbf{\Delta r}^T H_0 \mathbf{\Delta r} \tag{5.3.13}$$

must be reconstructed from surrounding data. In this equation ∇u is the usual solution gradient and H is the Hessian matrix of second derivatives

$$H = \begin{bmatrix} u_{xx} & u_{xy} \\ u_{xy} & u_{yy} \end{bmatrix}.$$

Least-Squares Quadratic Reconstruction

Following the same procedure developed for linear reconstruction, a least-squares problem for the gradient and Hessian derivatives can be devised, see [Bar93] for full details. The result is the following nonsquare matrix problem

$$\begin{bmatrix} \vec{L}_1 & \vec{L}_2 & \vec{L}_3 & \vec{L}_4 & \vec{L}_5 \end{bmatrix} \begin{pmatrix} u_x \\ u_y \\ u_{xx} \\ u_{xy} \\ u_{yy} \end{pmatrix} = \overrightarrow{\Delta u}. \tag{5.3.14}$$

$$\begin{aligned}
\overrightarrow{\Delta u} &= [\Delta u_1, \Delta u_2, \ldots, \Delta u_n]^T \\
\vec{L}_1 &= [\Delta x_1, \Delta x_2, \ldots, \Delta x_n]^T \\
\vec{L}_2 &= [\Delta y_1, \Delta y_2, \ldots, \Delta y_n]^T \\
\vec{L}_3 &= \frac{1}{2}[\Delta x_1^2, \Delta x_2^2, \ldots, \Delta x_n^2]^T \\
\vec{L}_4 &= [\Delta x_1 \Delta y_1, \Delta x_2 \Delta y_2, \ldots, \Delta x_n \Delta y_n]^T \\
\vec{L}_5 &= \frac{1}{2}[\Delta y_1^2, \Delta y_2^2, \ldots, \Delta y_n^2]^T
\end{aligned}$$

where n is the number of quadrature points, u^{int} is the value of the reconstructed data at a quadrature point, and $\Delta u = u^{int} - u_0$. Note that rows of this matrix can be scaled by weighting factors without sacrificing the property of k-exactness. Multiplication by a matrix transpose produces the normal form of the least-squares problem:

$$\begin{bmatrix} L_{11} & L_{12} & L_{13} & L_{14} & L_{15} \\ L_{12} & L_{22} & L_{23} & L_{24} & L_{25} \\ L_{13} & L_{23} & L_{33} & L_{34} & L_{35} \\ L_{14} & L_{24} & L_{34} & L_{44} & L_{45} \\ L_{15} & L_{25} & L_{35} & L_{45} & L_{55} \end{bmatrix} \begin{pmatrix} u_x \\ u_y \\ u_{xx} \\ u_{xy} \\ u_{yy} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} \tag{5.3.15}$$

This equation requires the calculation of 15 inner products L_{ij} and 5 inner products f_i. Note that certain identities exist relating L_{ij} which further reduce the number of L_{ij} needed to 12.

Two concerns arise in the general implementation of this least-squares technique. First, we desire that the reconstruction be invariant to affine coordinate transformations, i.e. translation, rotation, dilatation, and shear. These effects can be eliminated or reduced by a preimage mapping to a normalized control volume. Second, the least-squares solution by way of normal equations can be poorly conditioned. When implementing quadratic reconstruction, we have noted some conditioning problems when the cell aspect ratio gets very large. Other methods for solving the basic least-squares problem are much less sensitive to the matrix condition number: Householder transforms, singular value decompositions, etc. For highly stretched meshes the use of these methods may be a necessity.

Enforcing Monotonicity of the Reconstruction

In [Bar93] we devised a limiting procedure similar to that used in linear reconstruction. Consider the reconstructed quadratic polynomial for the control volume Ω_0

$$u(x,y)_0 = u_0 + \mathbf{\Delta r}^T \nabla u_0 + \frac{1}{2} \mathbf{\Delta r}^T H_0 \mathbf{\Delta r}.$$

One approach to enforcing monotonicity of the reconstruction is to introduce a parameter Φ into the reconstruction polynomial

$$u(x,y)_0 = u_0 + \Phi_0 \left[\mathbf{\Delta r}^T \nabla u_0 + \frac{1}{2} \mathbf{\Delta r}^T H_0 \mathbf{\Delta r} \right] \tag{8.0}$$

with the goal of finding the largest admissible $\Phi_0 \in [0,1]$ while invoking a monotonicity principle that values of the reconstructed function must not exceed the maximum and minimum of neighboring nodal values and u_0. To calculate Φ_0 first compute

$$u_0^{max} = \max_{i \in \mathcal{N}_j} (u_0, u_i)$$

$$u_0^{min} = \min_{i \in \mathcal{N}_j} (u_0, u_i)$$

then require that $u_0^{min} \leq u(x,y)_0 \leq u_0^{max}$. Extrema in $u(x,y)_0$ can occur anywhere in the interior or on the boundary of the control volume surrounding v_0. Determining the location and type of extrema is clumsy and computationally expensive. One approximation which considerably simplifies the task is to interrogate the reconstructed polynomial for extreme values at the quadrature points used in the flux integration. For each quadrature point location in the flux integral compute the extrapolated state u_{0i}^L and determine the smallest Φ_0 so that

$$\Phi_0 = \begin{cases} \min(1, \frac{u_j^{max} - u_0}{u_{0i}^L - u_0}), & \text{if } u_{0i}^L - \overline{u}_0 > 0 \\ \min(1, \frac{u_j^{min} - u_0}{u_{0i}^L - u_0}), & \text{if } u_{0i}^L - u_0 < 0 \\ 1 & \text{if } u_{0i}^L - u_0 = 0 \end{cases} .$$

This limiting procedure is very effective in removing spurious solution oscillations although the discontinuous nature of the limiter can hinder steady-state convergence of the scheme.

To assess the merits of quadratic reconstruction, we consider a test problem which solves the two-dimensional scalar advection equation

$$u_t + (yu)_x - (xu)_y = 0$$

or equivalently

$$u_t + \vec{\lambda} \cdot \nabla u = 0, \quad \vec{\lambda} = (y, -x)^T$$

on a grid centered about the origin, see Fig. 5.4.1. Discontinuous inflow data is specified along an interior cut line, $u(x,0) = 1$ for $-.6 < x < -.3$ and $u(x,0) = 0$, otherwise. The exact solution is a solid body rotation of the cut line data throughout the domain.

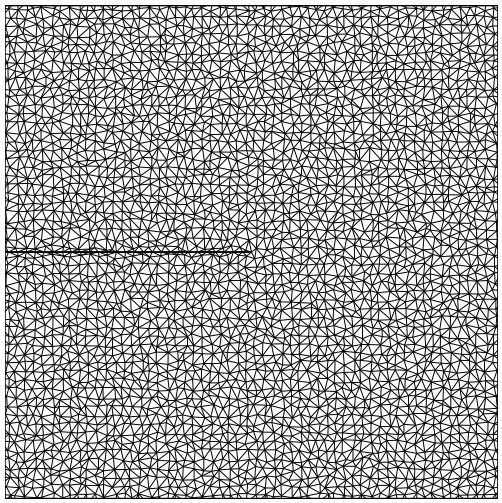

Figure 5.4.1 Grid for the circular advection problem.

The discontinuities admitted by this equation are similar to the linear contact and slip line solutions admitted by the Euler equations. Linear discontinuities are often more difficult to compute accurately than nonlinear shock wave solutions which naturally steepen due to converging characteristics. Figures 5.4.2 - 5.4.4 display solution contours for the schemes.

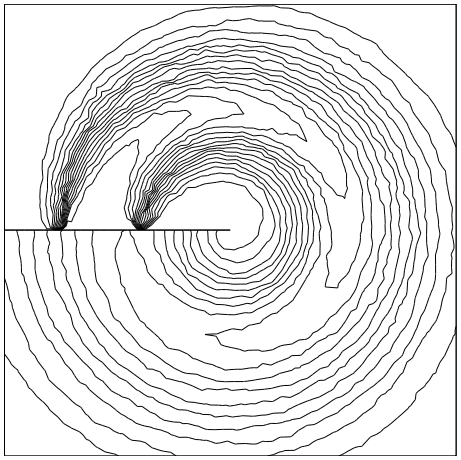

Figure 5.4.2 Solution contours, piecewise constant reconstruction.

