FLOATING SHOCK FITTING VIA LAGRANGIAN ADAPTIVE MESHES*

John Van Rosendale
Institute for Computer Applications in Science and Engineering
M.S. 132 C, NASA Langley Research Center
Hampton, VA 23681

SUMMARY

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 **S**hock-fitting **L**agrangian **A**daptive **M**ethod (SLAM) is, in effect, a reliable shock-capturing algorithm which yields shock-fitted accuracy at convergence.

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 apriori 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

*This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480.

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 gives virtually exact answers there. This is the basic idea of the **S**hock-fitting **L**agrangian **A**daptive **M**ethod. 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.

RELATED WORK

While shock-fitting has existed for decades [2, 3, 4], 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 discretizations, but beyond that the details of these approaches differ. Parpia and Parikh [5] use the waves occuring 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 [6, 7]. 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 [8, 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 occuring 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.

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 [10]. 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.

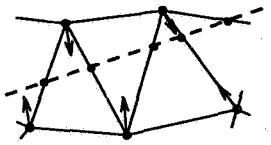

Figure 1: Attraction of Vertices to Shock

Our scheme for warping the mesh consists of two separate components:

1. A shock detector
2. A vertex attraction force

The latter is applied only at points detected as shocks. There is no direct consideration of either attracting or orienting edges, we simply attract vertices to shocks. 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.*

Our current 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:

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

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.

Once we have flagged the vertices along the shock, we attract vertices to the shock, by applying forces to each shocked vertex. The force at vertex v in our 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

ρ_v^n ..... angle-weighted vertex average

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

δ_v^n ..... magnitude of local density range

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

Quantities δ_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

In our 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. 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

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

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 [11], 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 [12].

Our approach is, instead, to use a multidimensional generalization of the Van Albada limiter [13]. 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 [13] 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

$$w_a = \frac{(b\,c + \epsilon^2)}{a\,b + b\,c + c\,a + 3\,\epsilon^2},$$

$$w_b = \frac{(c\,a + \epsilon^2)}{a\,b + b\,c + c\,a + 3\,\epsilon^2},$$

$$w_c = \frac{(a\,b + \epsilon^2)}{a\,b + b\,c + c\,a + 3\,\epsilon^2},$$

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.

EXPERIMENTAL RESULTS

Results obtained with SLAM, while preliminary, seem quite 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 2 shows an un-adapted mesh of 8,000 points around a 10% circular arc airfoil, while Figure 3 shows the same mesh after modification by SLAM to align mesh edges with shocks. Figure 4 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 [6-7] show density contours on cross sections one and four chords from the axis, computed using the the 8,000 point mesh in Figure 3. 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 the smooth regions. However, there is a slight anticipation of the bow shock, due to small errors in the shock location. This is caused by several numerical effects, which create second-order errors in the shock location.

For this simple problem, one can obtain qualitatively reasonable solutions via SLAM, using as few than 1,000 mesh points. However, accuracy is lacking until the smooth regions are resolved. By contrast, Figures [8-9] show the solution on the un-adapted 8,000 point mesh of Figure 2, one and four chords from the axis. With the Lagrangian mesh adaptivity turned off, the sonic boom profiles are now badly distorted. Figures [10-11] show the same solution on a 32,000 point mesh. The solution is still quite smeared, even though this is a second order accurate scheme, and we made a real effort to locate mesh in regions where the sonic boom was expected.

On the 8,000 point mesh, at four chords from the axis, Figure 9, 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 on the 8,000 point mesh. In particular, comparing Figures 11 and 7, notice that the extrema are substantially blurred on the 32,000 point un-adapted mesh. The combination of Lagrangian adaptivity and our generalized Van Albada limiter is particularly effective at getting the extrema right.

Figure 5 shows a 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 lambda shock forms in the free stream, some distance away, where the acoustic waves coalesce. Figures 12 show the density cross section just off the body (0.1 chords from the axis). The smooth profile at the nose in Figure 12 rapidly steepens into a shock. By 0.4 chords, shown in Figure 13, the eventual N-wave solution is beginning to form.

The point of this second example 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 are in the process of comparing the SLAM algorithm with standard mesh-enrichment schemes. SLAM achieves shock-fitted accuracy without addition of mesh points, while enrichment strategies substantially increase the number of mesh points, and still produce diffused shocks. Thus SLAM should, in general, 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.

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 several 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.

ACKNOWLEDGMENTS

The author would like to thank K. Anderson for providing the robust unstructured grid flow solver used [2]. We also had help from R. Rausch and D. Mavriplis in grid generation. In particular, our high-quality initial grids were created using Mavriplis' advancing front grid generator [5], which allows clustering of points near the expected shocks, and had no trouble with the sharp-nosed airfoil with 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 limiters.

