

NASA Contractor Report 194997

ICASE Report No. 94-89

IN-64  
34798  
24P



# ICASE

## FLOATING SHOCK FITTING VIA LAGRANGIAN ADAPTIVE MESHES

**John Van Rosendale**

(NASA-CR-194997) FLOATING SHOCK  
FITTING VIA LAGRANGIAN ADAPTIVE  
MESHES Final Report (ICASE) 24 p

N95-18110

Unclas

G3/64 0034998

Contract NAS1-19480  
November 1994

Institute for Computer Applications in Science and Engineering  
NASA Langley Research Center  
Hampton, VA 23681-0001



Operated by Universities Space Research Association



# Floating Shock Fitting via Lagrangian Adaptive Meshes\*

*John Van Rosendale*

Institute for Computer Applications in Science and Engineering  
NASA Langley Research Center  
Hampton, VA 23681

## Abstract

In recent work we have formulated a new approach to compressible flow simulation, combining the advantages of shock-fitting and shock-capturing. Using a cell-centered Roe scheme discretization on unstructured meshes, we warp the mesh while marching to steady state, so that mesh edges align with shocks and other discontinuities. This new algorithm, the Shock-fitting Lagrangian Adaptive Method (SLAM) is, in effect, a reliable shock-capturing algorithm which yields shock-fitted accuracy at convergence.

Shock-capturing algorithms like this, which warp the mesh to yield shock-fitted accuracy, are new and relatively untried. However, their potential is clear. In the context of sonic booms, accurate calculation of near-field sonic boom signatures is critical to the design of the High Speed Civil Transport (HSCT). SLAM should allow computation of accurate N-wave pressure signatures on comparatively coarse meshes, significantly enhancing our ability to design low-boom configurations for high speed aircraft.

---

\*This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480, while the author was in residence at ICASE, NASA Langley Research Center, Hampton, VA 23681.



# 1 Introduction

One of the principal difficulties in computing compressible flows is that such flows are generally only piecewise smooth. The solutions are smooth, except along a sequence of arcs or surfaces at which the solution or its derivatives have jump discontinuities. In the vicinity of these discontinuities, difference approximations are problematic. Moreover, errors at shocks can contaminate the solution everywhere.

There are two basic approaches to computation of compressible flows, shock-capturing and shock-fitting. Shock-capturing, in which one applies a well chosen difference scheme throughout the flow field, is effective and reliable, but is usually only first order accurate near shocks. Such schemes smear shocks over several mesh cells, limiting the accuracy and resolution obtainable.

The alternative is shock-fitting, in which the shocks are treated as internal boundaries in the flow across which one applies the Rankine-Hugoniot jump conditions. Shock-fitting algorithms can achieve an arbitrarily high order of accuracy, though properly locating shocks is difficult, especially for flows containing complex embedded shocks.

In recent work we have formulated a new approach to compressible flow simulation, combining the advantages of shock-fitting and shock-capturing. The fundamental difficulty in shock-fitting has always been that of unambiguously detecting and locating shocks. In simple cases, such as that of a strong bow shock, one has enough *a priori* knowledge of the shock location that fitting schemes are highly successful. However, in more complex situations, shock-fitting becomes difficult and unreliable. For this reason, given the simplicity and effectiveness of modern shock-capturing schemes, the latter have come to dominate computational aerodynamics, despite their limited resolution.

The new approach we are exploring begins with a cell-centered Roe discretization, on unstructured meshes [1]. Roe's scheme is a popular and effective method, which has an interesting property: at steady state, this scheme imposes the exact Rankine-Hugoniot jump conditions on any cell face which is oriented to and lying along a shock or other flow discontinuity. Thus if we warp the mesh while marching to steady state, so that shocks and other discontinuities lie along cell faces, Roe's scheme will give virtually exact answers there. This is the basic idea of the Shock-fitting Lagrangian Adaptive Method. SLAM is, in effect, a reliable shock-capturing algorithm, which incidentally yields shock-fitted solutions at convergence, with the attendant improvement in accuracy and resolution. Our algorithm is closely related to two other methods presented recently [2], [3]. We give the precise similarities and differences in the next section.

## 2 Related Work

While shock-fitting has existed for decades [4, 5, 6], the idea of warping an unstructured grid in a shock-capturing code to effectively fit shocks is new. The basic idea of using a conservative shock-capturing scheme on unstructured meshes, and warping the mesh to fit shocks during iteration to convergence, was independently developed by several groups, including ourselves.

All three of the groups we are aware of used algorithms based on Roe-scheme discretiza-

tions, but beyond that the details of these approaches differ. Parpia and Parikh [3] use the waves occurring in a six-wave multidimensional Riemann solver to align mesh edges with shocks, without actually fitting shocks. The multidimensional Riemann solver, due originally to Roe, is described in references [7, 8]. Aligning the grid allows them to achieve true “one-point” shocks, free of the “splitting error” that occurs when shocks cross the mesh at an oblique angle. The other group, Trepanier et al. also used the six-wave multidimensional Riemann solver to control mesh warping [2, 9]. However, unlike Parpia and Parikh, they also move mesh edges to coincide with shocks, thus obtaining shock-fitting accuracy in the final solution.

Our algorithm is similar to that of Trepanier et al., differing in that we warp the mesh using only density gradients, rather than using the waves occurring in a multidimensional Riemann solver. Thus unlike these other groups, we do not need a separate discretization to control mesh movement. In effect, we are reusing information from the Roe scheme discretization. Our approach is clearly simpler and cheaper. Also, extension to 3D is straight forward with our approach. The other approaches should extend to 3D as well, though how well they extend is not clear multidimensional Riemann solvers are not nearly as robust and effective in 3D as in 2D. Design of better 3D multidimensional Riemann solvers is an area of active research [10].

### 3 Algorithm Design

The discretization used here is the cell centered Roe scheme. This is an effective and heavily used scheme, whose occasional failings are now well understood and easily overcome [11]. In our algorithm we march to steady state using the Roe scheme coupled to a “locally implicit” time stepping scheme [1]. For the first 30 or 40 iterations, we keep the mesh frozen, allowing initial transients to dissipate. After that, we allow the mesh to warp at each time step to fit the developing shocks.

#### Scheme A

Several schemes for warping the mesh have been tried. The two found to work best are described here. The first of these, which we will call “scheme A” for convenience, is based on reconstruction of 1D shocks on the edges of the mesh. We begin by interpolating the cell-centered densities to edges and vertices. Edge values are obtained by simple averaging of adjacent cell values, while vertex values are obtained by weighting the density value in each cell by the angle subtended by that cell. That is, if triangles  $\{T\}_{i=1}^k$  touch vertex  $v$ , the angle-weighted density at  $v$  is

$$\rho_v = \frac{\sum_T \theta_{T,v} \rho_T}{\sum_T \theta_{T,v}}$$

where  $\theta_{T,v}$  is the angle subtended by triangle  $T$  at vertex  $v$ . The denominator here is 360 degrees, except at boundary points.

“Angle-weighting” is crucial to our scheme, since it means that vertex values along a fitted straight shock separating states  $U_l$  and  $U_r$  will be correct, independent of the local

mesh topology –  $U_l$  and  $U_r$  will both be weighted by 180 degrees. While angle-weighting does produce second order errors along curved shocks, this has not been found to be a problem.

Now given density values on all edges and vertices, we reconstruct a shock on every edge. Assume the density profile along each edge is piecewise constant, with a single jump. At the end of each edge, the density must match that of the vertex, while the jump location is determined by requiring that the average of the reconstructed piecewise constant density match the known value on this edge, if possible, as shown in Figure 1.

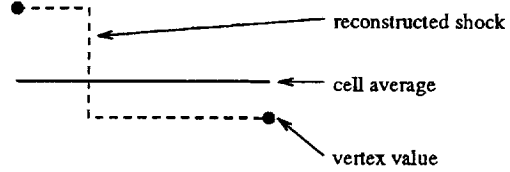


Figure 1: Reconstruction of a Shock Along an Edge

Once we have reconstructed shocks on the edges of the mesh, we attempt to attract each vertex to the shocks on the edges incident on that vertex. To do this, we compute the center of mass of these shocks, assuming the mass of each shock is proportional to its strength. This center of mass is a point in the plane, to which we attempt to attract this vertex. Note that we are attracting only vertices to shocks. There is no direct consideration of either attracting or orienting edges. In effect, we are making use of the following principle:

*When two vertices of a triangle (or three of a tetrahedron) lie on a straight shock, the intervening cell face exactly fits the shock.*

## Scheme B

Scheme A is effective, but does not work well for very weak shocks. Scheme B remedies this. It is composed of two separate components:

1. A shock detector
2. A vertex attraction force

The latter is used only at points detected as shocks.

