A.2.30

LBNL-41343

ERNEST ORLANDO LAWRENCE
BERKELEY NATIONAL LABORATORY

Steady-State Solution-Adaptive Euler Computations on Structured Grids

Scott A. Dudek and Philip Colella

Computing Sciences Directorate

January 1998

Presented at the
36th AIAA Aerospace Sciences Meeting and Exhibit,
Reno, NV,
January 12–15, 1998,
and to be published in
the Proceedings

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor The Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, or The Regents of the University of California.

Ernest Orlando Lawrence Berkeley National Laboratory
is an equal opportunity employer.

LBNL-41343

STEADY-STATE SOLUTION-ADAPTIVE EULER COMPUTATIONS ON STRUCTURED GRIDS

*Scott A. Dudek, and **Philip Colella

*Numerical Modeling Section
McDermott Technology, Inc.
Alliance, OH

and

**Applied Numerical Algorithms Group
Lawrence Berkeley National Laboratory
Berkeley, Ca

January 1998

This work was supported by the Office of Energy Research, Office of Computational and Technology Research, Mathematical, Information, and Computational Sciences Division, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

AIAA-98-0543

STEADY-STATE SOLUTION-ADAPTIVE EULER COMPUTATIONS ON STRUCTURED GRIDS

Scott A. Dudek*
Numerical Modeling Section
McDermott Technology, Inc.
Alliance, OH

and

Phillip Colella†
Applied Numerical Algorithms Group
Lawrence Berkeley National Laboratory
Berkeley, CA

Abstract

A local solution-adaptive mesh refinement algorithm is used to produce steady-state flow results on structured grids for the two-dimensional Euler equations. The solution is marched to steady state using an explicit, cell-centered, second-order unsplit multidimensional upwind method. Convergence is accelerated by local time stepping and a multigrid method. The flexibility and efficiency of the algorithm are shown by presenting three test cases, a variety of subsonic and transonic internal and external flows.

Introduction

An algorithm has been developed to compute steady-state solutions to the Euler equations using a multidimensional upwind method and local solution-adaptive mesh refinement on structured grids.

The second-order unsplit multidimensional upwind method of Colella [1], as modified by Dudek and Colella for steady-state flows [2], is used to calculate the convective fluxes of the Euler equations. This method has been used in many contexts [3, 4, 5, 6], so it is well-tested and robust.

The adaptive mesh refinement (AMR) algorithm allows computational resources to be utilized efficiently by placing refined grids only in areas in which they are needed. Thus, excess computations and memory are not wasted. In addition, the AMR algorithm gives great flexibility in problem solving, since it is not necessary to know beforehand where resolution will be needed. For instance, refinement around shocks is handled automatically, and the shock locations do not need to be known *a priori*. Thus, a simple base grid can be created and the AMR algorithm will ensure proper refinement where necessary. Since the grids are structured, the data structures are simply ordered, and optimization is straightforward, particularly on vector machines.

The local adaptive mesh refinement method developed by Berger and Oliger [7] uses a sequence of nested levels of refined structured grids. After a solution is computed on the hierarchy of meshes, an error estimate is calculated, and blocks of cells where this error is high are refined locally in an efficient manner to produce a new hierarchy of levels. The solution is then computed on this new hierarchy and the process continues. Originally developed in the context of hyperbolic conservation laws in two-dimensions [3, 7], the algorithm has been extended to three dimensions [8], viscous two-dimensional flows on mapped grids [4], and incompressible flows [5].

There is extensive infrastructure for and experience with adaptive mesh refinement in conjunction with multidimensional upwind methods. In fact, a C++ library of functions which are used in adaptive finite difference calculations has been developed and is widely used [4, 9]. The motivation for this research is to take this well-established multidimensional upwind method and adaptive mesh refinement machinery and to combine them into an algorithm which can produce efficient and accurate steady-state solutions on struc-

*Research Engineer, Member AIAA.
†Group Leader, Member AIAA. Also: Professor in Residence, Department of Mechanical Engineering, The University of California, Berkeley.

Copyright © 1998 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

tured grids.

This paper is a continuation of work begun by Berger and Jameson [10], who were the first to implement block-structured local adaptive mesh refinement on body-fitted mapped grids. We have modified and extended their work in a number of areas. They solved the two-dimensional Euler equations to steady state using the centered space differencing, Runge-Kutta time stepping algorithm of Jameson, Schmidt, and Turkel [14]. We advance the solution using an unsplit multidimensional upwind method with local time stepping [2]. We also take advantage of multigrid convergence acceleration, both at each level of refinement and over the entire mesh hierarchy. In addition, the grid generation process has been simplified and automated. Berger and Jameson required a fully refined grid at the finest level of refinement, from which all coarser grids were produced. Not only does this require an *a priori* knowledge of the amount of refinement needed, but this procedure may be difficult and expensive for complex geometries. In our approach, creation of refined body-fitted grids only requires the coarse base grid, from which the refined grids are interpolated only to the extent that they are needed, rather than over the entire domain. Finally, Berger and Jameson applied their algorithm to a limited number of non-lifting external flow test cases, using only refinement ratios of two. Our algorithm has been tested on a wide variety of internal and external (both lifting and non-lifting) flows, using a variety of refinement ratios between grid levels.

Governing Equations

The two-dimensional time-dependent Euler equations for inviscid fluid flow in integral form for a control volume Ω with boundary $\partial\Omega$ are

$$\frac{\partial}{\partial t}\int_{\Omega} U\,d\mathbf{x} + \oint_{\partial\Omega} \mathbf{F}(U)\cdot\mathbf{n}\,dS = 0, \qquad (1)$$

where $\mathbf{n}$ is an outward-pointing normal. The variable U is the array of conserved quantities (mass, momentum in x-direction, momentum in y-direction, and energy), and $\mathbf{F}(U) = (F^x(U), F^y(U))$ is the vector of inviscid fluxes in the x- and y-directions:

$$U(\mathbf{x},t) = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix},$$

$$F^x(U) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uE + up \end{pmatrix},$$

$$F^y(U) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vE + vp \end{pmatrix}.$$

The density is denoted by ρ, the velocity in the x- and y-directions by $\mathbf{u} = (u, v)$, the total energy per unit mass by E, and the pressure p is given by the equation of state for a perfect gas,

$$p = (\gamma - 1)\left(\rho E - \rho\left(\frac{u^2+v^2}{2}\right)\right),$$

where γ is the ratio of specific heats.