The improvement from piecewise constant reconstruction to piecewise linear is quite dramatic. The improvement from piecewise linear to piecewise quadratic also looks impressive. The width of the discontinuities is substantially reduced with little observable grid dependence. Note however, that the quadratic approximation has roughly quadruple the number of solution unknowns because of the use of 6 noded triangles. In the Section 6 we will show computations of Euler flow using these same approximations.

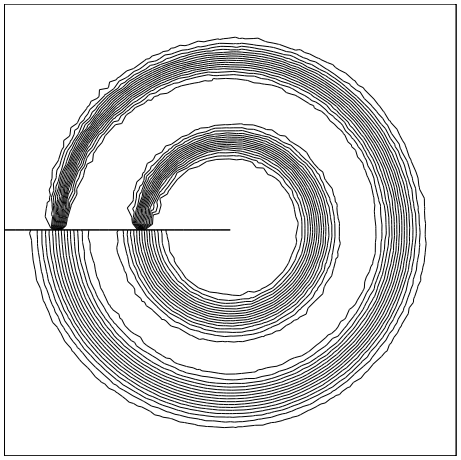

Figure 5.4.3 Solution contours, p.w. linear reconstruction.

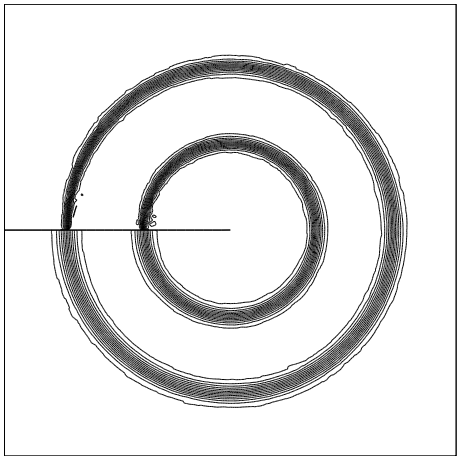

Figure 5.4.4 Solution contours, p.w. quadratic reconstruction.

5.5 k-Exact Reconstruction

In this section, a brief account is given of the reconstruction scheme presented in Barth and Frederickson [BarF90] for arbitrary order reconstruction. The task at hand is to determine reconstruction polynomials for each Ω_i. These polynomials are assumed to be of the form

$$U^k(x,y)_i = \sum_{m+n\leq k} \alpha_{(m,n)} P_{(m,n)}(x-x_c, y-y_c) \tag{5.5.0}$$

where $P_{(m,n)}(x-x_c, y-y_c) = (x-x_c)^m (y-y_c)^n$ and (x_c, y_c) is the control volume centroid. Upon first inspection, the use of high order reconstruction appears to be an expensive proposition. The present reconstruction strategy optimizes the efficiency of the reconstruction by precomputing as a *one time* preprocessing step the set of weights $\mathbf{W}_j$ in each cell Ω_j with neighbor set $\mathcal{N}_j$ such that

$$\alpha_{(m,n)} = \sum_{i\in\mathcal{N}_j} W_{(m,n)i} \overline{u}_i \tag{5.5.1}$$

where $\alpha_{(m,n)}$ are the polynomial coefficients to be used in (5.5.0). This effectively reduces the problem of reconstruction to multiplication of predetermined weights and cell averages to obtain polynomial coefficients.

During the preprocessing to obtain the reconstruction weights $\mathbf{W_j}$ a coordinate system with origin at the centroid of Ω_j is assumed to minimize roundoff errors. To insure that the reconstruction is invariant to affine transformations, we then temporarily transform (rotate and scale) to another coordinate system $(\overline{x}, \overline{y})$ which is normalized to the cell Ω_j

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

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

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

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

Polynomials on Ω_j are temporarily represented using the polynomial basis functions

$$\overline{P} = [1, \overline{x}, \overline{y}, \overline{x}^2, \overline{xy}, \overline{y}^2, \overline{x}^3, ...].$$

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

$$\overline{P}_{(m,n)} = \sum_{s+t\le k} G_{m,n}^{s,t} P_{(s,t)} \tag{5.5.2}$$

Satisfaction of conservation of the mean is guaranteed 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 $\overline{P}^0 = [1, \overline{x}, \overline{y}, \overline{x}^2 - 1, \overline{x}\,\overline{y}, \overline{y}^2 - 1, \overline{x}^3 - A_j(\overline{x}^3), ...]$. Note that using these polynomials requires a minor modification of (5.5.2) but retains the same form:

$$\overline{P}^0_{(m,n)} = \sum_{s+t\le k} \overline{G}_{m,n}^{s,t} P_{(s,t)} \tag{5.5.3}$$

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

Minimum Energy (Least-Squares) Reconstruction

We note that the set of control volume neighbors $\mathcal{N}_j$ must contain at least $(k+1)(k+2)/2$ members in two space dimensions if the reconstruction operator $\mathbf{R}_j^k$ is to be k-exact. That $(k+1)(k+2)/2$ members is not sufficient in all situations is easily observed. If, for example, the control volume-centers all lie on a single straight line one can find a linear function u such that $\mathbf{A}_j(u) = 0$ for every Ω_j, which means that reconstruction of u is impossible. In other cases a k-exact reconstruction operator $\mathbf{R}_j^k$ 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 $\mathbf{R}_j^k$ 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 $\mathbf{R}_j^k$ 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 minimizes 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 $\mathbf{W}_j$ for each control volume Ω_j which when applied via (5.5.1) produces piecewise polynomial approximations. We begin by first rewriting the piecewise polynomial for Ω_j in terms of the reconstruction weights

$$U^k(x,y) = \sum_{m+n\leq k} P_{(m,n)} \sum_{i\in\mathcal{N}_j} W_{(m,n)i}\overline{u}_i$$

or equivalently

$$U^k(x,y) = \sum_{i\in\mathcal{N}_j} \overline{u}_i \sum_{m+n\leq k} W_{(m,n)i}P_{(m,n)}$$

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

$$U^k(x,y) = \sum_{i\in\mathcal{N}_j} \overline{u}_i \sum_{m+n\leq k} W'_{(m,n)i}\overline{P}^0_{(m,n)} \tag{5.5.4}$$

Using (5.5.3), we can relate weights in the transformed system to weights in the original system

$$W_{(s,t)i} = \sum_{m+n\leq k} \overline{G}^{s,t}_{m,n} W'_{(m,n),i} \tag{5.5.5}$$

We satisfy k-exactness by requiring that (5.5.4) is satisfied for all linear combinations of the polynomials $\overline{P}^0_{(s,t)}(x,y)$ such that $s+t\leq k$. In particular, if $U^k(x,y) = \overline{P}^0_{(s,t)}(x,y)$ for some $s+t\leq k$ then

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

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

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

Transforming basis polynomials back to the original coordinate system we have

$$\sum_{i\in\mathcal{N}_j} W'_{(m,n)i} \sum_{u+v\leq k} \overline{G}^{u,v}_{s,t} A_i(P_{(u,v)}) = \delta^{st}_{mn} \tag{5.5.6}$$

This can be locally rewritten in matrix form as

$$\mathbf{W}'_j\mathbf{A}'_j = \mathbf{I} \tag{5.5.7}$$

and transformed in terms of the standard basis weights via

$$\mathbf{W}_j = \mathbf{G}\mathbf{W}'_j \tag{5.5.8}$$

Note that $\mathbf{W}'_j$ is a $(k+1)(k+2)/2$ by $\mathcal{N}_j$ matrix and $\mathbf{A}'_j$ has dimensions $\mathcal{N}_j$ by $(k+1)(k+2)/2$. To solve (5.5.7) in the optimum sense described above, an $\mathbf{L}_j\mathbf{Q}_j$ decomposition of $\mathbf{A}'_j$ is performed where the orthogonal matrix $\mathbf{Q}_j$ and the lower triangular matrix $\mathbf{L}_j$ have been constructed using a modified Gram-Schmidt algorithm (or a sequence of Householder reflections). The weights $\mathbf{W}'_j$ are then given by

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

Applying (5.5.5) these weights are transformed to the standard centroid basis and the preprocessing step is complete.

We now show a few results presented earlier in [BarF90]. The first calculation involves the reconstruction of a sixth order polynomial with random normalized coefficients which has been cell averaged onto a random mesh. Figures 5.5.0 shows a sample mesh and fig. 5.5.1 graphs the absolute L_2 error of the reconstruction for various meshes and reconstruction degree.

