S16-34

7338

p.8

ADAPTIVE MESH REFINEMENT IN CURVILINEAR BODY-FITTED GRID SYSTEMS

Erlendur Steinthorsson and David Modiano
Institute for Computational Mechanics in Propulsion
NASA Lewis Research Center, Cleveland, Ohio

Phillip Colella
Department of Mechanical Engineering
University of California, Berkeley, California

SUMMARY

To be truly compatible with structured grids, an AMR algorithm should employ a block structure for the refined grids to allow flow solvers to take advantage of the strengths of structured grid systems, such as efficient solution algorithms for implicit discretizations and multigrid schemes. One such algorithm, the AMR algorithm of Berger and Colella, has been applied to and adapted for use with body-fitted structured grid systems. Results are presented for a transonic flow over a NACA0012 airfoil (AGARD-03 test case) and a reflection of a shock over a double wedge.

INTRODUCTION

Solution adaptive mesh refinement (AMR) can be used to enhance accuracy and efficiency of many practical flow solvers. By now it is used, almost routinely, in flow solvers employing unstructured grids. Yet, in flow solvers designed for body-fitted curvilinear grid systems (structured grids), similar techniques are rarely used. A possible reason is that the limitations of FORTRAN77, the programming language typically used for structured flow solvers, make it difficult to implement AMR algorithms that are truly compatible with the use of structured grids.

To be compatible with the use of structured grids, an AMR algorithm should allow the flow solver to take full advantage of the strengths of structured grid systems, such as allowing effective use of various efficient solution algorithms for implicit discretizations, various schemes based on dimensional splitting, and multigrid schemes. In essence, the AMR algorithm should use a block structure for the refined grid. To date, only a few methodologies of this nature have been proposed. The method of Berger and Oliger[1] is one of the earliest. In their method, the refined grids are allowed to overlay the underlying coarse grids in an arbitrary manner. The blocks of the refined grids are constructed in physical space based on the shape and size of the region to be refined. The resulting grids tend to align with discontinuities and other features that determine the shape and size of the region. Although this approach has not been widely adopted, the work systematically addressed many important issues related to adaptation of structured grids via local refinement, such as proper nesting of the grid levels and a suitable time-stepping scheme for multi-level grids.

Building on the work of Berger and Oliger, Berger and Colella[2] devised a methodology in which the refined grids conform with the coarse grids, i.e., the boundaries of the fine-grid blocks are made

to coincide with grid lines of the coarse grid. The block structure for the refined grids is created using a special algorithm that operates purely in the index space of the coarse grid. The algorithm fits topologically rectangular patches over the regions of the coarse grids where the error in the solution is estimated to be above a specified threshold value. The blocks on the refined grids are then created by subdividing the coarse grid cells within each of the patches. Advantages of the new approach, compared to that of Berger and Oliger, include greatly simplified prolongation and restriction operators for transferring data between a coarse grid and a refined grid, and rigorously enforced conservation at interfaces between coarse and fine grids. This approach has been used extensively for Cartesian grids (see e.g. Ref. 2–5) but only to a limited extent for body-fitted structured grids (see Ref. 6 to 8). Algorithms of similar nature to those by Berger and co-workers have been suggested by other authors but have not been as fully developed (see Ref. 1 for a review). Also, Quirk[9] developed an AMR algorithm, one very similar to that of Berger and Colella, to locally refine Cartesian grid systems.

All the approaches reviewed above have in common that the block-structure of the refined grid is determined automatically at run time when the solution is computed. A different approach was proposed by Davis and Dannenhoffer[10]. In this approach, the entire structured grid system is divided up into sub-blocks of uniform size and dimensions. During adaptation, each sub-block is refined either in its entirety or not at all. In a recursive manner, a sub-block that has been refined is itself divided into sub-blocks, each of which can be refined. In the algorithm of Davis and Dannenhoffer, directional refinement is used, i.e., the grid can be refined in only one, two or three directions as desired.

The methodology used in the present work is based on the AMR algorithm of Berger and Colella. It inevitably resembles the methodology used in Ref. 7, although important differences exist. The present approach is described in some details in Ref. 8. Here, only an outline is given. Results are presented for two inviscid flows, a transonic flow over a NACA0012 airfoil (AGARD-03 test case) and a reflection of a shock over a double wedge.