Multidimensional Upwind Method

Consider a structured-grid cell (i, j) as shown in Figure 1. Let us define $U^n_{i,j}$ to be a discrete approximation to the average of the vector of conserved variables U over the cell at time t^n:

$$U^n_{i,j} = \frac{1}{\sigma_{i,j}}\int_{(i,j)} U(\mathbf{x}, t^n)d\mathbf{x},$$

where $\sigma_{i,j}$ is the area of cell (i, j), and $\mathbf{x} = (x, y)$. Using this definition, we then discretize the integral equations (1) using a simple forward difference in time, to determine the value of the solution at time t^{n+1}:

$$\begin{aligned} U^{n+1}_{i,j} = U^n_{i,j} - \frac{\Delta t}{\sigma_{i,j}}\big(&\mathbf{F}_{i+1/2,j}\cdot\mathbf{n}_{i+1/2,j} \\ &+ \mathbf{F}_{i-1/2,j}\cdot\mathbf{n}_{i-1/2,j} \\ &+ \mathbf{F}_{i,j+1/2}\cdot\mathbf{n}_{i,j+1/2} \\ &+ \mathbf{F}_{i,j-1/2}\cdot\mathbf{n}_{i,j-1/2}\big), \end{aligned} \qquad (2)$$

where $\Delta t = t^{n+1} - t^n$, $\mathbf{n}$ is an outward-facing area normal at the cell edges, and $\mathbf{F}$ is an approximation to the flux at the cell edge.

The convective fluxes $\mathbf{F}\cdot\mathbf{n}$ are calculated by a version of the multidimensional upwind method of Colella [1], modified for steady-state flows by Dudek and Colella [2]. In this second-order two-dimensional Godunov method, a pair of edge states are extrapolated from adjacent cell-centers, taking into account the multidimensional nature of the flow. A Riemann problem is then approximately solved to uniquely define a state at the cell edge, from which the fluxes are computed.

To prevent oscillations in the solution, the slopes used in this extrapolation are limited using the van Albada limiter [11]. Since we are interested in the steady state solution, a local time step is used. Each cell uses the maximum allowable time step based on

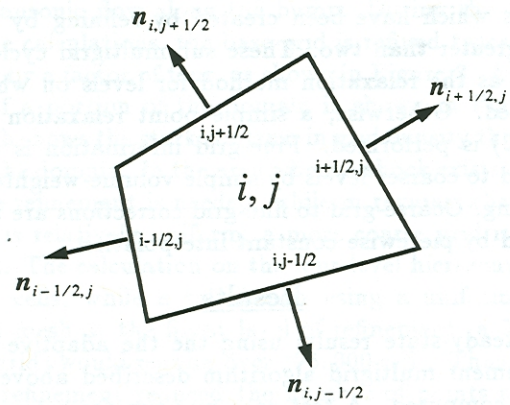


Figure 1: Structured Grid Cell (i, j) and Notation

a CFL constraint. In addition, it has been found necessary to modify the Godunov method in regions of transonic shocks, where the use of the Godunov flux resulted in poor convergence. Therefore, an Enquist-Osher flux [12, 13] is used in transonic regions, where the smooth flux function results in better convergence. Finally, a blended artificial diffusion term of second- and fourth-differences is added to the flux to damp acoustic waves present in the solution of the Euler equations, which allows convergence to steady-state. The dissipation is of the form presented in Jameson, Schmidt, and Turkel [14] and does not affect the accuracy of the method.

Adaptive Mesh Refinement

The adaptive mesh refinement method automatically places grid points in areas where they are deemed most necessary to adequately resolve relevant flow features. This allows efficient use of computational resources by providing sufficient resolution with a minimal number of cells.

In this work, we use the local block-structured adaptive mesh refinement (AMR) method of Berger and Colella [3], with the appropriate modifications for mapped grids described by Bell, et. al. [15] and Steinthorsson, et. al. [4]. The AMR algorithm we use to solve the two-dimensional Euler equations for steady state flows on structured grids is described in detail by Dudek [16], so only the main points are reviewed.

Refined logically-rectangular regions are recursively embedded within coarser grids, covering groups of coarser cells which need refinement, until a desired solution accuracy is achieved. The discretization on the hierarchy of refined levels is an extension of the cell-centered multidimensional upwind algorithm described above for steady-state flows. To ensure conservation between levels of refinement, we enforce a flux matching condition at these interfaces. Additionally, a multilevel multigrid method is used to accelerate the solution on the grid hierarchy to steady state. This is a straightforward extension of multigrid for a single grid, except that the update step at each level consists of a multigrid sub-cycle between levels of refinement.

Ferziger [17] argues that, for adaptive schemes, the truncation error should be driven to zero, since the truncation error acts as the source of the solution error, and because it is more localized than the solution error. This is the strategy we have used, forcing the truncation error uniformly below a specified tolerance. Once the solution is obtained on the grid hierarchy, a measure of the local truncation error is computed in a manner similar to the approach of Berger and Jameson [10]. Cells for which the truncation error estimate is above a specified tolerance are tagged, clustered together and refined [18].

The refined body-fitted grid hierarchy is produced by fitting a bicubic spline to the base grid to create a smooth transformation between index space and physical space. The spline is computed once, initially, and then used to interpolate finer grids from the base grid. Because the bicubic spline is C^2, interpolated finer grids are sufficiently smooth to ensure that the grid does not adversely affect local truncation error estimates.

After refinement, the solution is computed on the new mesh hierarchy, and the process continues until the error estimate for all cells is below the specified tolerance or a pre-defined number of levels of refinement is reached.

Boundary Conditions

The use of a fourth-difference dissipation results in a 13-point stencil for the operator. The boundary conditions are satisfied by the utilization of two rows of cell-centered "ghost" or "fictitious" cells outside of the grid interior.

A grid's ghost cells will be filled differently depending on the type of boundary with which they are associated. For cells outside of the domain boundary, "physical" boundary conditions will be applied: solid wall, inflow, outflow, far-field, and periodic boundary conditions [2]. For grids which are embedded within the mesh hierarchy, there are two types of "interior" boundary conditions. If a grid abuts another grid at the same level of refinement, the ghost cells are merely copied from interior cells of the neighbor. If the grid does not abut a domain boundary or another grid at the same level, then the boundary conditions must be interpolated from the next coarser level. We use a

quadratic interpolation along the coarse/fine boundaries.

Convergence Acceleration

Two methods of convergence acceleration are used: local time stepping, and multigrid.

Local Time-Stepping

When using the time-accurate multidimensional upwind algorithm, the time step Δt is the smallest allowable time step over the entire domain. Since Δt is determined by restricting the CFL number to be less than unity, it is controlled by the smallest cell. This results in extremely slow convergence to a steady-state solution, which can be traced to the low damping of high wave-number modes in cells where the effective CFL number is quite small. This feature makes the use of a multigrid convergence acceleration method ineffective, since it requires good smoothing of high wave-number modes to be successful. This is especially apparent for external flows, where the cell areas along the far-field boundary can be orders of magnitude greater than those along the body.