The shock detector uses density gradients at the vertices, computed, for example, by Green's theorem path integrals. Let  $g_v^n$  denote the density gradient at vertex  $v$  and time step  $n$ . For all neighboring vertices,  $a$  of  $v$ , define a weight

$$w_a^n = (a - v) \cdot g_v^n$$

where  $\cdot$  denotes the normal inner product. Then we take the weighted average of gradient norms, with respect to these weights:

$$c_v^n = \frac{\sum_{\text{neighbors}} w_a^n \|g_a^n\|}{\sum_{\text{neighbors}} w_a^n}$$

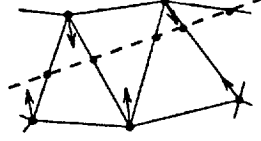


Figure 2: Attraction of Vertices to Shock

We flag points at which the density gradient exceeds this average of gradients at surrounding points. In particular, we threshold the quantity

$$\frac{\max(0, \|g_v^n\| - c_v^n)}{\|g_v^n\| + \epsilon},$$

where  $\epsilon > 0$  is needed to avoid division by zero in smooth regions. With both numerator and denominator proportional to the density gradient (neglecting epsilon), this detector is equally effective at detecting weak and strong shocks.

The vertex movement force at vertex  $v$  in this scheme is of the form:

$$s_v^n \frac{g_v^n}{\|g_v^n\|}$$

That is, a scalar multiple, of the unit vector in the direction of the density gradient. The scale factor  $s_v^n$ :

$$s_v^n = h_v^n (\rho_v^n - \bar{\rho}_v^n) \delta_v^n$$

with:

$h_v^n$  ..... local mesh size

$\rho_v^n$  ..... angle-weighted vertex average

$\bar{\rho}_v^n$  ..... local ambient density

$\delta_v^n$  ..... magnitude of local density range

We take  $h_v^n$  to be the minimum length of edges incident on  $v$ , while  $\rho_v^n$  is the angle-weighted vertex average, as before.

Quantities  $\delta_v^n$  and  $\bar{\rho}_v^n$  both depend on the range of density in a small surrounding region. Define the “local maximum density”  $\rho_{1,v}^n$  as the maximum density in cells touching vertices neighboring  $v$ . Thus  $\rho_{1,v}^n$  is the maximum density in the 20 or so surrounding cells. Similarly, define the “local minimum density”  $\rho_{0,v}^n$  as the minimum density in these surrounding cells. Then the local density range is:

$$\delta_v^n = \rho_{1,v}^n - \rho_{0,v}^n$$

Similarly, the local “ambient density” is

$$\bar{\rho}_v^n = (\rho_{1,v}^n + \rho_{0,v}^n)/2$$

Note that this is an average of density extremes, as opposed to a direct average of densities.

The motivation behind all of this is the following. A vertex along a shock is correctly located when the density value there is midway the high and low density in a surrounding region. Thus we want to satisfy the equation

$$\rho_v^n = \bar{\rho}_v^n$$

at vertices along shocks. The scale factor  $s_v^n$  approximates the amount of the correction needed to satisfy this equation. Since we adjust the grid at every iteration, the precise scale factor used is unimportant; in effect it is just a relaxation parameter.

## Mesh Control

Using either scheme A or scheme, vertices are rapidly attracted to shocks. However, without constraints on mesh movement, one rapidly produces undesirably thin cells, or negative cell areas. To avoid this two things are needed:

1. mesh control forces, partially counteracting the forces attracting shocks to vertices.
2. control of mesh movement step-size control.

We are currently using two forces, one proportional to the change in cell area, another based on angles at vertices. The latter, which applies torques on edges, prevents angles from approaching either 0 or 180 degrees. As angles approach either 0 or 180 degrees, the torques it produces become infinite. Thus if the ODE governing mesh movement is properly integrated, degenerate triangles cannot occur. By contrast, using “springs” on edges is not effective, since they cannot prevent angles from approaching 0 or 180 degrees.

There are a number of ways to control the step-size in the mesh movement scheme. Our approach is to compute a maximum step at each vertex, designed to prevent degenerate triangles. For every triangular cell  $R$  let  $h_{min}(T)$  be the minimum of the three altitudes. Then for each vertex define

$$h_{min}(v) = \min_{\text{neighboring triangles}} h_{min}(R)$$

If no vertex  $v$  moves further than  $\frac{1}{2}h_{min}(v)$ , degenerate triangles cannot occur.

Note that controlling step-size alone suffices to prevent degenerate triangles, but grid quality may be quite poor. The combination of these mesh control forces and step-size control suffices to maintaining mesh quality, while still allowing effective fitting of shocks.

## Generalized Van Albada Limiter

The first order scheme just described works well, but provides inadequate resolution in smooth regions. Second or third order accuracy can be achieved with a MUSCL-style scheme [12], in which one reconstructs a polynomial in every cell via an appropriate “limiter.” One way of doing this is to adapt the stencils, following the ENO approach. However ENO is complex and expensive on unstructured meshes [13].

Our approach is, instead, to use a multidimensional generalization of the Van Albada limiter [14]. This limiter is simple, reliable, and has the attractive property of not clipping extrema. Thus it can, in principle, achieve perfectly sharp approximations of N-waves on very coarse meshes.

The goal in a MUSCL scheme is to replace the constant value in each cell by a linearly varying distribution

$$q^n(\eta) = q_i^n + (\eta - \eta_i)(\delta q)_i^n$$

where  $(\delta q)_i^n$  is an approximation to the gradient. The Van Albada limiter takes this gradient as a nonlinear average of the gradients computed by forward and backward differencing:

$$(\delta q)_i^n = \text{ave}(q_{i+1}^n - q_i^n, q_i^n - q_{i-1}^n),$$

using the averaging function

$$\text{ave}(a, b) = \frac{(b^2 + \epsilon^2) a + (a^2 + \epsilon^2) b}{a^2 + b^2 + 2 \epsilon^2},$$

where epsilon is a small positive constant designed to provide smooth transitions. This kind of averaging was used in [14] for all quantities except density, which was handled slightly differently to avoid negative overshoots in strong astrophysical flows.

The Van Albada limiter generalizes easily to unstructured meshes. To see this, rewrite the above formulas as

$$(\delta q)_i^n = w_a (q_{i+1}^n - q_i^n) + w_b (q_i^n - q_{i-1}^n),$$

with

$$w_a = \frac{(b^2 + \epsilon^2)}{a^2 + b^2 + 2 \epsilon^2},$$

$$w_b = \frac{(a^2 + \epsilon^2)}{a^2 + b^2 + 2 \epsilon^2}$$

Now in a similar way, for triangular mesh cells, assume one has gradients  $g_a^n$ ,  $g_b^n$ ,  $g_c^n$  at the vertices of a triangle, obtained, as usual, by Green's theorem path integrals. Then one can compute the cell centered gradient as

$$g^n = w_a g_a^n + w_b g_b^n + w_c g_c^n$$

for suitable weights  $w_a$ ,  $w_b$ ,  $w_c$ . Constraining these weights by

$$w_a + w_b + w_c = 1$$

$$w_a, w_b, w_c \in [0, 1].$$

yields second order consistency of the overall scheme, assuming the nodal gradients are first order accurate. We also want to preserve the Van Albada property of not clipping extrema.

The particular choice we used was

$$\begin{aligned} w_a &= \frac{(b c + \epsilon^2)}{a b + b c + c a + 3 \epsilon^2}, \\ w_b &= \frac{(c a + \epsilon^2)}{a b + b c + c a + 3 \epsilon^2}, \\ w_c &= \frac{(a b + \epsilon^2)}{a b + b c + c a + 3 \epsilon^2}, \end{aligned}$$

with  $a = \|g_a\|^2$ ,  $b = \|g_b\|^2$ ,  $c = \|g_c\|^2$ . Other choices work about as well. In particular, one can chose

$$a = \|g_a\|, \quad b = \|g_b\|, \quad c = \|g_c\|,$$

in closer analogy with the original Van Albada scheme. We prefer the stronger switching that occurs in using the squares.

This generalized Van Albada limiter has the property that near strong jumps the reconstructed gradients use information entirely from one side of the jump, thus achieving second-order consistency while avoiding spurious oscillations. Thus one can think of this as an inexpensive approach to ENO, avoiding the use of complex adaptive stencils and to some extent the convergence difficulties they create.

## Parabolic Reconstruction

An interesting question to ask is whether the same approach can be used for higher order reconstruction. While we have implemented only the second, it is clear one can easily go to third or higher order. Given the original piecewise constant states in each cell, the Green's theorem path integrals yield first order accurate gradients at the vertices, so the resulting piecewise linear state in each cell is second order accurate. We can at this point repeat the path integrals to evaluate the gradients at the vertices to second order, or the second derivatives at the vertices to first order. The resulting information more than suffices for a Van Albada-style parabolic reconstruction in each cell, of third order accuracy, and the process can be continued.