AMR ALGORITHM—SPECIAL ISSUES FOR STRUCTURED GRIDS

In the AMR algorithm of Berger and Colella, the solution exists on several *levels* of finer and finer meshes which form an hierarchical structure. Each *mesh level*, excluding the coarsest which covers the entire domain, is embedded within the next coarser level and is created where high resolution is required by refining cells on that coarser level. The AMR algorithm performs two major tasks, i.e., to create and maintain hierarchy of mesh levels and to advance in time the solution on the hierarchy. Both are described in detail in Ref. 2. The present adaptation of the algorithm to body-fitted grids is described in Ref. 8. Here, only an outline of the AMR algorithm is given and some special issues related to curvilinear grid systems are pointed out.

Assuming a time-accurate discretization, the AMR algorithm refines simultaneously in space and time by an integer refinement ratio r. Refinement is done where an error estimation procedure based on Richardson extrapolation determines that error is above a specified tolerance. The refined cells are organized into a small number or rectangular blocks, the union of which make up the mesh level. The solution on the hierarchy of mesh levels is advanced in time recursively, level by level, such that every time the solution on a coarse mesh level is advanced by a time step Δt, the solution on the next finer level is advanced r times by a time step $\Delta t/r$. After the r fine grid time steps,

the solution on the fine mesh level is restricted to the underlying coarse grid. At interfaces between coarse and fine grids, boundary conditions for the fine grid are obtained by interpolating the solution on the coarse grid. Conservation on the interface is ensured by agglomorating, through the r time steps on the fine grid, the boundary fluxes on the fine grid that correspond to each coarse grid cell adjacent to the interface, and in a special "refluxing" step, correcting the solution in the adjacent coarse grid cell by the difference between the agglomorated flux and the original flux computed on the coarse grid. This treatment of the interface ensures that under mesh refinement the numerical solution converges to a weak solution of the governing equations.[11]

The AMR algorithm of Berger and Colella operates purely in the index space of the grid system rather than the physical space. Thus, it applies equally to structured body-fitted grid systems and Cartesian grids. Nonetheless, when the algorithm is applied to body-fitted grid systems, special issues arise as refined curvilinear grid systems must be generated from an existing coarser curvilinear grid. This grid refinement must be done carefully to ensure sufficiently smooth grids on all levels of refinement. In this study, the grid refinement is done by combining parametric cubic spline interpolation and Hermite interpolation. The cubic splines, here natural cubic splines, are used to "reconstruct" the grid lines from the discrete grid points. The interpolation is done grid line by grid line and produces cubic polynomials that bridge between any two neighboring grid points on a grid line. Then, the Hermite interpolation is used to bridge between the four cubic polynomials that define the edges of the coarse grid cell and to define the grid points of the refined grids. Since the polynomials describing the shape of the edges of the cell were constructed using cubic splines, the overall refined grid system, obtained by refining the coarse grid cell by cell, will be smooth and at least C^1 continuous. In the flow solver, the proper shape of the cells (*i.e.*, cubic polynomials) is taken into account when cell areas are computed. Thus, the total area of fine grid cells created by refining a single coarse grid cell will always equal that of the coarse grid cell. This greatly simplifies communication between grids on different levels of refinement over the case when straight-side cells are assumed.

For greater details on the AMR algorithm for body-fitted structured grids, see Ref. 8 as well as Ref. 2.

OBJECT ORIENTED IMPLEMENTATION

The methodology described in this paper has been implemented using mixed language programming. A driver module for the AMR algorithm was written in the C++ programming language while all routines performing floating point intensive parts of the algorithm (e.g., the approximate Riemann solver) were written in FORTRAN. This implementation is possible due to the modularity of the AMR algorithm and allows one to take advantage of the strengths of the different programming languages. Here, the strengths of C++, including object-oriented capability, flexible data structures and dynamic memory allocation, make that language very effective for the implementation of a driver for the AMR algorithm while the extensive optimization by FORTRAN compilers of floating point operations on array data structures makes FORTRAN the language of choice for most of the compute-intensive parts. The implementation makes extensive use of the AMR *library* developed by Crutchfield and Welcome.[12] The AMR library is written in C++ and FORTRAN, and is a collection of space-dimension independent *classes* specially designed to aid in implementation of schemes employing the AMR algorithm.