Furthermore, because the computed edge states depend on the time step, the convective fluxes are a function of Δt as well as the conserved quantities U^n. That is, $F = F(\Delta t, U^n)$. This dependence of the flux, and hence the steady-state solution, on the time step has disturbed some authors. They question the validity of a steady-state solution which depends on a global quantity, here the global time step.

As a solution to both of the above problems, we use a local time step within the upwind method described above. This means that each cell has its own time step, determined by the CFL restriction for that cell. This destroys time-accuracy, but allows waves to reach the boundary more quickly. It also allows proper damping of high wave-number modes which, when combined with the necessary artificial diffusion, allows an implementation of multigrid to be successful.

Furthermore, the use of local time steps means that the fluxes and hence the steady-state solution no longer depend on a global quantity. A local truncation error analysis shows that the scheme is second-order accurate in the local time and spatial steps.

Multigrid

In addition to local time-stepping, we accelerate convergence by use of a multigrid method over the entire hierarchy of refined grids. This method is a cell-centered adaptation of a nodal-point method given by Almgren, et. al. [19]. In conjunction with this, we use a nonlinear multigrid method of the form presented in the text of Wesseling [20] for the base level and for levels which have been created by refining by a factor greater than two. These sub-multigrid cycles are used as the relaxation method for levels on which it is used. Otherwise, a simple point relaxation given by (2) is performed. Fine-grid information is transferred to coarser levels by simple volume-weighted averaging. Coarse-grid to fine-grid corrections are transferred by piecewise constant interpolation.

Results

Steady-state results using the the adaptive mesh refinement multigrid algorithm described above have been computed. A test case was run for a transonic internal channel flow over a 10% sine-squared bump. External lifting flows were modeled for subsonic flow over a NACA 0012 airfoil and for transonic flow over an RAE 2822 airfoil.

For all of the results shown, a W multigrid cycle was employed for both the overall multilevel multigrid cycle and the level multigrid sub-cycle. Four update steps were taken during the downward portion of the cycle, and two updates during the upward portion. This cycling strategy was found to provide optimal convergence. When the max norm of the residual for each of the conservation equations was reduced by ten orders of magnitude, we considered the solution to have reached convergence. In the discussion below, the term "work unit" is used to mean the amount of computational effort to update the solution one time step on the entire mesh hierarchy.

Unless stated otherwise, the values for the coefficients of artificial dissipation are the same for all cases. On the finest multigrid levels, the coefficient of fourth difference dissipation is $\kappa^{(4)} = 1/64$, and the coefficient of second difference dissipation is $\kappa^{(2)} = 0$ for subsonic flows, and $\kappa^{(2)} = 1/4$ for transonic flows. These values were found to provide optimal convergence. For coarser multigrid sub levels, $\kappa^{(2)}_{\text{crse}} = 0.05$ for both subsonic and transonic flows, and no fourth-difference dissipation is used. Furthermore, the CFL number is taken to be 0.5 for all cases, and the ratio of specific heats is that for air, $\gamma = 1.4$.

Internal Flow

An internal flow case was run for flow over a 10% sine-squared bump in the bottom of a channel. The physical domain is $0 \leq x \leq 3$, $0 \leq y \leq 1$, and is initially covered by a mesh containing 48 cells in the flow direction and 16 cells in the transverse direction (see Fig. 2). The flow is initialized to the inflow boundary values of density and velocity, and the outflow value of pressure, which are chosen to prescribe an incoming Mach number.

The inflow Mach number is 0.675, which results in transonic flow along the bump. During the course of the calculations, the base grid is refined twice, each time by a factor of four, as shown in Figure 3. A close-up of a portion of the domain is shown in Figure 4, which shows the stark contrast in grid density throughout the domain. In the area where a shock exists, much more refinement is needed, while in regions where the flow is relatively uniform, a more coarse grid is sufficient. The calculation on the four-level hierarchy used 7488 cells, while a calculation using a uniformly refined mesh at the finest level of refinement (a 768 by 256 grid) would require over 196,000 cells. Thus adaptive refinement reduced the number of points needed by a factor of more than 26, resulting in a substantial savings in both memory and computations.

Figure 5 shows the steady-state Mach number distribution on the level hierarchy. (The hierarchy of levels is outlined by lines thinner than the contours.) The algorithm adaptively adds grid points in order to sharply resolve the shock.

The convergence history for the max-norm of the density residual for each of the level hierarchies is shown in Figure 6. For the most refined hierarchy, it takes 972 work units (46 multigrid cycles) for the residual norm to drop ten orders of magnitude. Also shown in Figure 6 is a plot of the coefficient of pressure,

$$C_p = \frac{p - p_{\text{outflow}}}{\frac{1}{2}\rho_{\text{inflow}}|\mathbf{u}_{\text{inflow}}|^2},$$

along the bottom surface of the domain. The grid density around the shock results in its steep representation.

External Flow

Two different external flow test cases were run: subsonic flow over a NACA 0012 airfoil and transonic flow over an RAE 2822 airfoil. For both cases, the initial base grid is a single structured O-grid with a periodic boundary along the trailing edge. Each grid has 32 cells in the radial direction and 64 cells in the circumferential direction and extends to a circle with radius of 20 chord lengths. The base grid is non-clustered since the adaptive refinement process will provide any needed clustering. The flow is initialized to the far-field values of density, velocity, and pressure, which are chosen to produce a specified Mach number at the far field. A circulation correction is applied at the far-field boundary in order to obtain the proper lift coefficient [21].

For the subsonic NACA 0012 case, the airfoil is at an angle of attack, $\alpha = 3°$, and the far-field Mach number is 0.5. Calculations begin on the base grid and three refinements take place as the adaptive process proceeds, with refinement ratios of two, two, and four, respectively. The evolution of the grids used in the adaptive process is shown in Figure 7. (Because of the density of grid points, the most refined level appears solid black.) The final hierarchy of grids consists of 24,224 cells. Over 524,000 cells would be required to obtain the same resolution using a fully-refined grid over the entire domain.

Levels	Cells	C_L	C_D
1	2048	0.4077	0.01198
2	3008	0.4254	0.00283
3	5472	0.4311	0.00065
4	24224	0.4325	0.00002

Table 1: C_L and C_D: NACA 0012, $\alpha = 3°$, $M_\infty = 0.5$

The Mach number distribution is presented in Figure 8. The grid hierarchy is outlined by lines thinner than the contour lines. The Mach number at the leading edge is shown in Figure 9. The stagnation region is clearly defined.