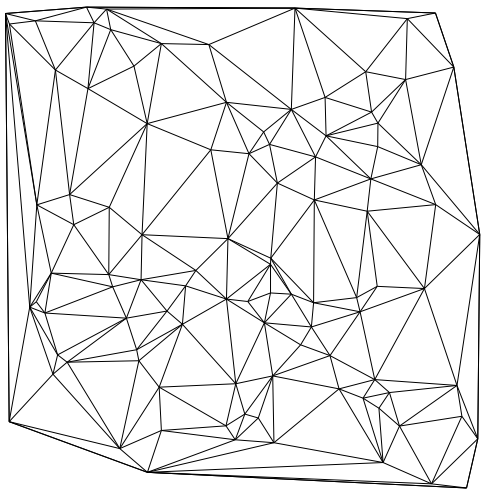

Figure 5.5.0 Random mesh.

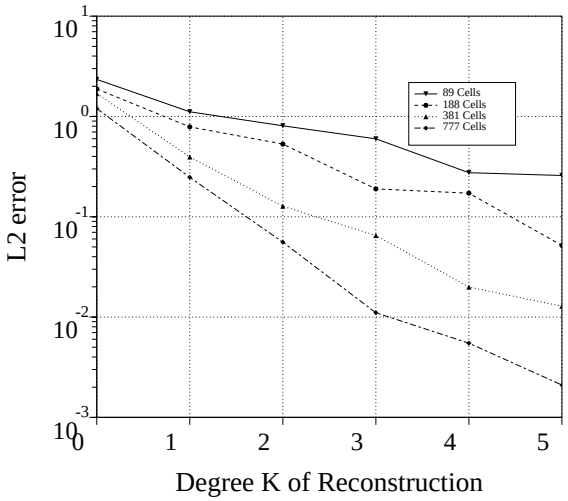

Figure 5.5.1 L_2 error of reconstruction.

The reconstruction algorithm has also been tested on more realistic problems. Figures 5.5.2-5.5.4 show a mesh and reconstructions (linear and quadratic) of a cell averaged density field corresponding to a Ringleb flow, an exact hodograph solution of the gasdynamic equations, see [Chio85].

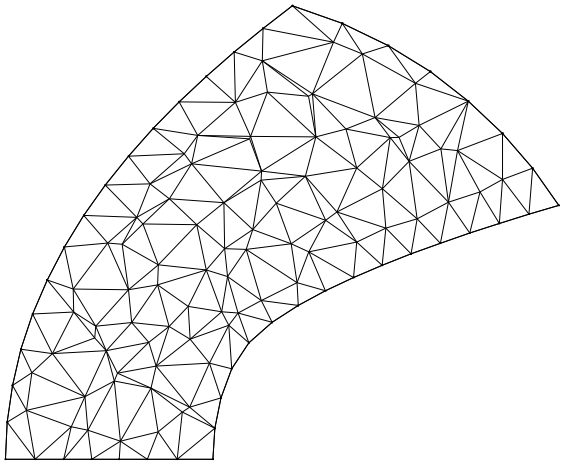

Figure 5.5.2 Randomized mesh for Ringleb flow.

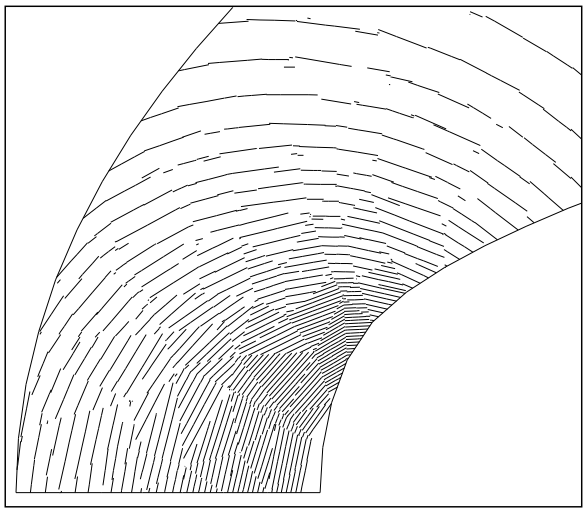

Figure 5.5.3 Piecewise linear reconstruction of Ringleb flow.

The reader should note that the use of piecewise contours gives a crude visual critique as to how well the solution is represented by the piecewise polynomials. The improvement from linear to quadratic is dramatic in the case of Ringleb flow. In the paper by Barth and Frederickson we show flow computations and error measures for the Ringleb problem.

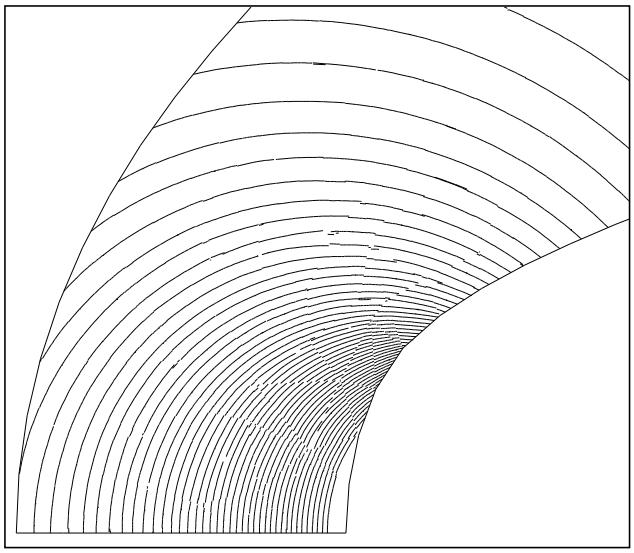

Figure 5.5.4 Piecewise quadratic reconstruction of Ringleb flow.

6.0 Finite-Volume Schemes for the Euler and Navier-Stokes Equations

In this section, we consider the extension of upwind scalar advection schemes to the Euler equations of gasdynamics. As we will see, the changes are relatively minor since most of the difficult work has already been done in designing the scalar scheme.

6.1 Euler Equations in Integral Form

The physical laws concerning the conservation of mass, momentum, and energy for an arbitrary region Ω can be written in the following integral form:

Conservation of Mass

$$\frac{d}{dt}\int_\Omega \rho \; da + \int_{\partial\Omega} \rho(\mathbf{V}\cdot\mathbf{n}) \; dl = 0 \tag{6.1.1}$$

Conservation of Momentum

$$\frac{d}{dt}\int_\Omega \rho\mathbf{V} \; da + \int_{\partial\Omega} \rho\mathbf{V}(\mathbf{V}\cdot\mathbf{n}) \; dl + \int_{\partial\Omega} p\mathbf{n} \; dl = 0 \tag{6.1.2}$$

Conservation of Energy

$$\frac{d}{dt}\int_\Omega E \; da + \int_{\partial\Omega} (E+p)(\mathbf{V}\cdot\mathbf{n}) \; dl = 0 \tag{6.1.3}$$

In these equations $\rho, \mathbf{V}, p$, and E are the density, velocity, pressure, and total energy of the fluid. The system is closed by introducing a thermodynamical equation of state for a perfect gas:

$$p = (\gamma - 1)(E - \frac{1}{2}\rho(\mathbf{V}\cdot\mathbf{V})) \tag{6.1.4}$$

These equations can be written in a more compact vector equation:

$$\frac{d}{dt}\int_\Omega \mathbf{u} \; da + \int_{\partial\Omega} \mathbf{F}(\mathbf{u})\cdot\mathbf{n} \; dl = 0 \tag{6.1.5}$$

with

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho\mathbf{V} \\ E \end{pmatrix}, \quad \mathbf{F}(\mathbf{u})\cdot\mathbf{n} = \begin{pmatrix} \rho(\mathbf{V}\cdot\mathbf{n}) \\ \rho\mathbf{V}(\mathbf{V}\cdot\mathbf{n}) + p\mathbf{n} \\ (E+p)(\mathbf{V}\cdot\mathbf{n}) \end{pmatrix}$$

In the next section, we show the natural extension of the scalar advection scheme to include (6.1.5).