REFERENCES

[1] R. ABGRALL, *Design of an essentially non-oscillatory reconstruction procedure on finite-element type meshes*, tech. rep., ICASE, Dec. 1991.

[2] W. K. ANDERSON, *A grid generation and flow solution method for the Euler equations on unstructured grids*, Journal of Computational Physics, 110 (1994), pp. 23–38.

[3] P. M. HARTWICH, *Fresh look at floating shock fitting*, AIAA Journal, 29 (1991), pp. 1084–1091.

[4] J. HASE AND I. PARPIA, *Flux limiters in a rotated upwind scheme for the Euler equations*, AIAA paper 93-0066, (1993).

[5] D. J. MAVRIPLIS, *An advancing front delaunay triangulation algorithm designed for robustness*, tech. rep., ICASE, Oct. 1992.

[6] G. MORRETI, *A technique for integrating two-dimensional Euler equations*, Computers and Fluids, 15 (1987).

[7] J. PARASCHIVOIU, *Une méthod adaptive pour la résolution exact des ondes de chocs et des discontinuités de contact*, Master's thesis, École Polytechnique de Montréal, 1993.

[8] J. PARASCHIVOIU, J.-Y. TREPANIER, M. REGGIO, AND R. CAMARERO, *A conservative dynamic discontinuity tracking algorithm for the Euler equations*, no. AIAA 94-0081, Jan. 1994.

[9] I. PARPIA AND P. PARIKH, *A solution adaptive mesh generatinon method with cell-face orientation control*, Jan. 1994.

[10] J. QUIRK, *A contribution to the great Riemann solver debate*, tech. rep., ICASE, Nov. 1992.

[11] P. ROE, *Discrete models for the numerical analysis of time-dependent multidimensional gas dynamics*, Journal of Computational Physics, 63 (1986).

[12] M. D. SALAS, *Shock-fitting method for complicated two-dimensional supersonic flows*, AIAA Journal, 14 (1976), pp. 583–588.

[13] 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 (1982), pp. 76–84.

[14] B. VAN LEER, *Towards the ultimate conservative difference scheme. V. a second-order sequel to Gudonov's method*, Journal of Computational Physics, 32 (1979).