The coefficient of pressure,

$$C_p = (p - p_\infty) / \left(\frac{1}{2}\rho_\infty |\mathbf{u}_\infty|^2\right),$$

along the airfoil surface is presented in Figure 10. The coefficients of lift and drag are defined:

$$C_L = \frac{L}{\frac{1}{2}\rho_\infty |\mathbf{u}_\infty|^2 A}, \quad C_D = \frac{D}{\frac{1}{2}\rho_\infty |\mathbf{u}_\infty|^2 A},$$

where L is the lift force, D is the drag force, and A is taken to be the chord length, which in our cases is always unity. Table 1 shows the values of C_L and C_D for each level hierarchy in the adaptive process. Convergence is evident for both quantities as we refine, with C_D converging to zero. At four levels of refinement, $C_L = 0.4325$. Jameson and Yoon [22] calculate $C_L = 0.4231$ for a mesh with 4096 cells. This compares best with our result for two levels of refinement.

The outstanding convergence characteristics provided by multigrid are shown in Figure 11, where the convergence history of the max norm of the density equation residual for each of the level hierarchies leading up to the final hierarchy are presented. For the four-level hierarchy, 1852 work units (28 multigrid iterations) are needed to drop the residual norm by ten orders of magnitude.

The second external flow case run was an RAE 2822 airfoil at 3° angle of attack. The far field Mach number is 0.75, which results in transonic flow. This is AGARD Test Case 06 [23].

Levels	Cells	C_L	C_D
1	2048	1.062	0.04968
2	3776	1.089	0.04612
3	8384	1.093	0.04473
4	17984	1.093	0.04447

Table 2: C_L and C_D: RAE 2822, $\alpha = 3°$, $M_\infty = 0.75$

The computations are begun on the base grid shown in Figure 12. This grid is refined three times, each time by a factor of two. The four-level hierarchy is shown in Figure 13, with a close-up of the leading-edge region shown in Figure 14. The total number of cells used is 17,984, while a fully refined grid at the highest level of refinement would require about 131,072 cells, a savings of over a factor of seven. This compares favorably with the most dense grid used in the AGARD study, which consisted of 20,480 grid points, all at the same level of refinement. Using less grid points, we are able to get higher resolution in the areas where it is needed without the need for *a priori* clustering.

Figure 15 displays steady-state Mach number contours for the final level hierarchy. The coefficient of pressure along the surface of the airfoil is shown in Figure 16, which is in good qualitative agreement with the results presented in AGARD report 211 [23]. The converged values of the coefficients of lift and drag are shown in Table 2. On the four-level hierarchy, the values are $C_L = 1.093$ and $C_D = 0.04447$, both in good agreement with the AGARD results. The convergence history of the max-norm of the density residual is shown in Figure 17. For the four-level hierarchy, 3624 work units (61 multigrid iterations) are needed to drop the residual norm by ten orders of magnitude.

Conclusions

A local block-structured adaptive mesh refinement algorithm has been used to compute solutions to the two-dimensional Euler equations on structured grids. A multi-dimensional upwind algorithm was used to march the solution in time. Both local time stepping and multigrid were used to accelerate convergence.

Test cases were run for both internal and external flows. The results show that the use of adaptive mesh refinement significantly reduces the number of cells needed for proper resolution of the flow without inhibiting convergence of the solution to steady state.

Furthermore, the use of adaptive mesh refinement changes the well-established grid generation paradigm and in fact simplifies the grid generation process. Traditionally, the creation of a structured grid system for complex domain proceeds as follows. First, a grid is generated with clustering in regions where it is felt that refinement is necessary to resolve a flow feature such as a boundary layer or shock. It may take several iterations with a grid generator before an acceptable grid is produced, in terms of skewness and smoothness. Then the flow simulation is run. If the flow parameters are changed, such as the Mach number or angle of attack, a grid with different clustering will likely be required.

With adaptive mesh refinement, this process is simplified considerably. Initially, a relatively coarse base mesh can be produced with little or no clustering. Typically, this is not difficult to generate. Any clustering needed to resolve the solution is automatically generated by the adaptive algorithm. If the flow parameters are modified, the same base mesh can be used. The error estimation and adaptive process will determine where to place refined grids.

Acknowledgements

The authors gratefully acknowledge the support of this work by the following grants.

At UC Berkeley: "Adaptive Numerical Methods for Partial Differential Equations," US Department of Energy Mathematical, Computing, and Information Sciences Division, Grant DE-FG03-94ER25205; and "Computational Fluid Dynamics and Combustion Dynamics," US Department of Energy High-Performance Computing and Communications Grand Challenge Program, Grant DE-FG03-92ER25140.

At LBNL: US Department of Energy Mathematical, Information, and Computing Sciences Division, Contract No. DE-AC03-76SF00098.

The first author thanks Dr. Lou Povinelli and Professor Ted Keith of The Institute for Computational Mechanics in Propulsion at the NASA Lewis Research Center for the use of facilities and computational resources.

References

[1] Colella, P., "Multidimensional Upwind Methods for Hyperbolic Conservation Laws," *J. Comput. Phys.*, Vol. 87, pp. 171–200, 1990.

[2] Dudek, S.A., and Colella, P., "A Godunov Steady-State Euler Solver for Structured Grids," AIAA–97–0875, January 1997.

[3] Berger, M.J. and Colella, P., "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *J. Comput. Phys.*, Vol. 82, pp. 64-84, 1989.

[4] Steinthorsson, E., Modiano, D., and Colella, P., "Computations of Unsteady Viscous Compressible Flows Using Adaptive Mesh Refinement in

Curvilinear Body-Fitted Grid Systems," AIAA-94-2330.

[5] Almgren, A.S., Bell, J.B., Colella, P., Howell, L.H., Welcome, M.L., "A Conservative Adaptive Projection Method of the Variable Density Incompressible Navier-Stokes Equations," to appear in *J. Comput. Phys.*

[6] Bell, J.B., Colella, P., Glaz, H.M., "A Second-Order Projection Method for the Incompressible Navier-Stokes Equations," *J. Comput. Phys.*, Vol 85, pp. 257–283, December 1989.

[7] Berger, M. J., and Oliger, J., "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations," *J. Comput. Phys.*, Vol. 53, pp. 484–512, 1984.

[8] Bell, J., Berger, M., Saltzman, J., and Welcome, M., "Three-dimensional Adaptive Mesh Refinement for Hyperbolic Conservation Laws," *SIAM J. Sci. Comput.*, Vol. 15, No. 1, pp. 127–138, January 1994.

[9] Crutchfield, W.Y. and Welcome, M.L., "Object Oriented Implementation of Adaptive Mesh Refinement Algorithm," *Scientific Programming*, Vol. 2, No. 4, winter 1993.

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

[11] van Albada, G.D., van Leer, B., and Roberts, W.W., Jr., "A Comparative Study of Computational Methods in Cosmic Gas Dynamics," *Astron. Astrophys.*, Vol. 108, 1982, pp. 76–84.