6.2 Extension of Scalar Advection Schemes to Systems of Equations

The extension of the scalar advection schemes to the Euler equations requires three rather minor modifications:

(1) *Vector Flux Function.* The scalar flux function is replaced by a vector flux function. In the present work, the mean value linearization due to Roe [Roe81] is used. The form of this vector flux function is identical to the scalar flux function (4.2.33), i.e.

$$\begin{aligned}\mathbf{h}(\mathbf{u}^R, \mathbf{u}^L; \mathbf{n}) =& \frac{1}{2}\left(\mathbf{f}(\mathbf{u}^R, \mathbf{n}) + \mathbf{f}(\mathbf{u}^L; \mathbf{n})\right) \\ & -\frac{1}{2}|A(\mathbf{u}^R, \mathbf{u}^L; \mathbf{n})|\left(\mathbf{u}^R - \mathbf{u}^L\right)\end{aligned} \tag{5.6}$$

where $\mathbf{f}(\mathbf{u}; \mathbf{n}) = \mathbf{F}(\mathbf{u}) \cdot \mathbf{n}$, and $A = d\mathbf{f}/d\mathbf{u}$ is the flux Jacobian.

(2) *Componentwise limiting.* The solution variables are reconstructed componentwise. In principle, any set of variables can be used in the reconstruction (primitive variables, entropy variables, etc.). Note that conservation of the mean can make certain variable combinations more difficult to implement than others because of the nonlinearities that may be introduced. The simplest choice is obviously the conserved variables themselves. When conservation of the mean is not important (steady-state calculations), we typically use primitive variables in the reconstruction step.

(3) *Weak Boundary Conditions.* Boundary conditions for inviscid flow at solid surfaces are enforced weakly. For solid wall boundary edges, the flux is calculated with $\mathbf{V} \cdot \mathbf{n}$ set identically to zero

$$\mathbf{f}(\mathbf{u}; \mathbf{n}) = \begin{pmatrix} 0 \\ n_x p \\ n_y p \\ 0 \end{pmatrix}.$$

Boundary conditions at far field boundaries are also done weakly. Define the characteristic projectors of the flux Jacobian A in the following way:

$$P^{\pm} = \frac{1}{2}[I \pm \text{sign}(A)].$$

At far field boundary edges the fluxes are assumed to be of the form:

$$\mathbf{f}(\mathbf{u}^n; \mathbf{n}) = (\mathbf{F}(\mathbf{u}^n_{proj}) \cdot \mathbf{n})$$

where $\mathbf{u}^n_{proj} = P^+ \ \mathbf{u}^n + P^- \ \mathbf{u}_\infty$ and $\mathbf{u}_\infty$ represents a vector of prescribed far field solution values. At first glance, prescribing the entire vector $\mathbf{u}_\infty$ is an overspecification of boundary conditions. Fortunately the characteristic projectors remove or ignore certain combinations of data so that the correct number of conditions are specified at inflow and outflow.

The remainder of this section will present calculations of Euler flow using various reconstruction schemes and mesh adaptation. The resulting scheme for the Euler equations has essentially the same shock resolving characteristics as the scalar scheme. The actual solution strategy is based on an implicit Newton-like solver. Details of the implicit scheme are discussed in Section 7.

Transonic Airfoil Flow

Figures 6.2.0a-b show a simple Steiner triangulation and the resulting solution obtained with a linear reconstruction scheme for transonic Euler flow ($M_\infty = .80, \alpha = 1.25°$) over a NACA 0012 airfoil section.

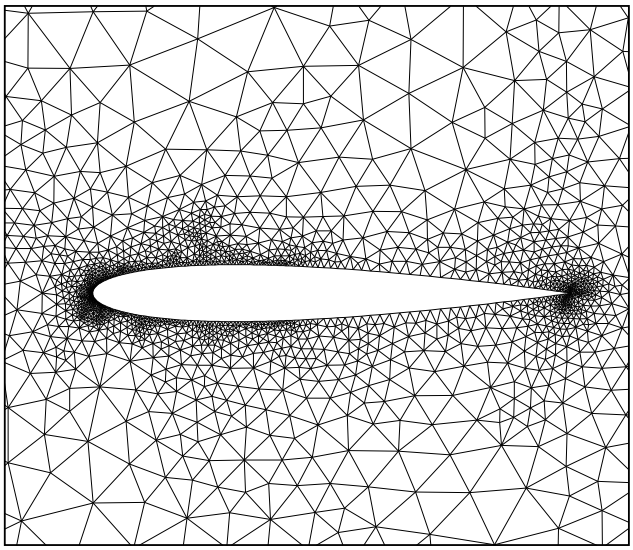

Figure 6.2.0a Initial triangulation of airfoil, 3155 vertices.

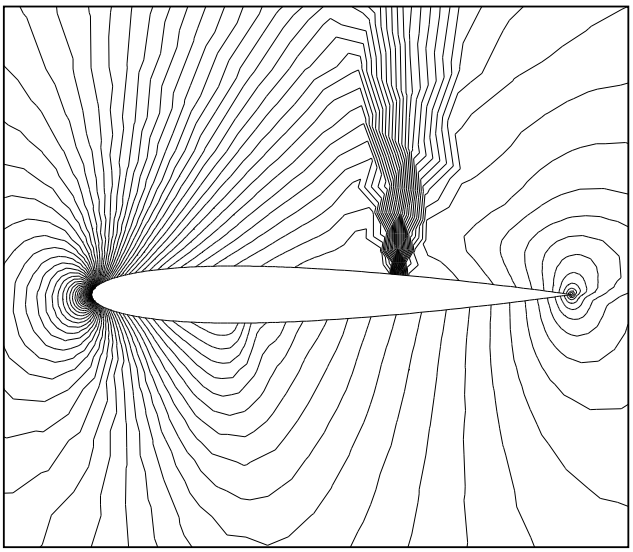

Figure 6.2.0b Mach number contours on initial triangulation, $M_\infty = .80, \alpha = 1.25°$.

Even though the grid is very coarse with only 3155 vertices, the upper surface shock is captured cleanly with a profile that extends over two cells of the mesh. Clearly, the power of the unstructured grid method is the ability to locally adapt the mesh to resolve flow features. Figures 6.2.1a-b show an adaptively refined mesh and solution for the same flow.

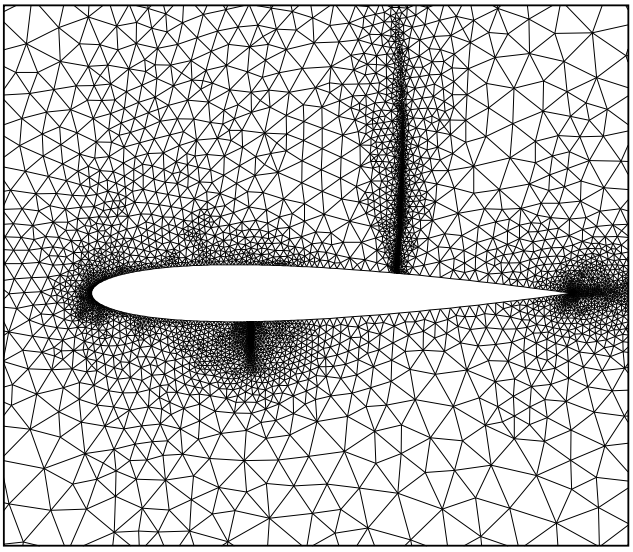

Figure 6.2.1a Solution adaptive triangulation of airfoil, 6917 vertices.

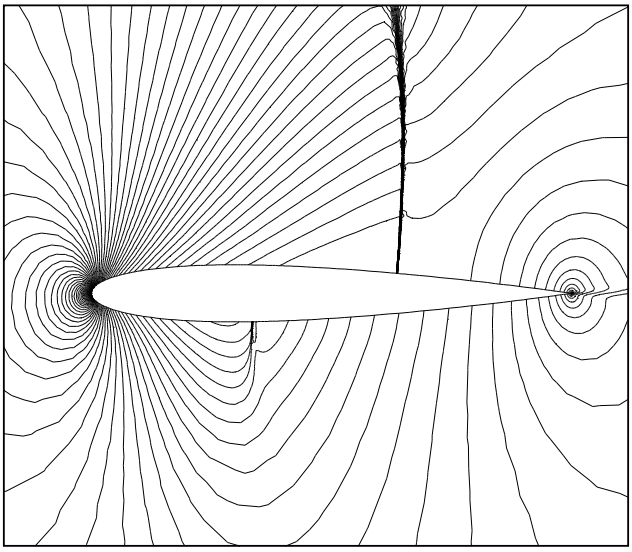

Figure 6.2.1b Mach number solution contours on adapted airfoil.