Though this "boot strapping" approach might at first seem suspect, it is perfectly valid. In repeating the path integrations, one is effectively widening the difference stencil, thus allowing higher order approximation. There are, however, a couple of difficulties here. First, the higher order path integrals required are more expensive. Second, to fully exploit higher order reconstructions, one has to compute the flux balances via higher order Gauss formulas. For example, with parabolic reconstruction, one needs two quadrature points per side, doubling the cost of the method. However, even without higher order quadrature of the fluxes, it may still make sense to perform parabolic reconstruction, as in the "Piecewise Parabolic Method" of Woodward and Collela [15] which, in effect, reduces the constant in the truncation error while remaining formally second order accurate.

## 4 Sonic Boom Computations

While SLAM can be applied to steady-state Euler calculation (and presumably steady-state Navier Stokes calculations as well) one application for which it is especially well suited is computation of sonic booms. Any aircraft moving faster than the speed of sound generates a sonic boom. Near the aircraft, the various shocks, rarefactions, wakes and boundary layers interact in complex, nonlinear ways. However, a few body-lengths from the aircraft, these nonlinear waves usually coalesce into a characteristic N-wave, which then propagates linearly to the ground. As it does so, this N-wave undergoes diffraction through the atmospheres density gradients, and Doppler shifts in the wind layers encountered.

One can compute the shock waves near the aircraft with any three dimensional flow solver. However, current mesh resolutions are inadequate for detailed sonic boom calculation [16, 17]. Moreover, shock steepening is still occurring many body-lengths from the aircraft, so that a very large mesh may be necessary. Through Lagrangian mesh movement, SLAM can find even extremely weak shocks and compute them accurately on relatively coarse meshes. Thus its potential capabilities in this kind of computation should be much better than competing adaptive-mesh methods.

Though SLAM is not yet implemented in 3D, its theoretical potential is easy to assess. One can make the argument for the dramatic increase in efficiency of Lagrangian fitting strategies over mesh-enrichment strategies through a simple geometric argument, like that given by Swartz [18]. Instead, we look here at a simple model problem. Take as model problem the problem of calculating the bow shock from a high speed projectile. Assume, for convenience the shock is a perfect cone, and assume we wish to resolve the shock in cells of size  $\delta$  in a domain of radius  $R$ , as shown:

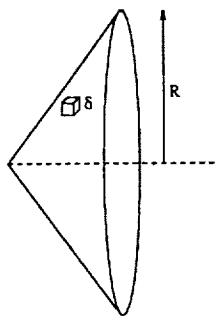


Figure 3: Resolving a Conical Shock

Using a uniform mesh, the number of cells required,  $N$ , evidently satisfies

$$N \propto \left(\frac{R}{\delta}\right)^3$$

or equivalently:

$$\delta \propto RN^{-1/3}$$

Adaptive refinement strategies greatly improve the resolution achieved with a given number of cells. Consider, for example, a Cartesian adaptive strategy based on oct-tree refinement. In this case, the total number of cells required is proportional to the number required to tile the shock (assuming cells sizes grow geometrically away from the shock). Thus we have

$$N \propto \left(\frac{R}{\delta}\right)^2$$

or:

$$\delta \propto RN^{-1/2}$$

In effect the problem dimension has been reduced by one.

With SLAM, one can resolve the shock to within a distance of  $\delta$  using tetrahedrons of much larger diameter. The positional error in approximating a fitted shock with tetrahedrons of diameter  $h$  is

$$e \propto h^2 \kappa$$

where  $\kappa$  is the local curvature of the shock. Then setting  $e = \delta$  we find we should take:

$$h = \sqrt{\delta/\kappa}$$

Now the curvature  $\kappa$  of a conical shock goes as  $1/r$ , where  $r$  is the distance from the axis. Thus the size of cells in a properly graded mesh would be:

$$h(r) = \sqrt{\delta r}$$

Now to count the cells tiling the shock, define the cell density  $d(r) = 1/h^2(r)$ . The number of tetrahedrons needed to tile the shock is then the integral of  $d(r)$  over the shock

$$N \propto \int_0^R 2\pi r d(r) dr = 2\pi \int_0^R \frac{1}{\delta} dr$$

so that:

$$N \propto \frac{R}{\delta}; \quad \delta \propto RN^{-1}$$

The total number of cells will again be proportional to the number required to tile the shock. Thus we have the following surprising result:

*The cost of resolving a conical shock wave in three dimensions to within  $\delta$  using SLAM is asymptotically the same as that of solving a one dimensional shock problem on a uniform mesh of spacing  $\delta$ .*

In other words, the combination of fitting together with mesh-grading is sufficient to achieve third order global accuracy, despite the point-singularity at the nose.

This is a theoretical, asymptotic result, neglecting the many complex implementation issues involved. It also assumes that the flow near the body can be solved with cost at most proportional to the cost of computing the shock, and that the smooth regions in the far field can be resolved on such coarse meshes that their cost is negligible. Modulo these caveats, this analysis is valid, and suggests that computation of sonic booms produced by supersonic aircraft should be quite tractable, with SLAM-style algorithms.

## 5 Experimental Results

Results obtained with SLAM, while preliminary, are very promising. We have, in general, no trouble with strong shocks, including attached and detached bow shocks, fish tail shocks, and standing shocks on transonic airfoils. Similar techniques can be used to resolve slip lines and contacts [9], though we have not yet studied this.

Figure 4 shows an un-adapted mesh of 8,000 points around a 10% circular arc airfoil, while Figure 5 shows the same mesh after modification by SLAM to align mesh edges with shocks. Figure 6 shows the density field on this 8,000 point adapted mesh. The shocks are sharp all the way to the far-field boundary, and the limiter is also producing an accurate solution in the smooth region between shocks.

One can judge the solution more accurately by taking cross sections. Figures [7-10] show SLAM solutions on the 8,000 point mesh in Figure 5, and on an analogous 4,000 point mesh. Figures 7 and 8 show the density cross sections taken parallel to the axis one chord from the axis, while Figures 9 and 10 show density cross sections four chords from the axis. As can be seen, the sonic-boom profile is well captured, even four chords from the axis. These solutions are mesh-converged to graphical accuracy, in smooth regions. However, there is a slight anticipation of the bow shock, worse on the 4,000 point mesh, due to error in shock location. There are several numerical effects, creating second-order errors in shock location.

For this simple problem, one can obtain qualitatively reasonable solutions via SLAM, using fewer than 1,000 mesh points. However, accuracy is lacking until the smooth regions are resolved. By contrast, Figures [11-14] show the solution on the un-adapted 8,000 point mesh of Figure 4, and on an analogous 32,000 points mesh. With the Lagrangian mesh adaptivity turned off, the sonic boom profiles are now badly distorted, especially on the 8,000 point mesh, even though we made a real effort to concentrate mesh in regions where the sonic boom was expected.

On the 8,000 point mesh, at four chords from the axis, Figure 13, the bow and tail shocks are separated by about 15 mesh widths. Thus significant smearing is inevitable. This smearing is not as severe on the 32,000 point mesh, which has four times the mesh density throughout the flow field, but the answer there is still much poorer than the SLAM solutions. In particular, comparing Figures 14 and 10, notice that the extrema are substantially blurred on the 32,000 point un-adapted mesh. The combination of Lagrangian adaptivity and generalized Van Albada limiter is particularly effective at getting extrema right.

For this simple example, one requires at least 60 times more points with an un-adapted mesh than with SLAM for comparable. For example, halving the mesh size in the 32,000 point mesh, one would yield a 128,000 point mesh achieving accuracy approaching that of the 4,000 point SLAM solution.

Figure 15 shows a somewhat more complicated example, flow over an airfoil with a perfectly sharp nose. Since the interior angle at the nose of this airfoil is zero, there is no shock there. Instead, a shock forms in the free stream, some distance away, where the acoustic waves coalesce. Figures 16 and 17 show the cross section 0.1 and 0.2 chords respectively from the axis. The smooth profile at the nose in Figure 16 has begun to sharpen into a shock in Figure 17. Steepening continues, as shown in Figure 18, at 0.4 chords from the axis. At 0.8 chords, Figure 19, the eventual N-wave, is well established.

The point here is that, unlike most shock-fitting schemes, the Lagrangian adaptive scheme

has no effect on the underlying conservative discretization. Thus examples like this, with coalescing waves, intersecting shocks, and so on, present no difficulty, at least in principal.

We have not yet done experiments to compare SLAM with a standard mesh-enrichment adaptive strategy, though there is little doubt how the comparison will come out. SLAM achieves shock-fitted accuracy without addition of mesh points, while enrichment strategies substantially increase in the number of mesh points. Moreover, since enrichment strategies leave the mesh unaligned, the well resolved shocks there will still be smeared over several mesh points. Thus SLAM should require about one tenth as many mesh points as mesh-enrichment schemes in 2D, and should be relatively even better in 3D. The results of such a comparison will be reported in a sequel.