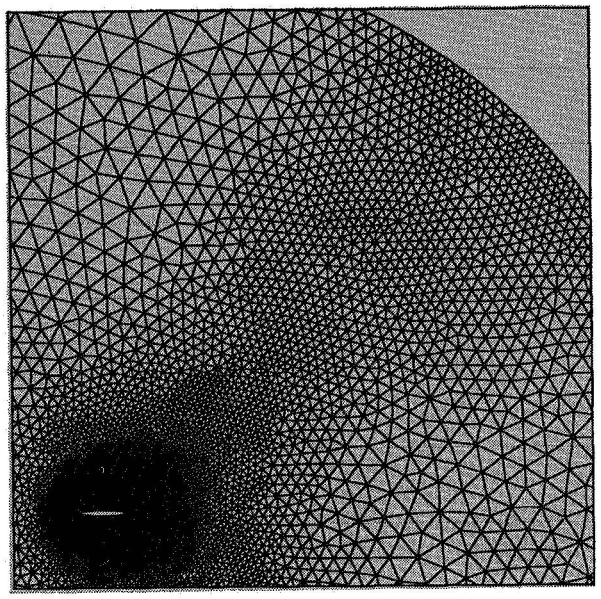

Figure 2. Initial Mesh for Circular Arc Airfoil

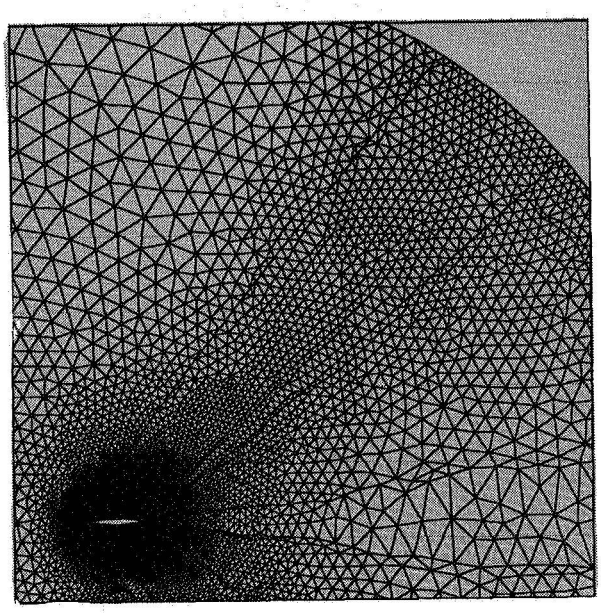

Figure 3. Adapted Mesh for Circular Arc Airfoil

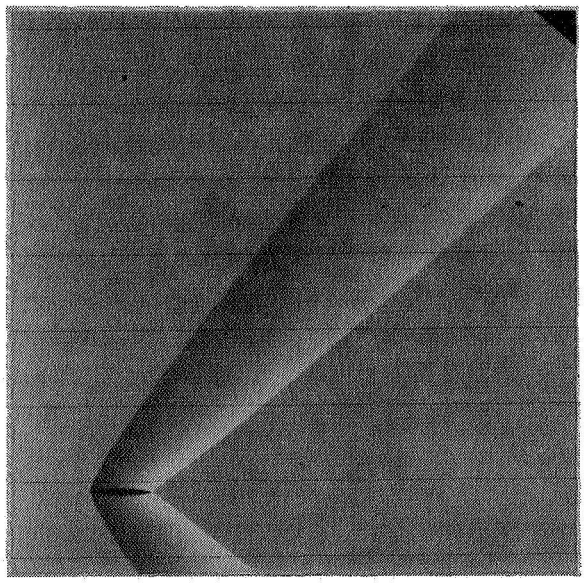

Figure 4. Flow Field, Circular Arc Airfoil, Mach 1.4, 8000-point Adapted Mesh

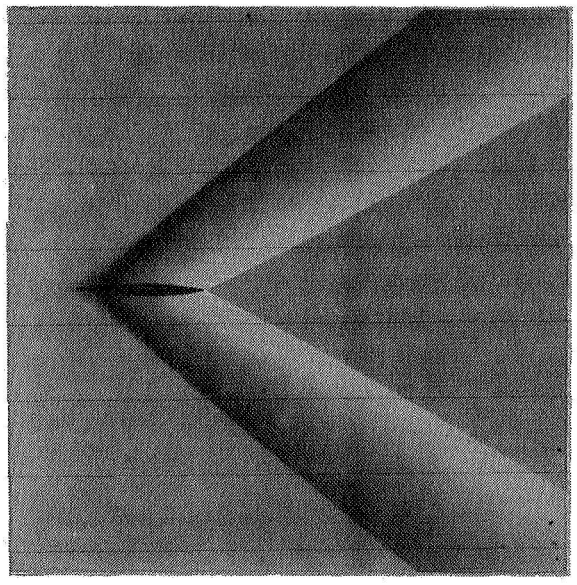

Figure 5. Flow Field, Pointed-nose Airfoil Mach 1.8, 9000-point Adapted Mesh

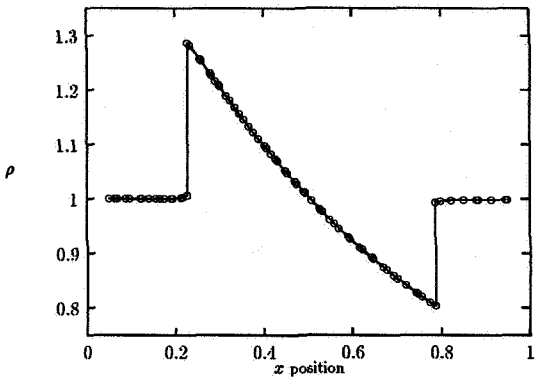

Figure 6. Density One Chord from Axis, Circular Arc Airfoil, Mach 1.4, 8000-point Adapted Mesh

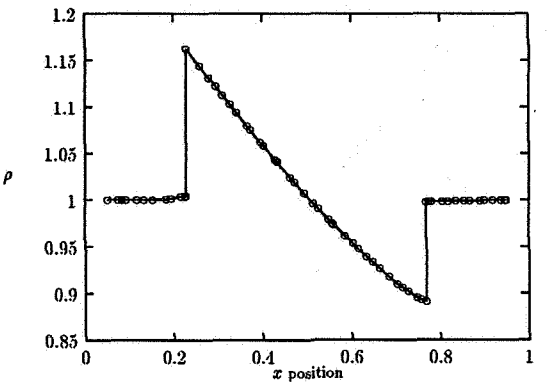

Figure 7. Density Four Chords from Axis, Circular Arc Airfoil, Mach 1.4, 8000-point Adapted Mesh