The mesh has been locally refined based on *a posteriori* error estimates. These estimates were obtained by performing k-exact reconstruction in each control volume using linear and quadratic functions and comparing the difference. The flow features in fig. 6.2.1b are clearly defined with a weak lower surface shock now visible. Figure 6.2.1c shows the surface pressure coefficient distribution on the airfoil. The discontinuities are monotonically captured by the scheme.

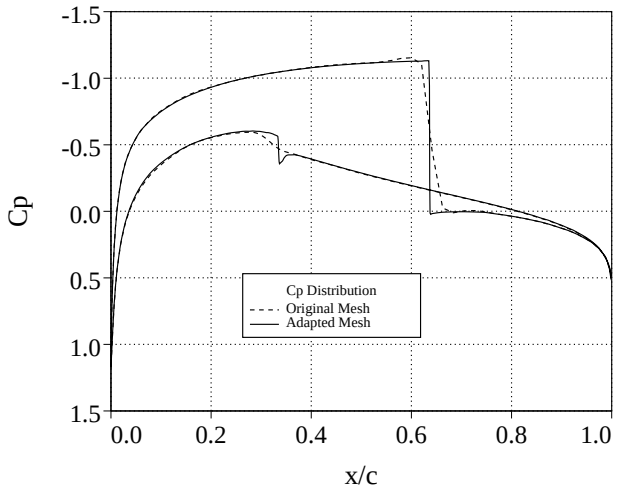

Figure 6.2.1c Comparison of C_p distributions on initial and adapted meshes.

Inviscid Multi-Element Airfoil Flow

As we have see from Section 2, one major advantage of unstructured grids is the ability to automatically mesh complex geometries. The next example shown in figure 6.2.2a-b is a Steiner triangulation and solution about a multi-element airfoil.

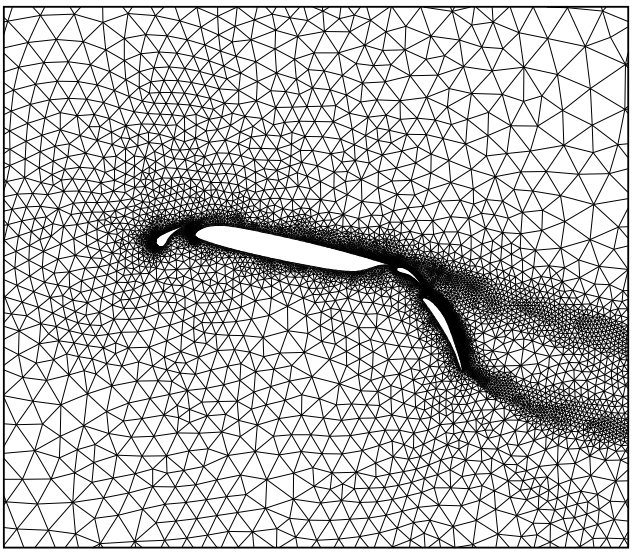

Figure 6.2.2a Steiner triangulation about multi-element airfoil.

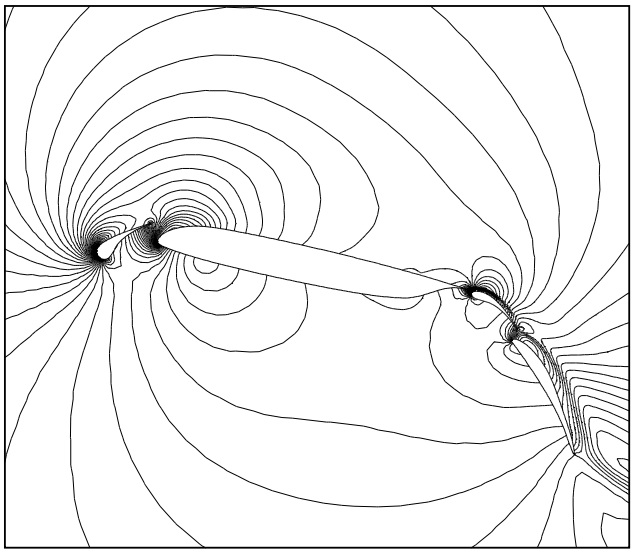

Figure 6.2.2b Mach number contours about multi-element airfoil, $M_\infty = .2, \alpha = 0°$.

Using the incremental Steiner algorithm discussed previously, the grid can constructed from curve data in about ten minutes time on a standard engineering workstation using less than a minute of actual CPU time. The flow calculation shown in figure 6.2.2b was performed on a CRAY supercomputer taking just a few minutes of CPU time using a linear reconstruction scheme with implicit time advancement. Details of the implicit scheme are given in the next section.

Supersonic Oblique Shock Reflections

In this example, two supersonic streams (M=2.50 and M=2.31) are introduced at the left boundary. These streams interact producing a pattern of supersonic shock reflections down the length of the converging channel, see Fig. 6.2.3. The grid is a subdivided 15x52 mesh with perturbed coordinates.

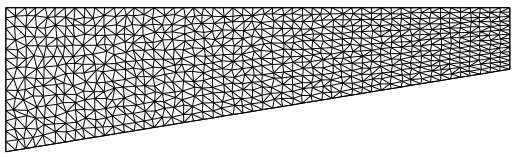

Figure 6.2.3a Channel grid (780 Verts).

Solution Mach contours are shown in Figs. 6.2.3b-d. Figure 6.2.5 graphs density profiles along a horizontal cut at 70 percent the vertical height of the left boundary. As expected, the piecewise constant reconstruction scheme severely smears the shock system while the scheme based on a linear solution reconstruction, fig. 6.2.4c, performs very well. The piecewise quadratic approximation, fig. 6.2.4d, shows some improvement in shock wave

thickness although the improvement is not dramatic given the increased number of unknowns in the quadratic element. The number of unknowns required for the quadratic approximation is roughly four times the number required for the piecewise linear scheme. The less than dramatic improvement is not a surprising result since the solution has large regions of constant flow which do not benefit greatly from the quadratic approximation. At solution discontinuities the quadratic scheme reduces to a low order approximation which again negates the benefit of the quadratic reconstruction.

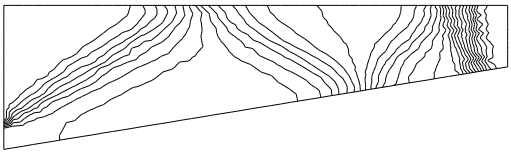

Figure 6.2.3b Mach contours, piecewise constant reconstruction.

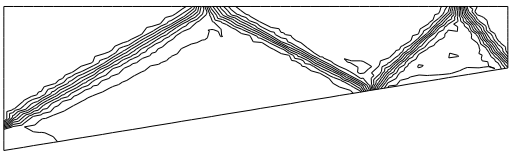

Figure 6.2.3c Solution Mach contours, piecewise linear reconstruction.

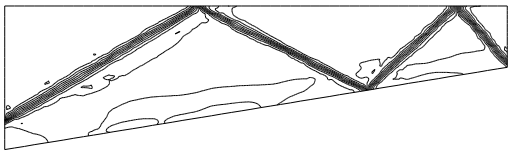

Figure 6.2.3d Solution Mach contours, piecewise quadratic reconstruction.

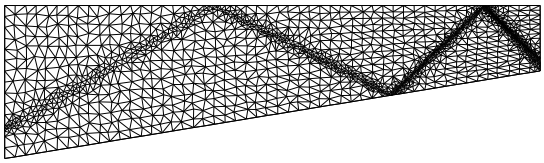

Figure 6.2.4a Adapted channel grid (1675 Verts).

Figure 6.2.4a shows the same mesh adaptively refined. The number of mesh points has roughly doubled. The solution shown in fig 6.2.4b was calculated using linear reconstruction. The results are very comparable with the calculation performed using quadratic reconstruction.

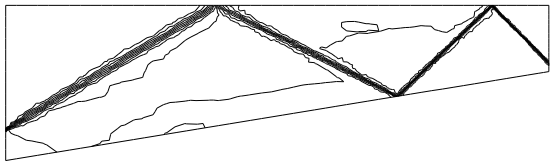

Figure 6.2.4b Mach contours on adapted mesh, piecewise linear reconstruction.