## 6 Conclusions

Lagrangian adaptive grid methods, like SLAM, have great potential for resolving shocks and other flow discontinuities. Unlike mesh-enrichment strategies, which put much finer mesh along shocks, fitting strategies can resolve discontinuities without increasing the number of mesh cells. This advantage is especially important in three dimensions, where the cost of tiling shocks with fine mesh is great. This improved resolution is achieved at little cost, and without loss of robustness, since we retain a Roe scheme-based shock capturing scheme. Thus even if the fitting scheme fails, we still have a robust and effective shock-capturing algorithm.

We have demonstrated the efficacy of SLAM in 2D, and are beginning work on an analogous 3D code. The latter is intended to be applied to the problem of predicting the sonic boom profiles of supersonic aircraft. Current CFD codes cannot adequately resolve the complex shock waves emanating from a supersonic vehicle, since one cannot afford a sufficiently fine grid extending 4-5 body-lengths from the aircraft. Fitting schemes, like SLAM, will be able to do much better, and should be able to reproduce the complex shock patterns observed in wind-tunnel experiments.

## Acknowledgment

The author would like to thank K. Anderson for providing the robust unstructured grid flow solver used [1]. We also had help from D. Mavriplis in constructing high-quality initial grids using his advancing front grid generator [19]. This grid generator allows careful clustering of points near the expected shocks, and has no trouble with the sharp-nosed airfoil, even though this airfoil has zero thickness at the nose. Kwan-Liu Ma helped produce the high-resolution graphics images, and finally, V. Venkatakrishnan provided sound advice on the design of effective limiters.

## References

- [1] W. K. Anderson. A grid generation and flow solution method for the Euler equations on unstructured grids. *Journal of Computational Physics*, 110(1):23–38, Jan. 1994.
- [2] J. Paraschivoiu, J.-Y. Trepanier, M. Reggio, and R. Camarero. A conservative dynamic discontinuity tracking algorithm for the Euler equations. Number AIAA 94-0081, Jan. 1994.
- [3] I.J. Parpia and P. Parikh. A solution adaptive mesh generation method with cell-face orientation control. Jan. 1994.
- [4] G. Morreti. A technique for integrating two-dimensional Euler equations. *Computers and Fluids*, 15(1), 1987.
- [5] M. D. Salas. Shock-fitting method for complicated two-dimensional supersonic flows. *AIAA Journal*, 14(5):583–588, 1976.
- [6] P. M. Hartwich. Fresh look at floating shock fitting. *AIAA Journal*, 29(7):1084–1091, July 1991.
- [7] P. L. Roe. Discrete models for the numerical analysis of time-dependent multidimensional gas dynamics. *Journal of Computational Physics*, 63, 1986.
- [8] J. Hase and I. Parpia. Flux limiters in a rotated upwind scheme for the Euler equations. *AIAA paper 93-0066*, Jan. 1993.
- [9] J. Paraschivoiu. Une méthode adaptive pour la résolution exacte des ondes de chocs et des discontinuités de contact. Master's thesis, École Polytechnique de Montréal, 1993.
- [10] H. Deconinck, R. Stuijs, G. Bourgois, H. Paillère, and P.L. Roe. Multidimensional upwind methods for unstructured grids. Technical report, AGARD, 1992.
- [11] J. J. Quirk. A contribution to the great Riemann solver debate. *Intl. J. Num. Meth. Fluids*, Vol. 18, pp. 555-574, 1994.
- [12] B. van Leer. Towards the ultimate conservative difference scheme. V. a second-order sequel to Godunov's method. *Journal of Computational Physics*, 32(1), July 1979.
- [13] R. Abgrall. Design of an essentially non-oscillatory reconstruction procedure on finite-element type meshes. Technical report, ICASE, Dec. 1991.
- [14] G. D. van Albada, B. van Leer, and W. W. Roberts. A comparative study of computational methods in cosmic gas dynamics. *Astronomy and Astrophysics*, 108:76–84, 1982.
- [15] P. R. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54:115–173, 1984.

- [16] C. Darden M. Siclari. CFD predictions of the near-field sonic boom environment for two low boom HSCT configurations. Number AIAA-91-1631, June 1991.
- [17] S. H. Cheung and Thomas A. Edwards and Scott L. Lawrence. Application of computational fluid dynamics to sonic boom near- and mid-field prediction. *Journal of Aircraft*, 29(5):920–926, Sept.-Oct. 1992.
- [18] B. Swartz. Courant-like conditions limit reasonable mesh refinement to order  $h^2$ . *SIAM J. on Sci. Comp.*, 8:924–933, Nov 1987.
- [19] D. J. Mavriplis. An advancing front delaunay triangulation algorithm designed for robustness. Technical report, ICASE, Oct. 1992.