In the present implementation for structured grids, the main objects manipulated by the AMR driver module are instances of *classes* called *LevelBlock* and *MeshLevel*. A *LevelBlock* consists of all data for a single block from a block structured grid and a set of routines that essentially form a self-contained generalized single block flow solver. A *MeshLevel* is a collection of *LevelBlocks* which form a multiblock grid system on a single level of refinement, along with routines that control communication between blocks on the level, communication between coarse and fine levels, and time stepping scheme for the levels. A doubly-linked list of *MeshLevels* forms the hierarchical, adaptively refined grid structure on which computations are performed.

RESULTS

Results are presented for two of the test cases proposed for this workshop, namely, the AGARD-03 test case of a transonic flow over a NACA0012 airfoil, and a reflection of a plane shock over a double wedge. Both cases are inviscid. The computations presented here were done using a discretization of the Euler equations based on the multi-dimensional upwind scheme of Colella.[13]

The first results to be presented are for steady flow over a NACA0012 airfoil at free-stream Mach number of 0.95 and zero angle of attack. This case is a surprisingly tough test case for AMR algorithms as the location of the normal shock that forms behind the airfoil is very sensitive to the proper capturing of the smooth expansion that takes place in the flow at the leading edge of the airfoil.[15] The starting grid system used in the computation was a C–grid with 32 cells on half the airfoil surface, 64 cells from the trailing edge to the outflow boundary, and 32 cells from the airfoil surface to the free-stream boundary. The outer limits of the grid were 100 cords away. The grid in the neighborhood of the airfoil is shown in Fig. 1. In the present computations, the flow conditions at the free-stream boundary were fixed while all variables are extrapolated at the outflow boundary. This treatment of the boundary conditions has the potential to affect the accuracy of the computed solutions. However, the large distance from the airfoil to the free-stream surface does to some extent compensate for the treatment at the boundary. The sensitivity of the solution to the distance to the free-stream boundary has not been studied.

The flow over the airfoil was computed using zero to three levels of refinement with refinement ratio $r = 2$. Figures 2-3 show the computed solution obtained using three levels of refinement. Apparent in Fig. 3 is a low amplitude oscillation behind the oblique shock. This oscillation is due to the low level of artificial dissipation in the present discretization. According to the computations, the location of the normal shock trailing the airfoil, based on where the Mach number on the symmetry line is unity, is at 3.456 cords, 3.346 chords, 3.279 chords, and 3.274, for zero to three levels of refinement, respectively. In comparison, Ref. 14 reports values between 3.32 and 3.35 chords as the correct location. Thus, the error in the solution on the finest grid is within 2%. In the computation shown here, less than 20% of the cells on the coarsest grid were refined to the finest level.

The second test case to be presented is the reflection of a plane shock over a concave double wedge. The wedge angles are 20° and 50°. The shock Mach number is 2.16. An ideal gas with specific heats ratio of 1.4 is assumed. Under these conditions, a Mach reflection is formed as the plane shock hits the first wedge. A second Mach reflection is formed as the Mach stem of the first reflection hits the second wedge. The triple point of the second reflection travels faster in the direction parallel to the plane shock than the triple point of the first reflection and gradually overtakes it. Complicated interactions take place as discontinuities of the two reflections intersect.

Computations were done using two levels of refinement with refinement ratio $r = 4$. The starting grid system used in the computations is shown in Fig. 4. Figures 5 and 6 show the computed solution at two instances in time. Figure 5 shows the solution shortly after the Mach stem of the first reflection has reflected off the second wedge. At the specific moment shown, the reflected shock from the second reflection is about to overtake the slip line emanating from the triple point of the first reflection. In Fig. 6, the triple point of the second reflection has overtaken the triple point of the first reflection.

The computed solution for the double Mach reflection case has not been compared in specific details to experimental data. However, at least qualitatively, the results compare very well to experimental results by Itoh, et al.[15] At any time in the simulations, only a small percentage of the coarse grid cells were refined to the finest level. Consequently, the overall cost of the simulation was only a small fraction of the cost of a simulation done using a single-level grid with the same resolution as the finest grid. This demonstrates how a simulation, intractable if traditional unadapted structured grids are used, can become feasible when adaptive refinement is used.