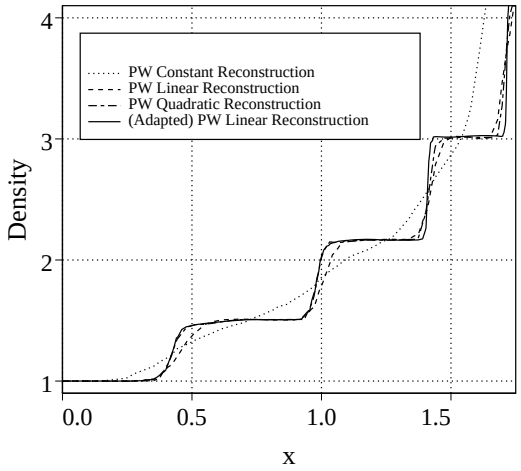

Figure 6.2.5 Density profiles, $y/h = .70$.

ONERA M6 Wing

The algorithms outlined in Sections 5-6 have been extended to the Euler equations in three dimensions. In [Bar91], we showed the natural extension of the edge data structure in the development of an Euler equation solver on tetrahedral meshes. One of the calculations presented in this paper simulated Euler flow about the ONERA M6 wing. The tetrahedral mesh used for the calculations was a subdivided 151x17x33 hexahedral C-type mesh with spherical wing tip cap. The resulting tetrahedral mesh contained 496,350 tetrahedra, 105,177 vertices, 11,690 boundary vertices, and 23,376 boundary faces. Figure 6.2.7 shows a closeup of the surface mesh near the outboard tip.

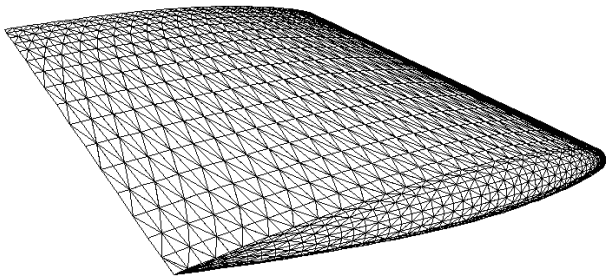

Figure 6.2.7 Closeup of M6 Wing Surface Mesh Near Tip.

Transonic calculations, $M_\infty = .84, \alpha = 3.06°$, were performed on the CRAY Y-MP computer using the upwind code with both the Green-Gauss and L_2 gradient reconstruction. Figure 6.2.8 shows surface pressure contours on the wing surface and C_p profiles at several span stations.

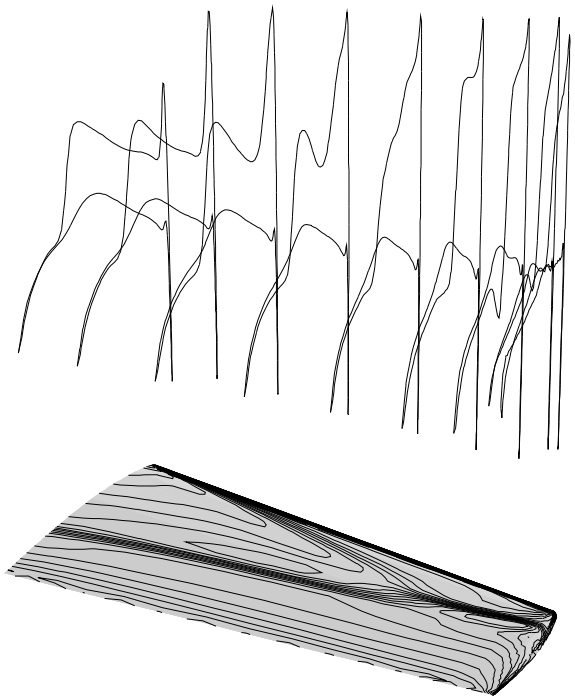

Figure 6.2.8 M6 Wing Surface Pressure Contours and Spanwise C_p Profiles ($M_\infty = .84, \alpha = 3.06°$).

Pressure contours clearly show the lambda type shock pattern on the wing surface. Figures 5.8a-c compare pressure coefficient distributions at three span stations on the wing measured in the experiment, y/b=.44,.65.,.95.

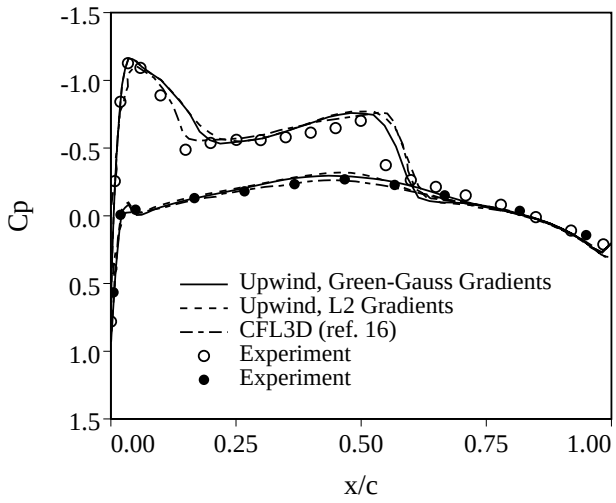

Figure 6.2.9a M6 Wing Spanwise Pressure Distribution, $y/b = .44$.

Each graph compares the upwind code with Green-Gauss and L_2 gradient calculation with the CFL3D results appearing in Thomas et al. [ThomLW85] and the experimental data reported by Schmitt [Schmitt79]. Numerical results on the tetrahedral mesh compare very favorably with the CFL3D structured mesh code. The results for the outboard station appear better for the present code than the CFL3D results. This is largely due to the difference in grid topology and subsequent improved resolution in that area.

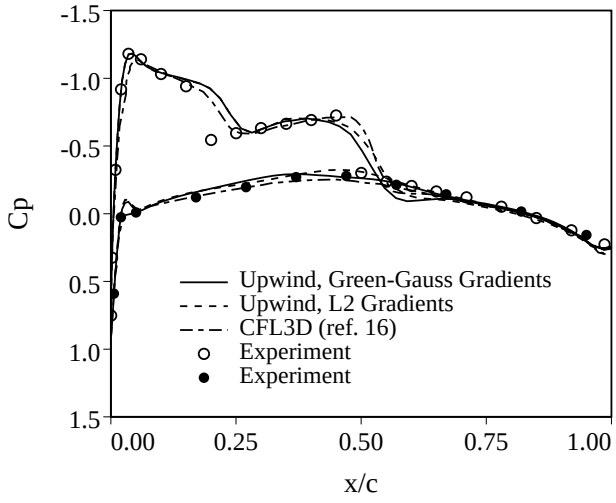

Figure 6.2.9b M6 Wing Spanwise Pressure Distribution $y/b = .65$.

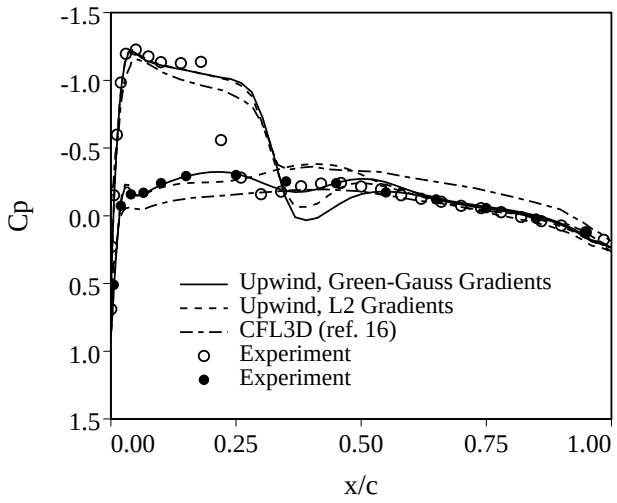

Figure 6.2.9c M6 Wing Spanwise Pressure Distribution $y/b = .95$.