[12] Enquist, B., and Osher, S., "Stable and Entropy-Satisfying Approximations for Transonic Flow Calculations," *Math. Comput.*, Vol. 34, pp. 45–75, 1980.

[13] Bell, J. B., Colella, P., Trangenstein, J. A., "Higher-Order Godunov Methods for General Systems of Hyperbolic Conservation Laws," *J. Comput. Phys.*, Vol. 82, No. 2, pp. 362–397, June 1989.

[14] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA-81-1259, June 1981.

[15] Bell, J. B., Colella, P., Trangenstein, J., and Welcome, M., "Adaptive Mesh Refinement on Moving Quadrilateral Grids," AIAA–89–1979–CP, June 1989.

[16] Dudek, S.A., "A Structured-Grid Adaptive Mesh Refinement Multigrid Algorithm for Steady-State Flows," PhD Thesis, The University of California, Berkeley, Department of Mechanical Engineering, December, 1996.

[17] Ferziger, J. H., "Estimation and Reduction of Numerical Error," in FED-Vol. 158, *Quantification of Uncertainty in Computational Fluid Dynamics*, pp. 1–7, ASME, 1993.

[18] Berger, M., and Rigoutsos, I., "An Algorithm for Point Clustering and Grid Generation," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 5, pp. 1278–1286, Sept/Oct 1991.

[19] Almgren, A.S., Buttke, T., and Colella, P., "A Fast Adaptive Vortex Method in Three Dimensions," *J. Comput. Phys.*, Vol 113, pp. 177–200, 1994.

[20] Wesseling, Pieter, An Introduction to Multigrid Methods, John Wiley & Sons, 1991.

[21] Hirsch, C., Numerical Computation of Internal and External Flows, Volume 2, pp. 385–387, John Wiley & Sons, 1990.

[22] Jameson, A., and Yoon, S., "LU Implicit Schemes with Multiple Grids for the Euler Equations," AIAA–86–0105, January 1986.

[23] Viviand, H., "Numerical Solutions of Two-Dimensional Reference Test Cases," in *Test Cases for Inviscid Flow Field Methods*, AGARD–AR–211, May 1985.

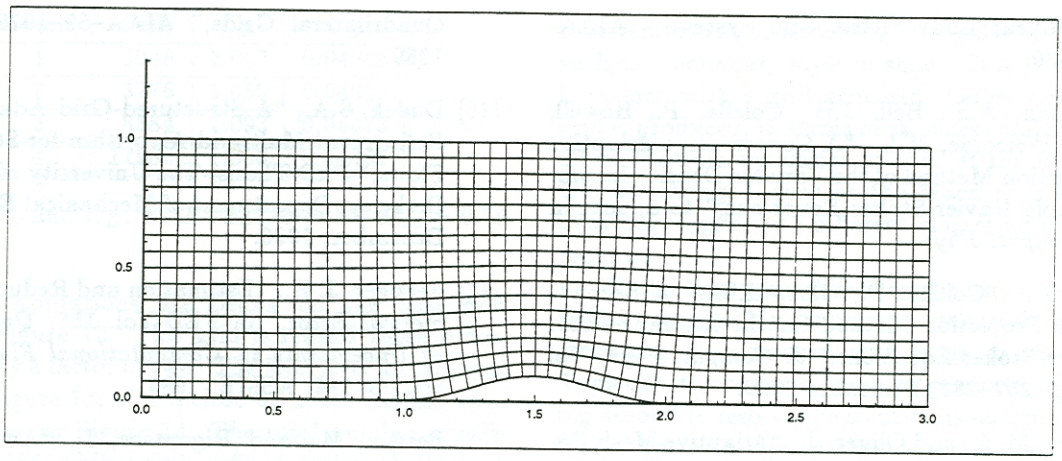


Figure 2: 10% Sine-Squared Bump Base Grid, 48 x 16 Cells

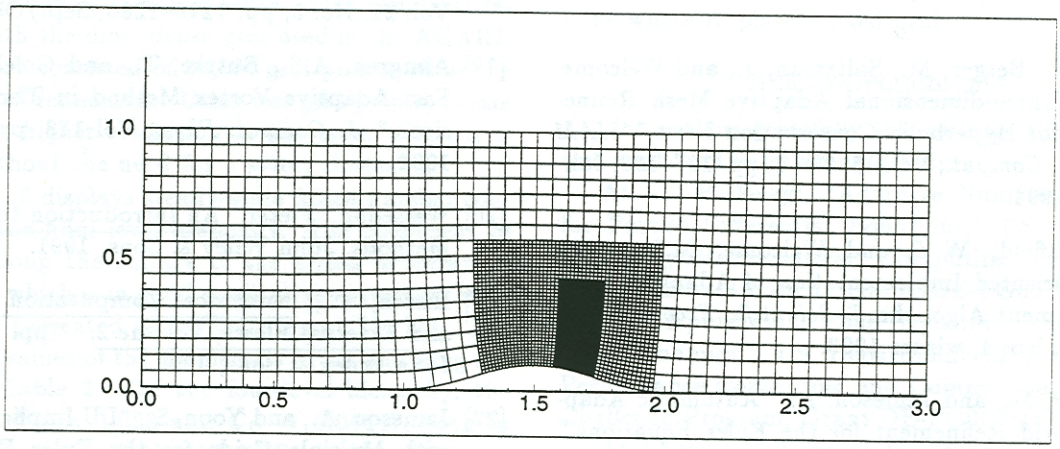


Figure 3: Three-Level Hierarchy, 10% Sine-Squared Bump, $M_{\text{inflow}} = 0.675$

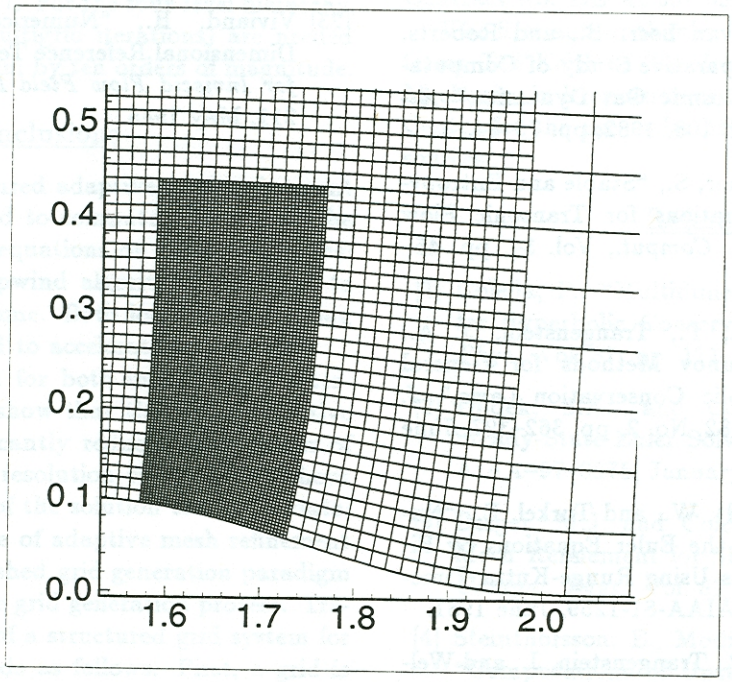


Figure 4: Close-Up of Three-Level Hierarchy, 10% Sine-Squared Bump, $M_{\text{inflow}} = 0.675$