CONCLUSIONS

An algorithm for adaptive refinement of body-fitted structured grids has been tested on two of the benchmark cases of this work shop. The algorithm used is based on the AMR algorithm of Berger and Colella.[2] It uses a block structure for the data on the refined grids, which makes the algorithm very well suited for adaptive refinement of structured body fitted grid systems. The AMR algorithm has been found to work very well for the two test cases attempted and holds great promise for use in general purpose flow solvers employing structured grids.

REFERENCES

1 Berger, M. J. and Oliger, J., "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations," *J. of Comp. Physics*, Vol. 53, pp. 484-512, 1984.

2 Berger, M.J. and Colella, P., "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *J. of Comp. Physics*, Vol. 82, pp. 64-84, 1989.

3 Colella, P. and Henderson, L.F., "The von Neumann paradox for the diffraction of weak shock waves," *J. Fluid Mech.*, vol. 213, pp. 71-94, 1990.

4 Bell, J.B., Berger, M., Saltzman, J. and Welcome, M.L., "Three-Dimensional Adaptive Mesh Refinement for Hyperbolic Conservation Laws," *SIAM J. of Sci. and Stat. Computing*, vol. 15, no. 1, pp. 127-138, Jan., 1994.

5 Pember, R.B., Bell, J.B., Colella, P., Crutchfield, W.Y., and Welcome, M.L., "Adaptive Cartesian Grid Methods for Representing Geometry in Inviscid Compressible Flow," AIAA-93-3385-CP, Proceedings, 11th AIAA Computational Fluid Dynamics Conference, July, 1993.

6 Berger, M. J. and Jameson, A., "Automatic Adaptive Grid Refinement for the Euler Equations," *AIAA Journal*, Vol. 23, No. 4, pp. 561-568, April 1985.

7 Bell, J., Colella, P., Trangenstein, J., Welcome, M., "Adaptive Mesh Refinement on Moving Quadrilateral Grids," AIAA-89-1979-CP, Proceedings, AIAA 9th CFD Conference, 1989.

8 Steinthorsson, E., Modiano, D. and Colella, P., "Computations of Unsteady Viscous Flows Using Solution Adaptive Mesh Refinement in Curvilinear Body-Fitted Grid Systems" AIAA-94-2330, June, 1994. Also, ICOMP-94-17, NASA TM 106704.

9 Quirk, J.J., Dissertation, Cranfield Institute of Technology.

10 Davis, R. L. and Dannenhoffer, J. F., "Three-Dimensional Adaptive Grid Embedding Euler Technique," AIAA 93-0330, January 1993.

11 Berger, M.J., "On Conservation at Grid Interfaces," *SIAM J. Numer. Anal.*, Vol. 24, No. 5, pp. 967-984, October, 1987.

12 Crutchfield, W.Y. and Welcome, M.L., "Object Oriented Implementation of Adaptive Mesh Refinement Algorithm,", *Scientific Programming*, Vol. 2, number 4, winter 1993.

13 Colella, P., "Multidimensional Upwind Methods for Hyperbolic Conservation Laws," *J. of Comp. Physics*, Vol. 87, pp. 171-200, 1990.

14 Warren, G.P., Anderson, W.K., Thomas, J.L., and Krist, S.L., "Grid Convergence for Adaptive Methods," AIAA-92-1592; AIAA 10th Computational Fluid Dynamics Conference, June 24-26, 1991.