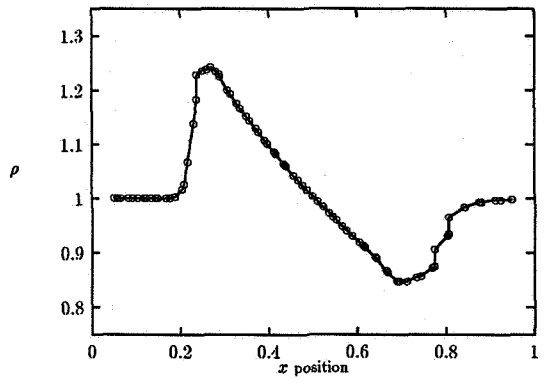

Figure 8. Density One Chord from Axis, Circular Arc Airfoil, Mach 1.4, 8000-point Un-adapted Mesh

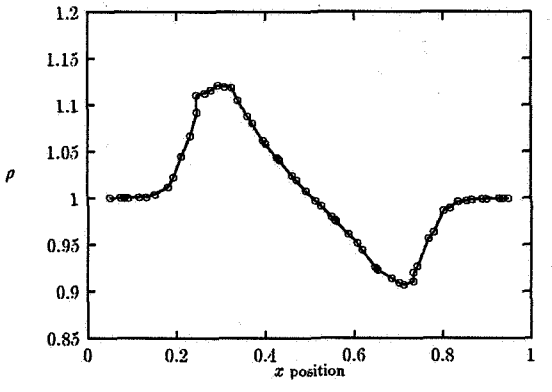

Figure 9. Density Four Chords from Axis, Circular Arc Airfoil, Mach 1.4, 8000-point Un-adapted Mesh

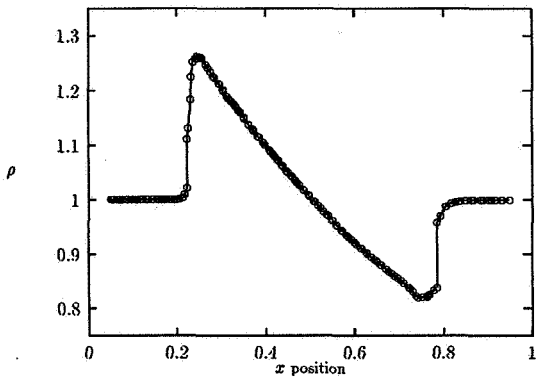

Figure 10. Density One Chord from Axis, Circular Arc Airfoil, Mach 1.4, 32000-point Un-adapted Mesh

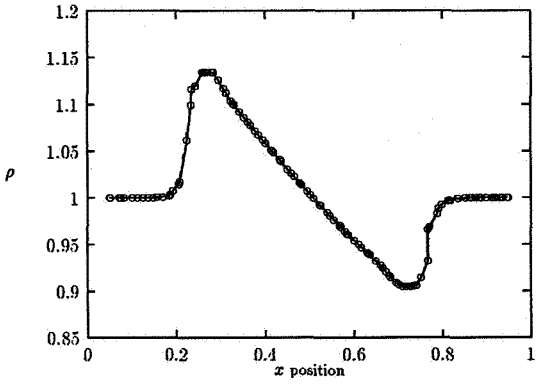

Figure 11. Density Four Chords from Axis, Circular Arc Airfoil, Mach 1.4, 32000-point Un-adapted Mesh

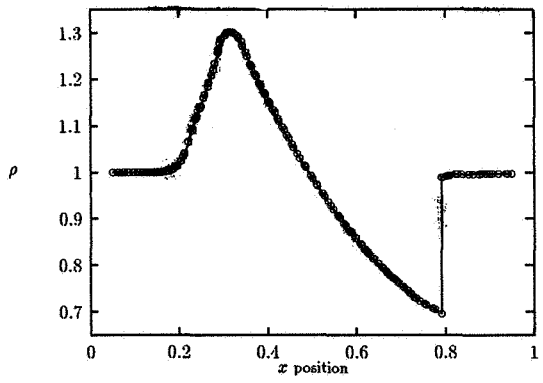

Figure 12. Density 0.1 Chords from Axis, Pointed-nose Airfoil, Mach 1.8, 9000-point Adapted Mesh

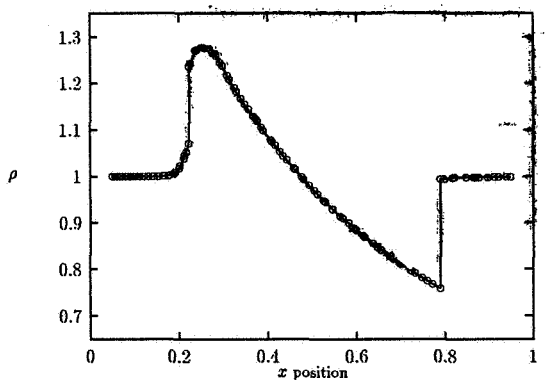

Figure 13. Density 0.4 Chords from Axis, Pointed-nose Airfoil, Mach 1.8, 9000-point Adapted Mesh