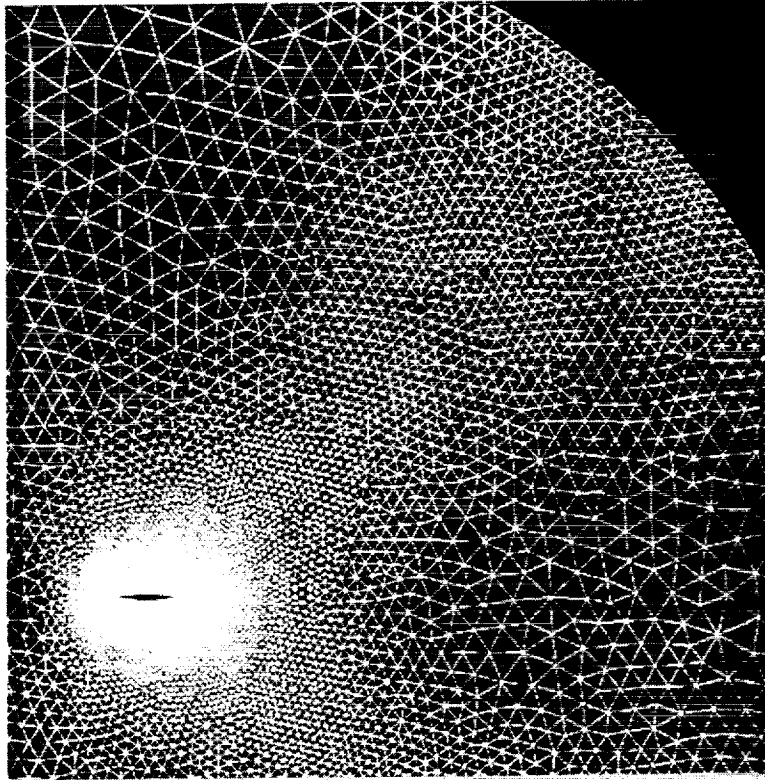


Figure 4: Initial Mesh for Circular-Arc Airfoil

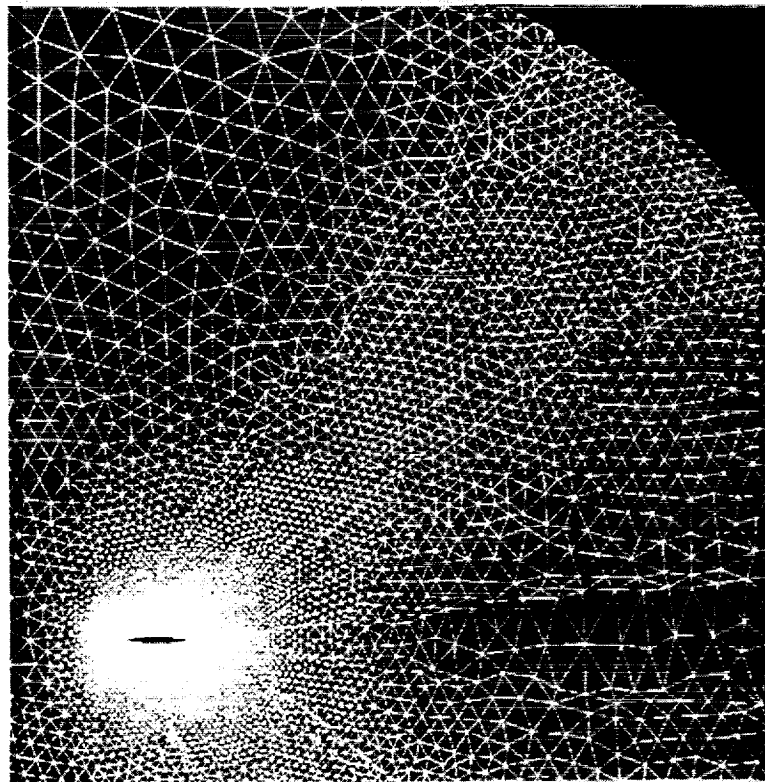


Figure 5: Adapted Mesh for Circular Arc Airfoil

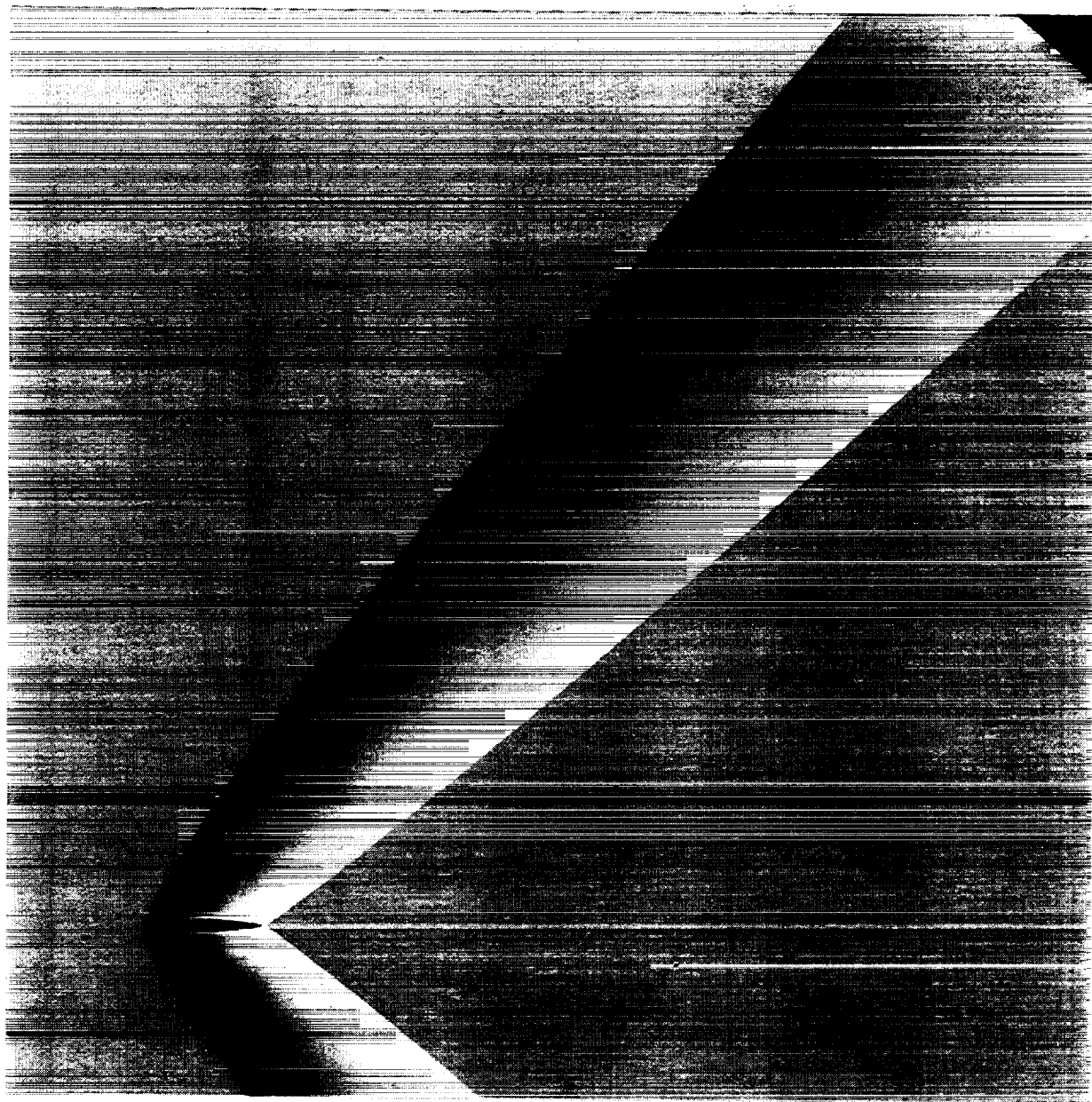


Figure 6: Flow Field, 8000 Point Adapted Mesh

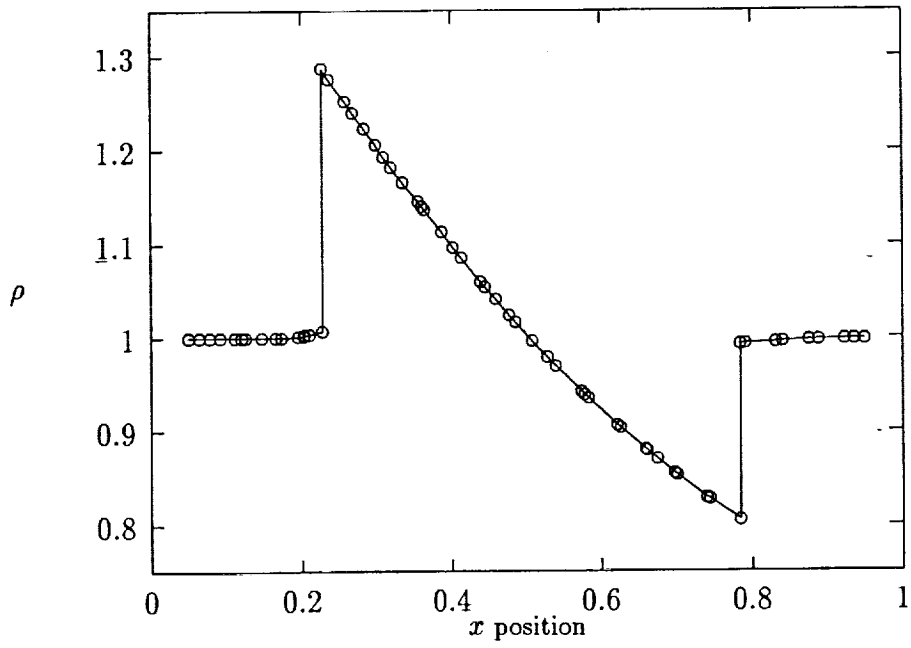


Figure 7: Density One Chord from Axis, Circular Arc Airfoil  
Mach 1.4, 4000-point Adapted Mesh

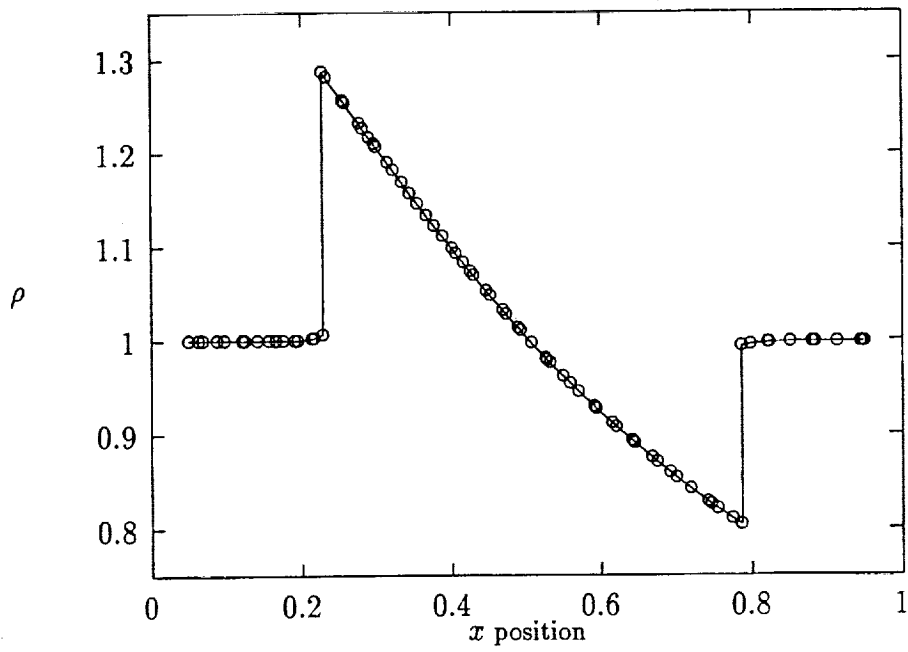


Figure 8: Density One Chord from Axis, Circular Arc Airfoil,  
Mach 1.4, 8000-point Adapted Mesh