6.3 Calculation of Navier-Stokes Flow

6.3a Maximum Principles

Section 4.1 discusses the maximum principle properties of Laplace's equation on structured and unstructured meshes. The maximum principle analysis entends naturally to the self-adjoint viscous-like term

$$\nabla \cdot \mu(x,y)\nabla u, \ \ \mu \geq 0.$$

Assuming a linear solution variation in each element, both the finite-element and finite-volume methods on planar triangulations yield the following result

$$\int_{\Omega_0} w(\nabla \cdot \mu \nabla u)\, d\,\Omega \approx \sum_{i \in \mathcal{N}_0} W_{0i}(U_i - U_0) \tag{6.3.0}$$

$$W_{0i} = \frac{1}{2}\left[\overline{\mu}^L \text{cotan}(\alpha^L) + \overline{\mu}^R \text{cotan}(\alpha^R)\right]_{0i}$$

where α^L and α^R are the left and right angles subtending the edge $e(v_0, v_i)$. Similarly, $\overline{\mu}^L$ and $\overline{\mu}^R$ are integral averaged values of $\mu(x,y)$ in the left and right adjacent triangles. Thus we immediately arrive at sufficient conditions for a maximum principle:
Consider discretizations of $\nabla \cdot \mu(x,y)\nabla u$ using a finite-volume or Galerkin finite-element method with linear elements. A sufficient condition for the discretized scheme to exhibit a discrete maximum principle is that all angles in the triangulation be acute.
The proof follows immediately from nonnegativity of weights appearing in (6.3.0). In Section 4.1 we saw that for planar triangulations a discrete maximum principle was assured

if the triangulation of the point set was a Delaunay triangulation. This result does not extend to (6.3.0).

While the existence of a discrete maximum principle simplifies the proofs for stability and uniform convergence of numerical approximations, the existence of a discrete maximum principle for self-adjoint equations is more of a luxury than a necessity. For example the L_2 finite-element theory still yields a stability estimate for the above equation even when the maximum principle analysis does not hold. In fact, the variational operator $a(u, w)$ associated with $\nabla \cdot \mu(x, y)\nabla u$

$$a(u, w) = \int_\Omega \mu(\nabla u \cdot \nabla w) d\,\Omega$$

serves as a proper energy norm $\|u\|_a^2 = a(u, u)$ for proving convergence of the method.

6.3b Alternative Tessellations

The computation of viscous flow on stretched triangulation places high demands on the discretization. Recall that the spatial accuracy of the generalized Godunov scheme relies heavily on the property of k-exactness, i.e. that certain complete polynomials are reconstructed exactly whenever the numerical solution has that polynomial dependence. If this is the case then the reconstructed polynomials are continuous at interface boundaries. From consistency of the numerical flux function we have

$$\mathbf{h}(\mathbf{u}, \mathbf{u}; \mathbf{n}) = (\mathbf{F}(\mathbf{u}) \cdot \mathbf{n})$$

so that the numerical flux function collapses to the true Euler flux. If we assume an exact flux integration in space, then *all* component calculations related to the spatial operator are done exactly. This implies that schemes that possess k-exactness for some value of k have artificial viscosity which vanishes completely when the true solution behaves as a k-th order polynomial.

Even so, the actual choice of control volume shape can significantly effect the absolute level of discretization error. Assume that the control volumes are formed from a geometric dual of a triangulation. A number of possible geometric duals exist. The most frequent choice is the median dual. It is known that the Galerkin finite-element method with linear elements can be rewritten as a finite-volume scheme on a median dual tessellation. The median dual is also attractive because it is well defined for any triangulation and has nice geometrical relationships which simplify many discretization formulas. Unfortunately, the median dual regions can become very distorted when triangles obtain high aspect ratio, see fig. 6.3.0a.

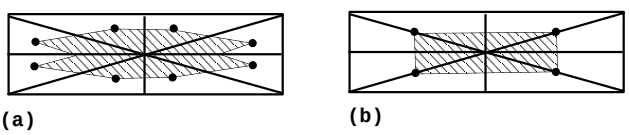

Figure 6.3.0 (a) Median Dual Tessellation, (b) Containment Circle (Sphere) Tessellation for Stretched Triangulations.

The Riemann problems associated with 6.3.0a become very nonphysical. The net result is a degradation in the accuracy of the schemes discussed in Sections 4-6. One interesting alternative which avoids this problem to a large extent is the containment circle tessellation shown in fig. 6.3.0b. The containment circle is defined as the smallest circle which contains a triangle. For acute triangles the minimum containment circle is the circumcircle, see fig. 6.3.1.

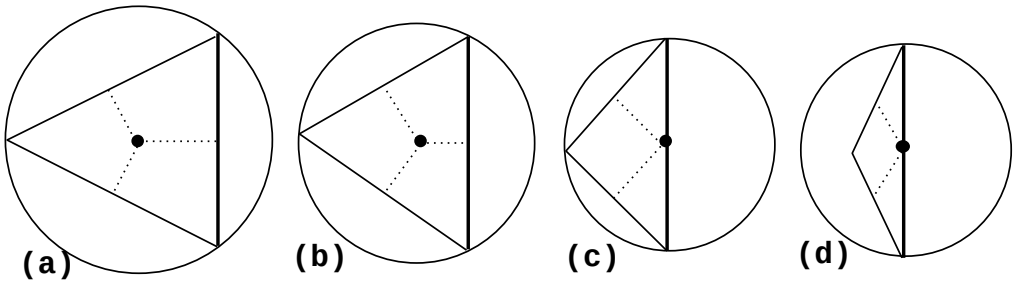

Figure 6.3.1 Containment circle centers for acute and obtuse triangules.

For an obtuse triangle the minimum containment circle has a center which lies on the longest edge in the triangle. The geometric dual is obtained by connecting containment circle centers to the midside of edges for a triangle. This construction produces portions of the Voronoi diagram for acute triangles but does *not* for obtuse triangles. Nicolaides and coworkers [XiaN92] have extensively studied properties of the divergence and curl operators on Voronoi duals. Note that connecting circumcenters only make sense for Delaunay triangulation, see fig. 6.3.2a. On the other hand, the notion of a containment circles remains well defined for all triangulations.

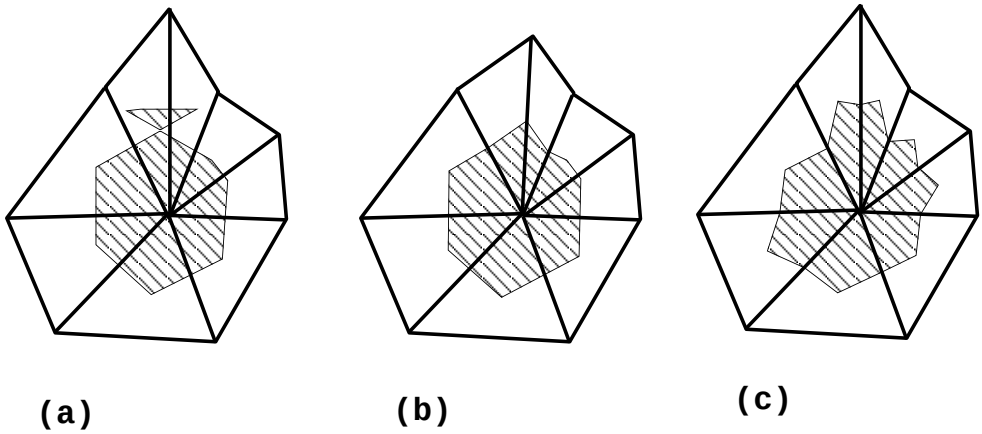

Figure 6.3.2 (a) Dual Obtained by Connecting Circumcenters, (b) Containment Circle Centers and Midsides, (c) Centroids and Midsides.

In fig. 6.3.3a we show a subdivided quadrilateral mesh with stretched elements near

the surface of the airfoil. Next we calculate subsonic *inviscid* compressible flow on this mesh using the scheme presented in the previous section using linear reconstruction and the median dual tessellation shown in fig. 6.3.3b. The calculation of inviscid flow on viscous-like stretched meshes provides an excellent test for the numerical methods.

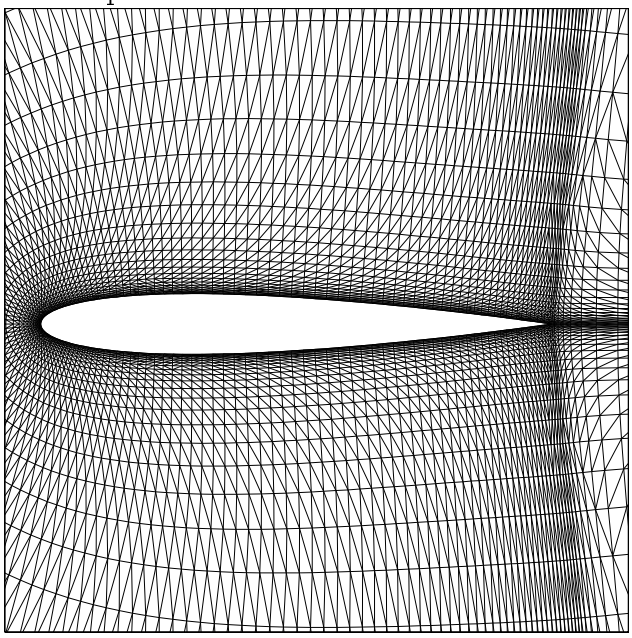

Figure 6.3.3a Triangulation About NACA 0012 Airfoil Obtained by Subdividing a Quadrilateral Mesh.