15 Itoh, K., Takayama, and Ben-Dor, G., "Numerical Simulation of a planar shock wave over a double wedge," *Int. J. for Num. Meth. in Fluids*, Vol. 13, pp. 1153-1170, 1991.
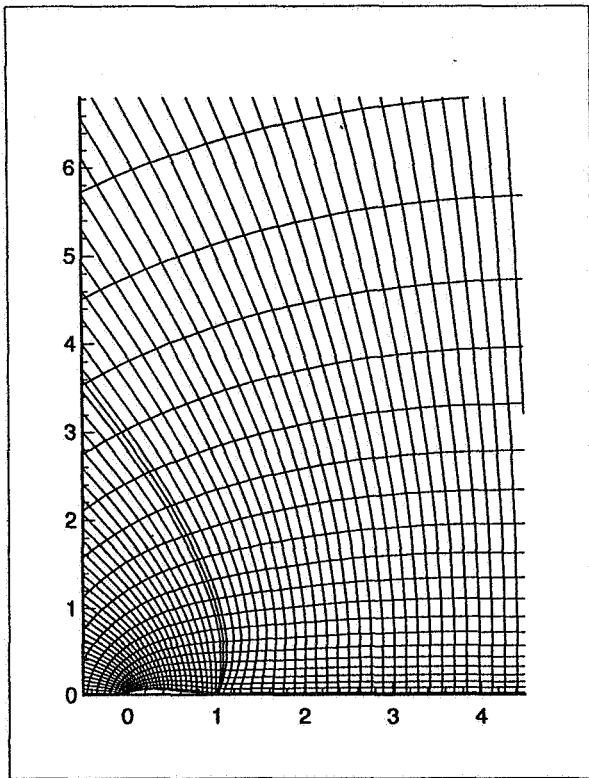
Figure 1. Grid system in two blocks for NACA0012 airfoil—grid lines drawn through cell centers and boundary points.
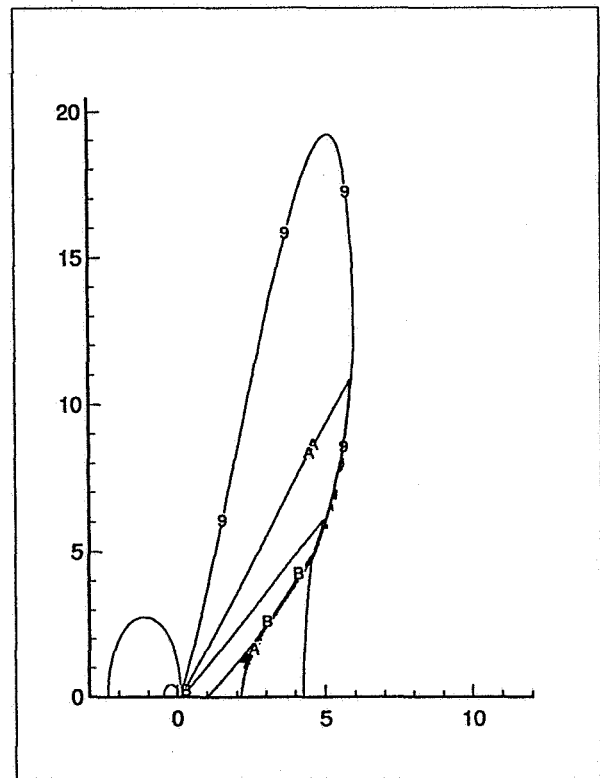

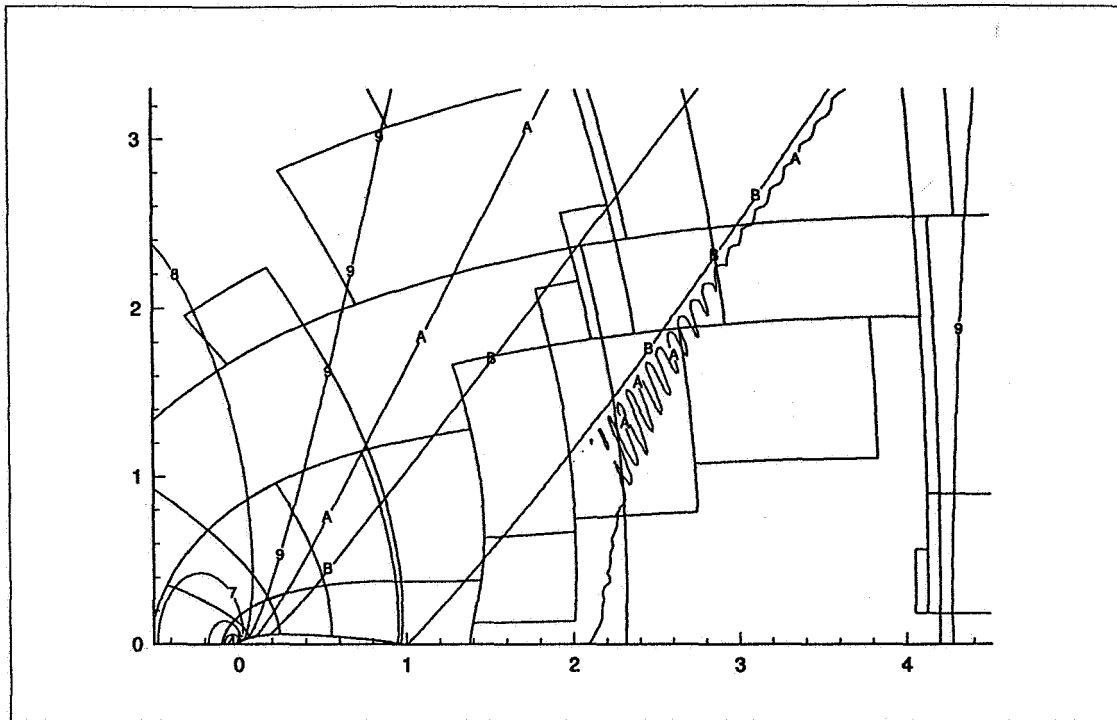Figure 2. Transonic flow over a NACA0012 airfoil at $M = 0.95$—contours of Mach number: $M_1 = 0.2$, $\Delta M = 0.1$.