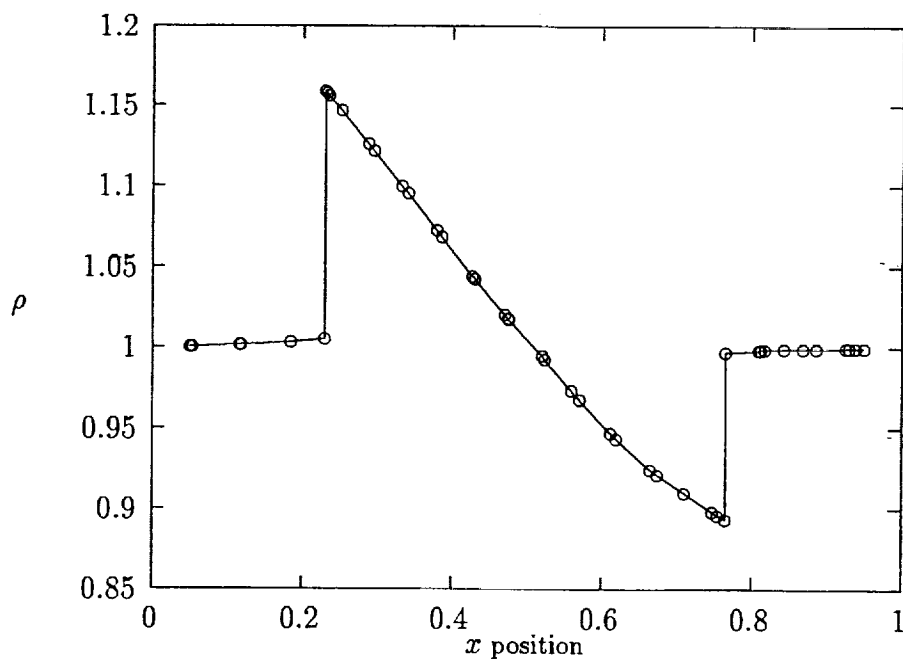


Figure 9: Density Four Chords from Axis, Circular Arc Airfoil  
Mach 1.4, 4000-point Adapted Mesh

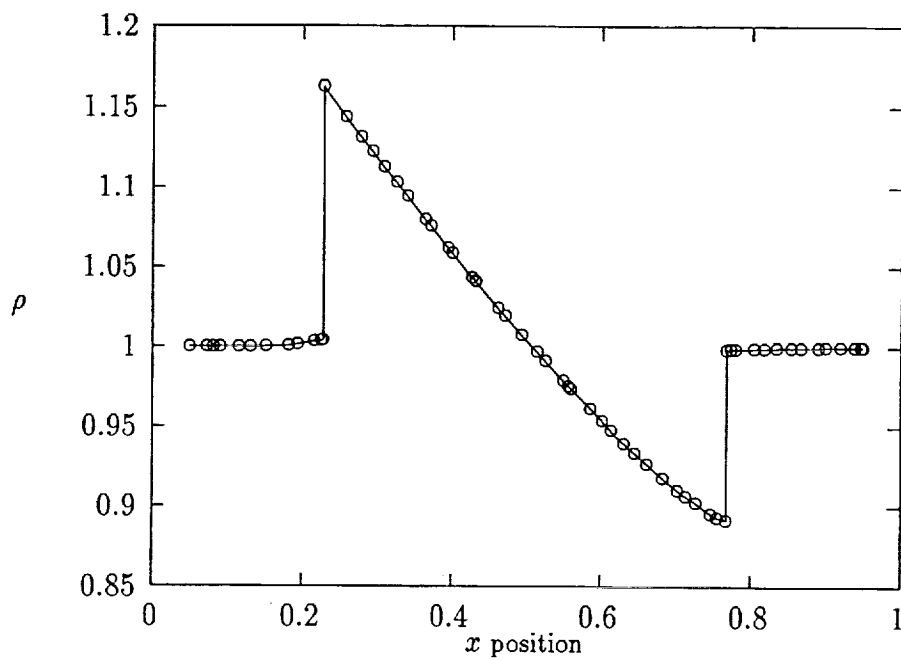


Figure 10: Density Four Chords from Axis, Circular Arc Airfoil  
Mach 1.4, 8000-point Adapted Mesh

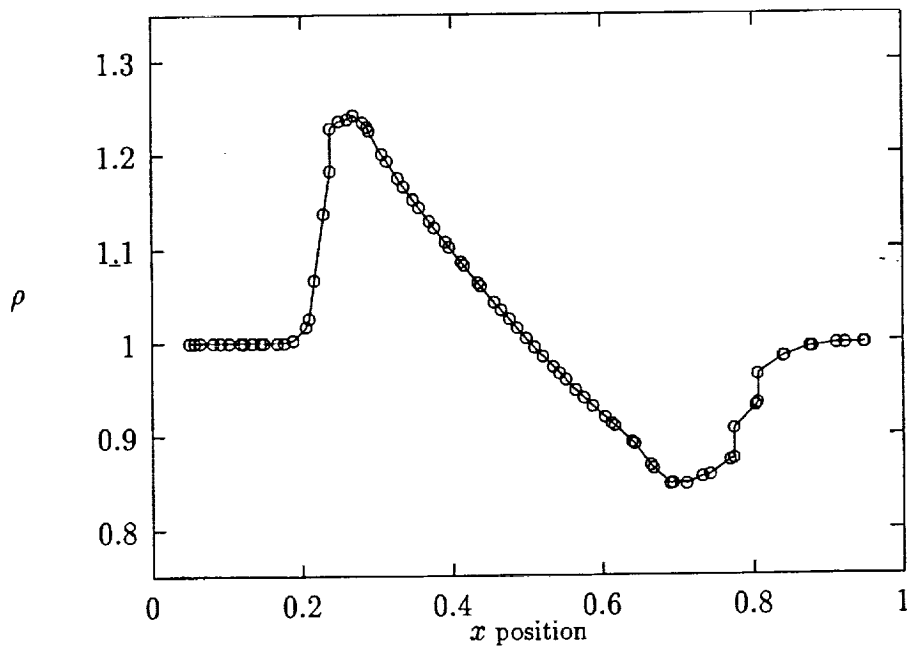


Figure 11: Density One Chord from Axis, Circular Arc Airfoil  
Mach 1.4, 8000-point Un-adapted Mesh

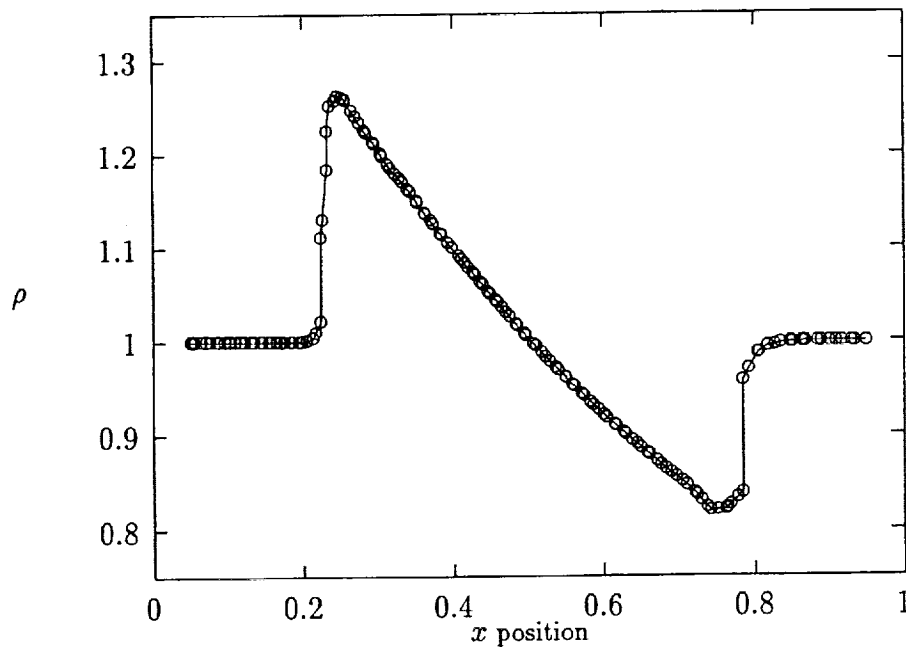


Figure 12: Density One Chord from Axis, Circular Arc Airfoil  
Mach 1.4, 32000-point Un-adapted Mesh