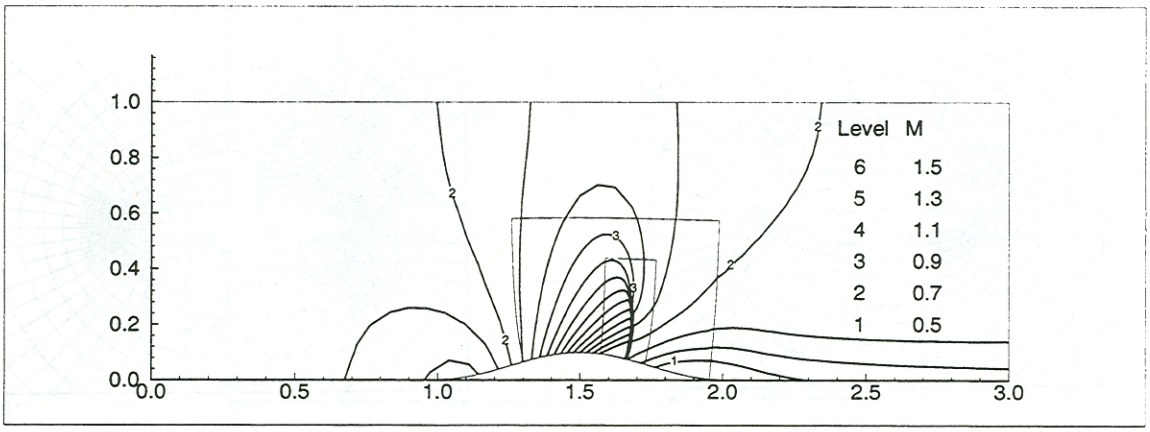


Figure 5: Mach Number (15 Contours, Min = 0.4531, Max = 1.5854), 10% Sine-Squared Bump, $M_{\text{inflow}} = 0.675$; Numbers on Contour Lines Correspond to Table Values.

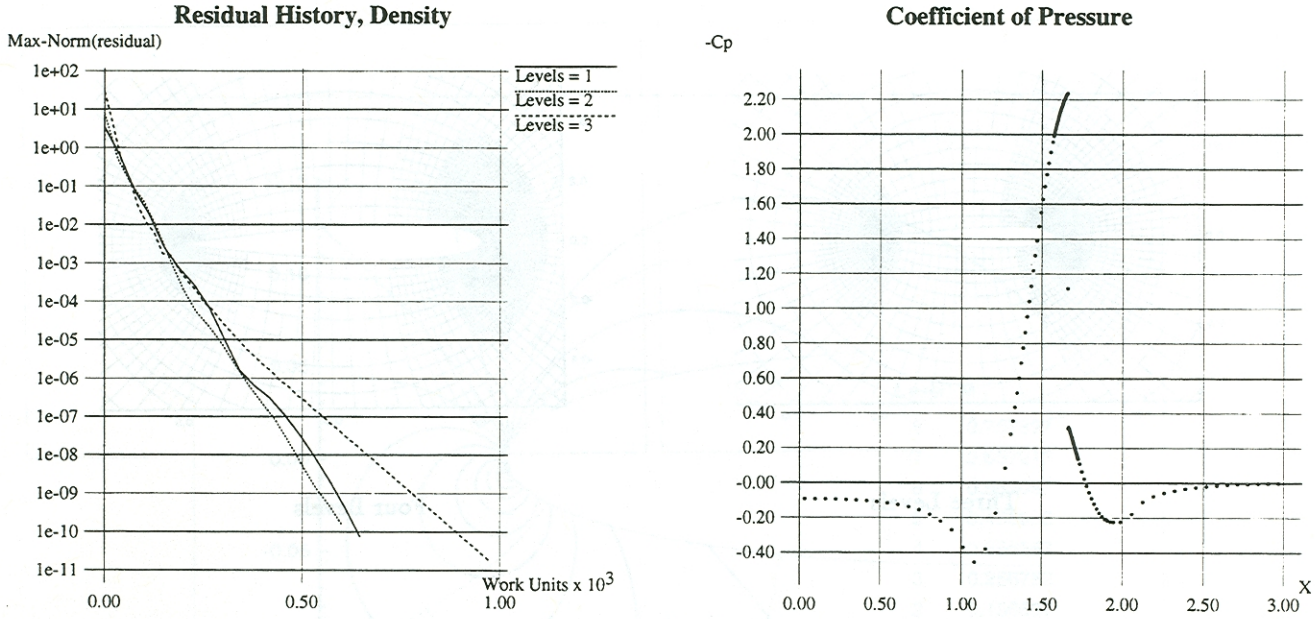
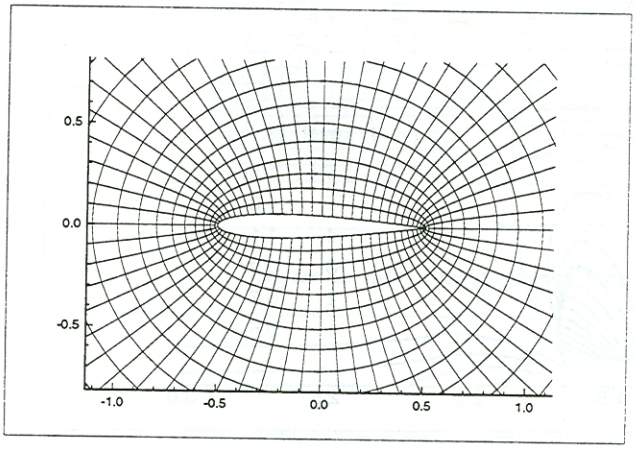
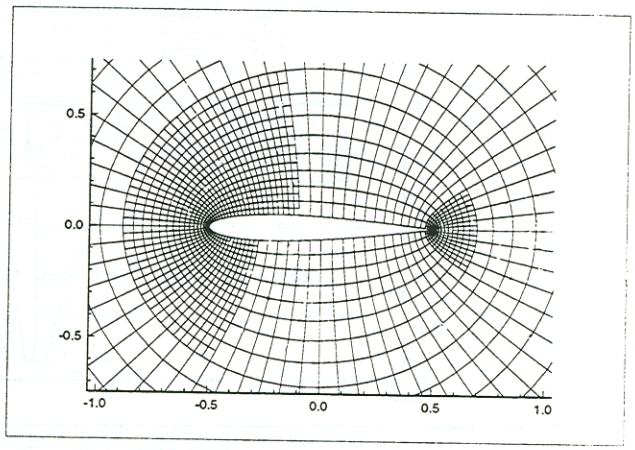