Figure 3. Transonic flow over a NACA0012 airfoil at $M = 0.95$—contours of Mach number: $M_1 = 0.2$, $\Delta M = 0.1$ (boxes indicate boundaries of blocks on the two finest mesh levels).
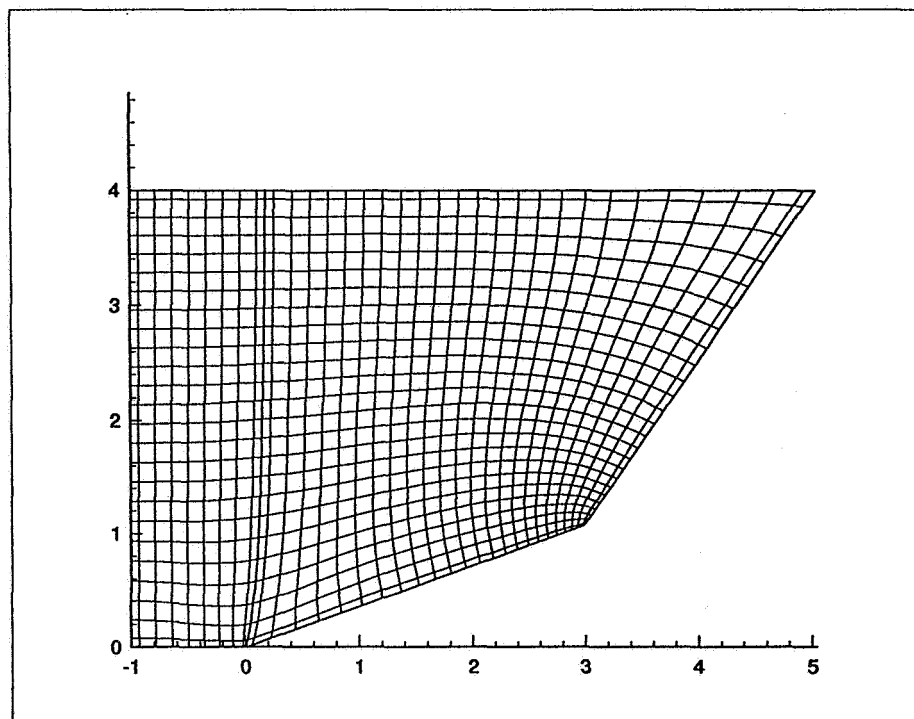