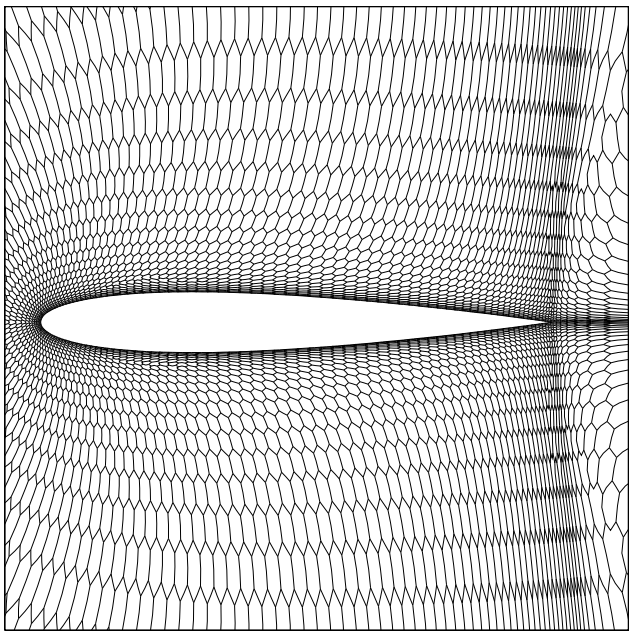

Figure 6.3.3b Median Dual Tessellation of the Triangulation Shown in fig. 6.3.3a.

Mach number contours of the solution are shown in fig. 6.3.3c. The correct solution has Mach contours which smoothly approach the surface of the airfoil. The present solu-

tion shows a noticeable inaccuracy near the surface of the airfoil with significant entropy production and ultimately several counts of airfoil drag.

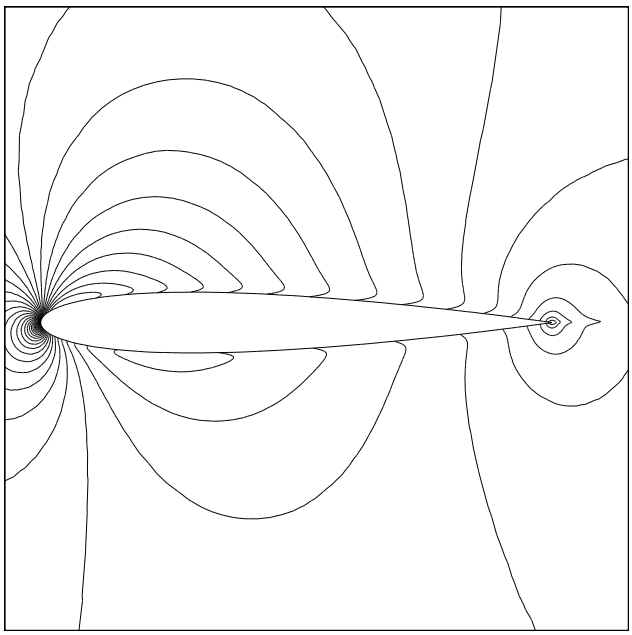

Figure 6.3.3c Mach Contours for Inviscid Subsonic Flow Over NACA 0012 ($M_\infty = .50$, $\alpha = 2.0°$) Using Median Dual Tessellation Showing Inaccuracy of the Computation Near the Airfoil Surface.

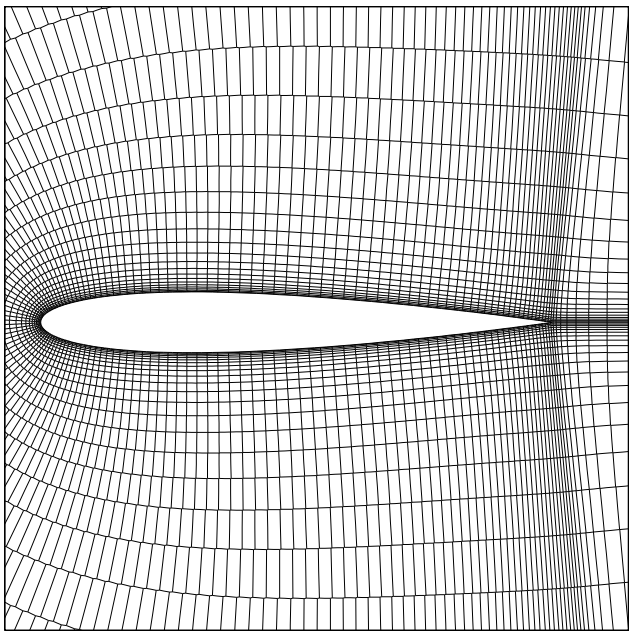

Figure 6.3.3d Containment Sphere Tessellation Derived From Figure 6.3.3a by Connecting Containment Sphere Centers and Edge Midsides.

Next we repeat the calculation using the same algorithm with linear reconstruction but a containment circle tessellation as shown in fig. 6.3.3d. The solution obtained on this

mesh has improved qualitative features. The peak entropy produced is reduced by almost a factor of 10 and the drag level is reduced to near zero.

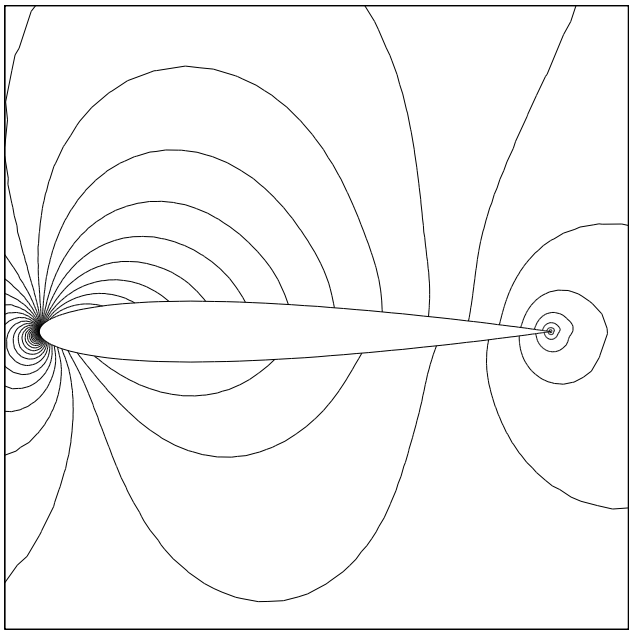

Figure 6.3.3e Mach Contours for Inviscid Subsonic Flow Over NACA 0012 ($M_\infty = .50$, $\alpha = 2.0°$) Using Containment Sphere Tessellation Showing Improved Accuracy of the Computation Near the Airfoil Surface.

6.3c The Navier-Stokes Equations

The development of efficient and robust solution strategies for solving the Navier-Stokes equations on unstructured meshes is probably one of the most difficult aspects facing algorithm developers. In this section we will show several steady-state Navier-Stokes calculations performed on stretched triangulations. The preferred method of reconstruction for these calculations is the least-squares method with unit weights. In solving high Reynolds number flows we must also content with the modeling of fluid turbulence. Two simple transport models suitable for unstructured meshes are presented. We will delay the discussion of the implicit solution strategy until Section 7. At that time we will also discussion other acceleration procedures for steady-state calculations.

We begin with the Navier-Stokes equations in integral form

$$\frac{d}{dt}\int_\Omega \mathbf{u}\, d\Omega + \int_{\partial\Omega} (\mathbf{F}\cdot\mathbf{n}) d\Gamma = \int_{\partial\Omega} (\mathbf{G}\cdot\mathbf{n}) d\Gamma \tag{6.3.1}$$

where $\mathbf{u}$ represents the vector of conserved variables, $\mathbf{F}$ the inviscid Euler flux vector,

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix} \hat{\mathbf{i}} + \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix} \hat{\mathbf{j}}$$