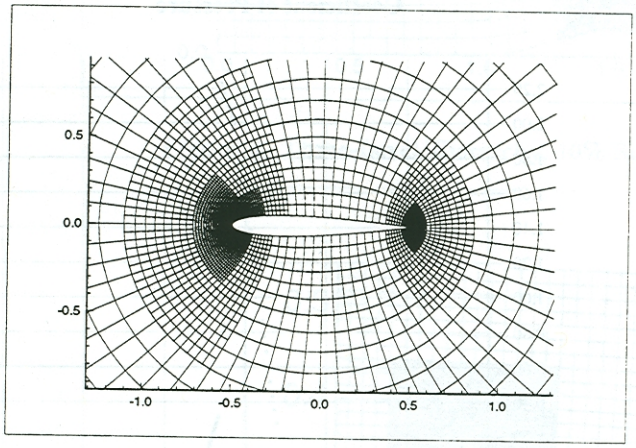
Figure 6: 10% Sine-Squared Bump, $M_{\text{inflow}} = 0.675$



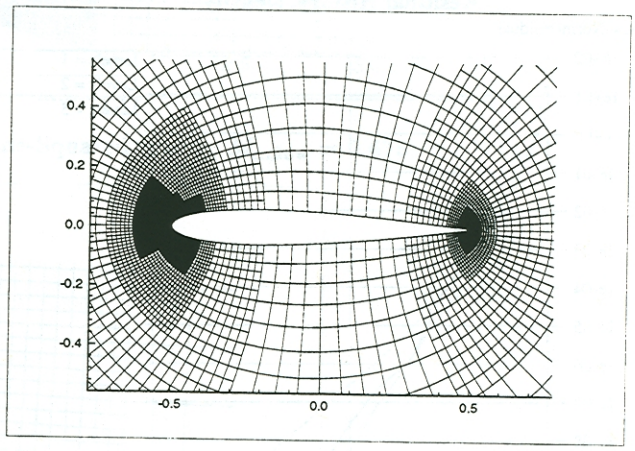
One Level



Two Levels



Three Levels



Four Levels

Figure 7: Evolution of Four-Level Grid Hierarchy, NACA 0012, $\alpha = 3°$, $M_\infty = 0.5$

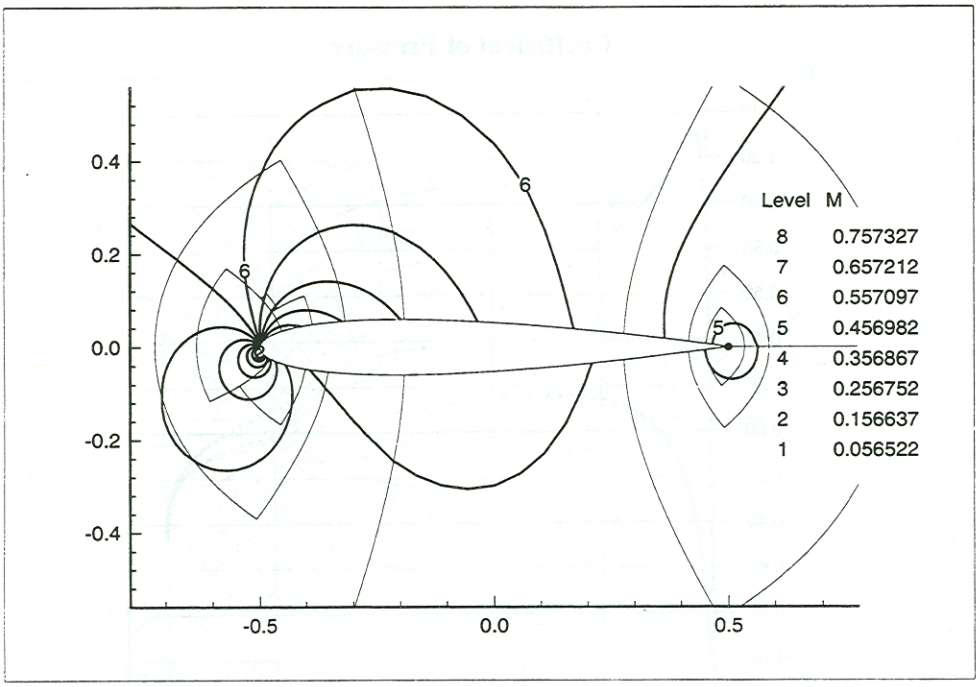


Figure 8: Mach Number (15 Contours, Min = 0.05652, Max = 0.7573), NACA 0012, $\alpha = 3°$, $M_\infty = 0.5$; Numbers on Contour Lines Correspond to Table Values.

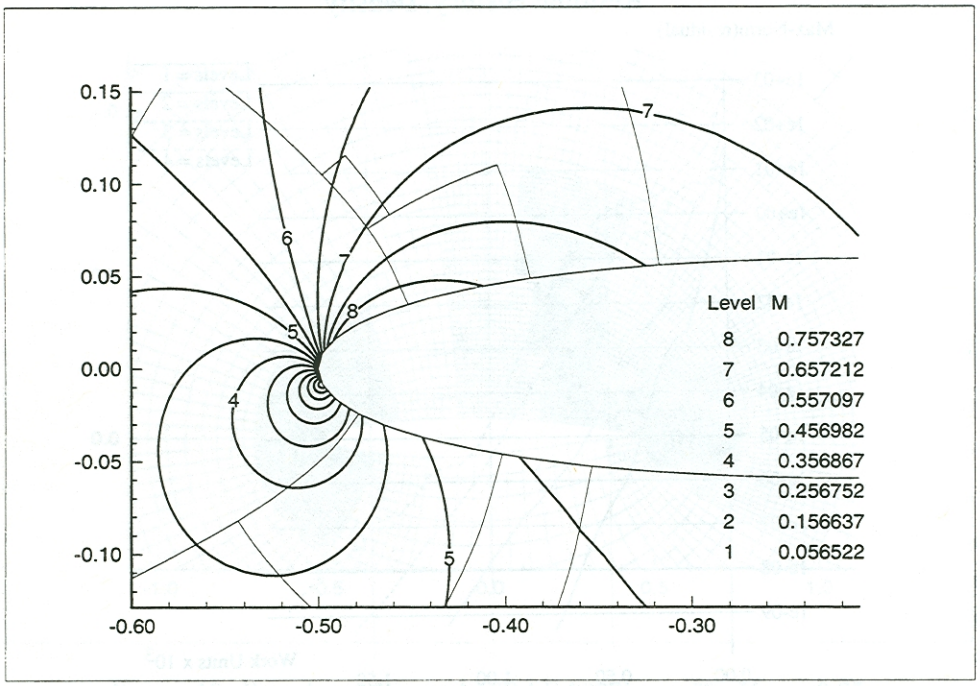


Figure 9: Mach Number (15 Contours, Min = 0.05652, Max = 0.7573), NACA 0012 Leading Edge, $\alpha = 3°$, $M_\infty = 0.5$; Numbers on Contour Lines Correspond to Table Values.