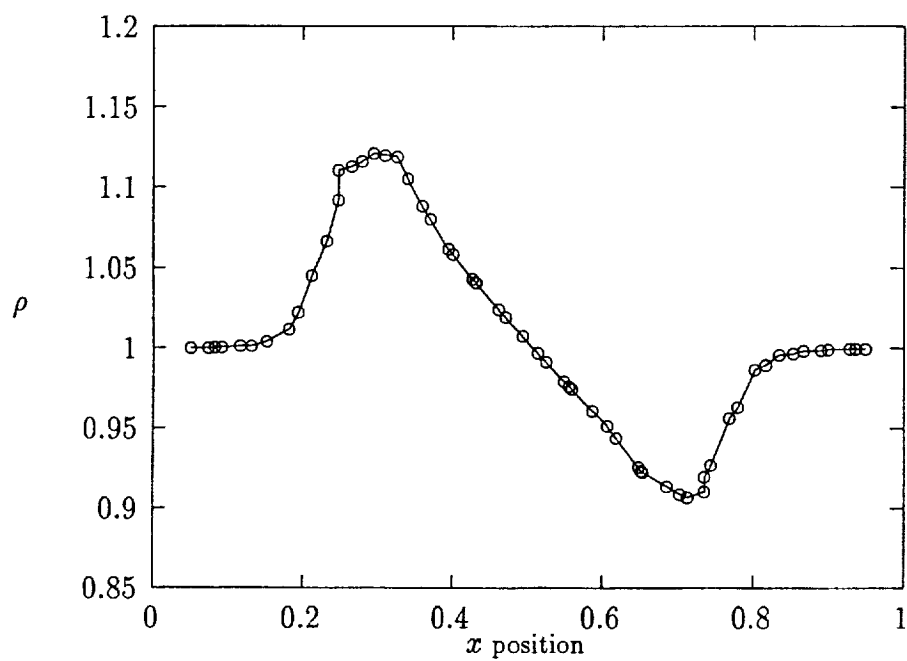


Figure 13: Density Four Chords from Axis, Circular Arc Airfoil  
Mach 1.4, 8000-point Un-adapted Mesh

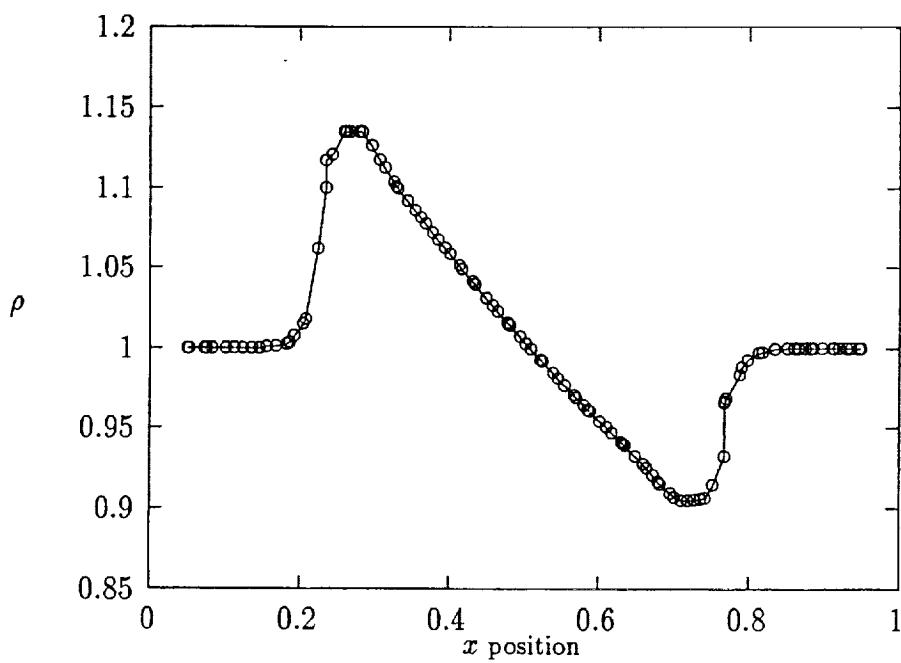


Figure 14: Density Four Chords from Axis, Circular Arc Airfoil  
Mach 1.4, 32000-point Un-adapted Mesh

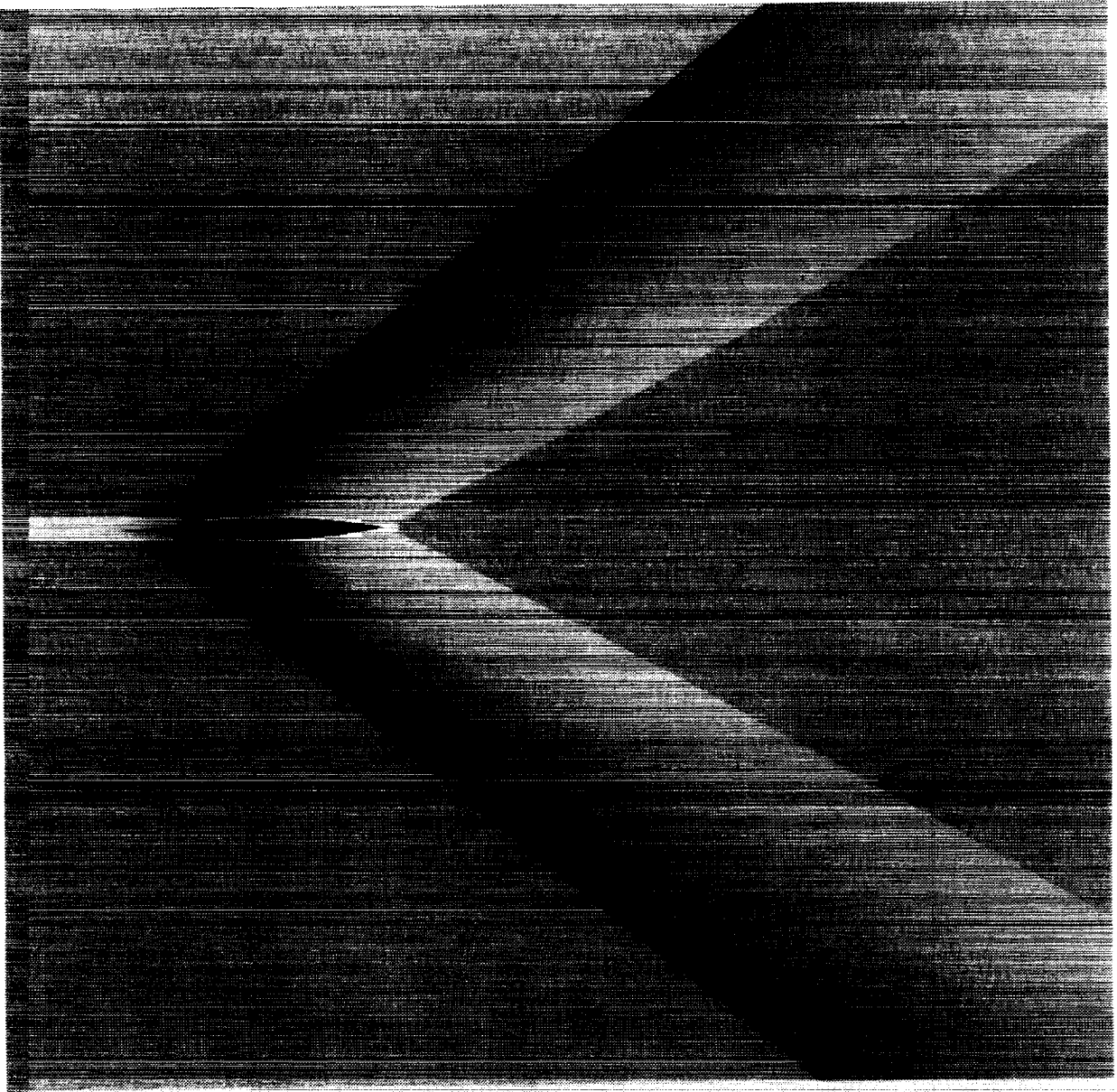


Figure 15: Flow Field, Pointed-nose Airfoil

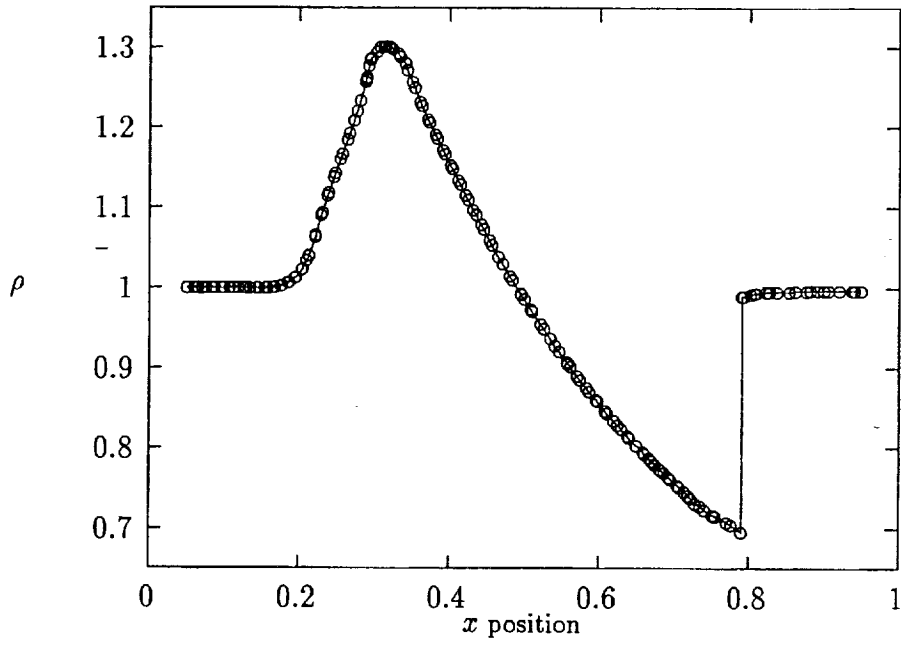


Figure 16: Density 0.1 Chords from Axis, Pointed-nose Airfoil  
Mach 1.8, 9000-point Adapted Mesh