Figure 4. Grid system in two blocks for concave double wedge—grid lines drawn through cell centers and boundary points.
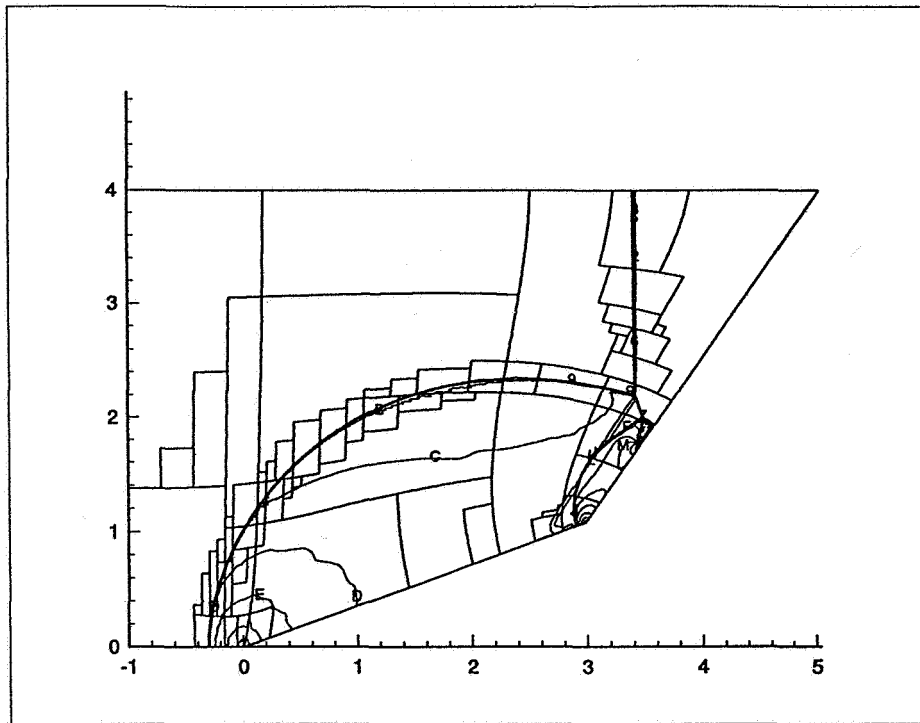
239

Figure 5. Shock reflection over a double wedge—Mach stem of first reflection has hit the second wedge. Contours of density; $\rho_1 = 1.35$, $\Delta\rho = 0.2$ (boxes indicate boundaries of blocks on the refined mesh levels).
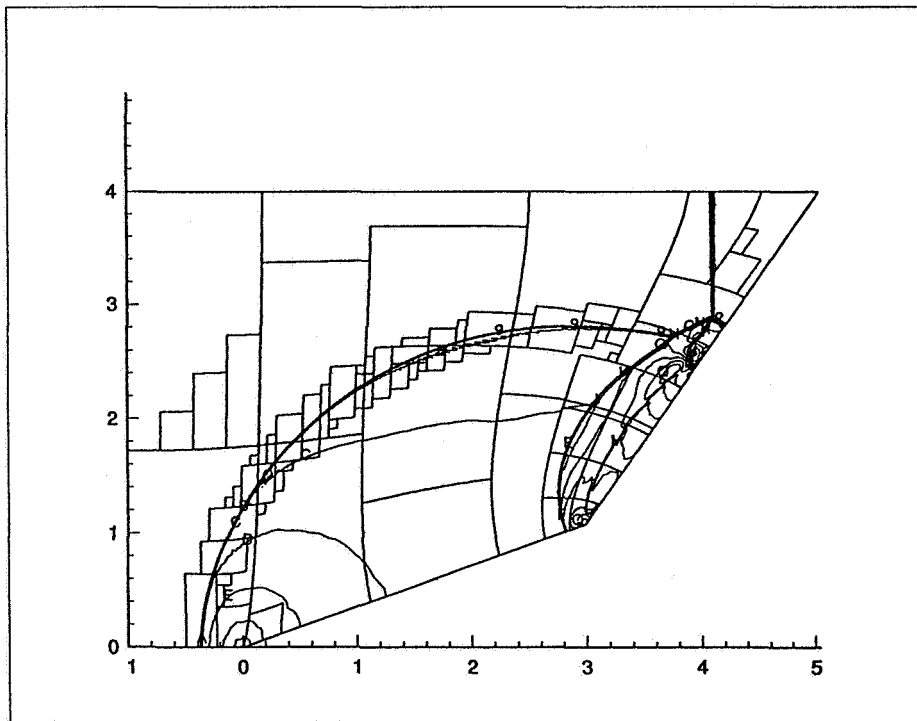

Figure 6. Shock reflection over a double wedge—triple point of second reflection has overtaken that of the first. Contours of density; $\rho_1 = 1.35$, $\Delta\rho = 0.2$ (boxes indicate boundaries of blocks on the refined mesh levels).

240