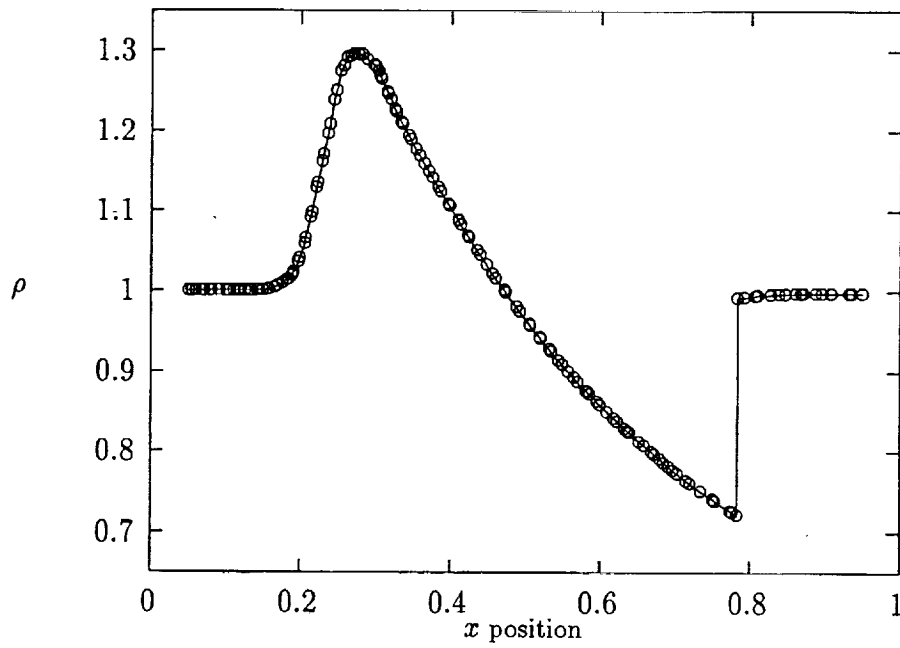


Figure 17: Density 0.2 Chords from Axis, Pointed-nose Airfoil  
Mach 1.8, 9000-point Adapted Mesh

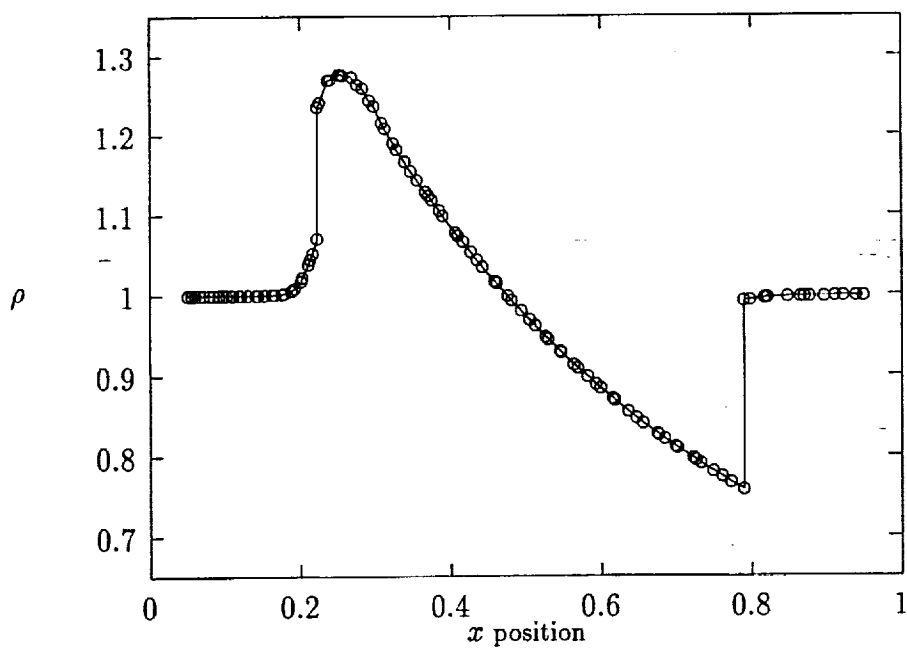


Figure 18: Density 0.4 Chords from Axis, Pointed-nose Airfoil  
Mach 1.8, 9000-point Adapted Mesh

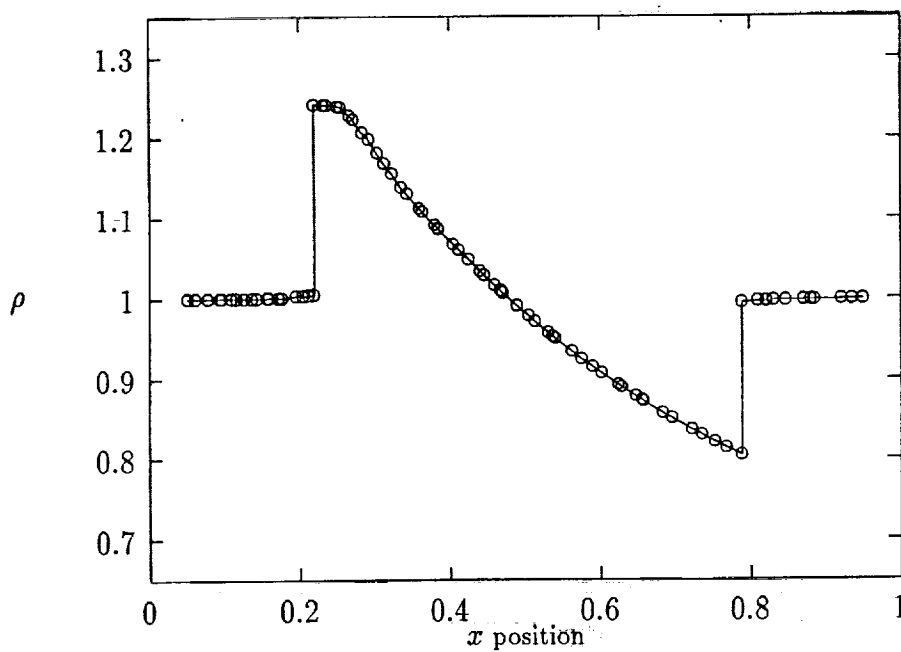


Figure 19: Density 0.8 Chords from Axis, Pointed-nose Airfoil  
Mach 1.8, 9000-point Adapted Mesh





REPORT DOCUMENTATION PAGE			Form Approved OMB No 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY(Leave blank)	2. REPORT DATE November 1994	3. REPORT TYPE AND DATES COVERED Contractor Report		
4. TITLE AND SUBTITLE FLOATING SHOCK FITTING VIA LAGRANGIAN ADAPTIVE MESHES		5. FUNDING NUMBERS C NAS1-19480 WU 505-90-52-01		
6. AUTHOR(S) John Van Rosendale				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23681-0001		8. PERFORMING ORGANIZATION REPORT NUMBER ICASE Report No. 94-89		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Langley Research Center Hampton, VA 23681-0001		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA CR-194997 ICASE Report No. 94-89		
11. SUPPLEMENTARY NOTES Langley Technical Monitor: Michael F. Card Final Report Submitted to 1995 AIAA Harbor Island CFD Conference				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited  Subject Category 64		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) In recent works we have formulated a new approach to compressible flow simulation, combining the advantages of shock-fitting and shock-capturing. Using a cell-centered Roe scheme discretization on unstructured meshes, we warp the mesh while marching to steady state, so that mesh edges align with shocks and other discontinuities. This new algorithm, the Shock-fitting Lagrangian Adaptive Method (SLAM) is, in effect, a reliable shock-capturing algorithm which yields shock-fitted accuracy at convergence. Shock-capturing algorithms like this, which warp the mesh to yield shock-fitted accuracy, are new and relatively untried. However, their potential is clear. In the context of sonic booms, accurate calculation of near-field sonic boom signatures is critical to the design of the High Speed Civil Transport (HSCT). SLAM should allow computation of accurate N-wave pressure signatures on comparatively coarse meshes, significantly enhancing our ability to design low-boom configurations for high-speed aircraft.				
14. SUBJECT TERMS adaptive refinement, shock-fitting, sonic booms			15. NUMBER OF PAGES 24	
			16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT	