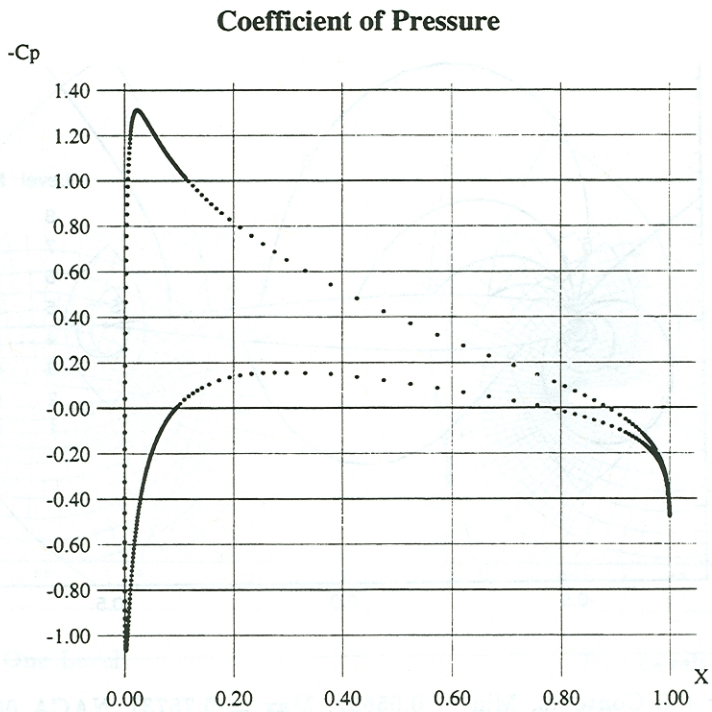


Figure 10: NACA 0012, $\alpha = 3°$, $M_\infty = 0.5$

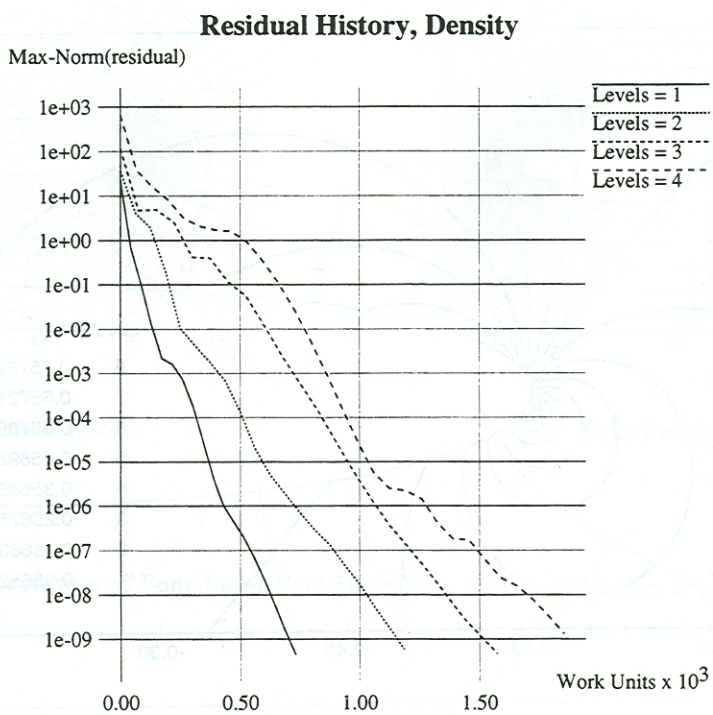


Figure 11: NACA 0012, $\alpha = 3°$, $M_\infty = 0.5$

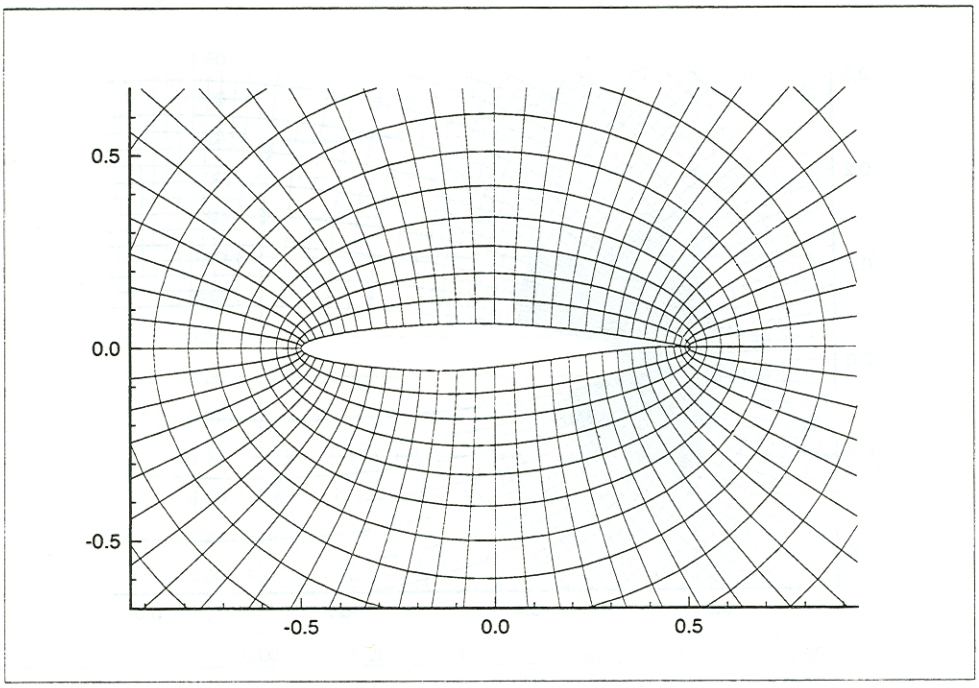


Figure 12: RAE 2822 Base Grid, 32 x 64 Cells

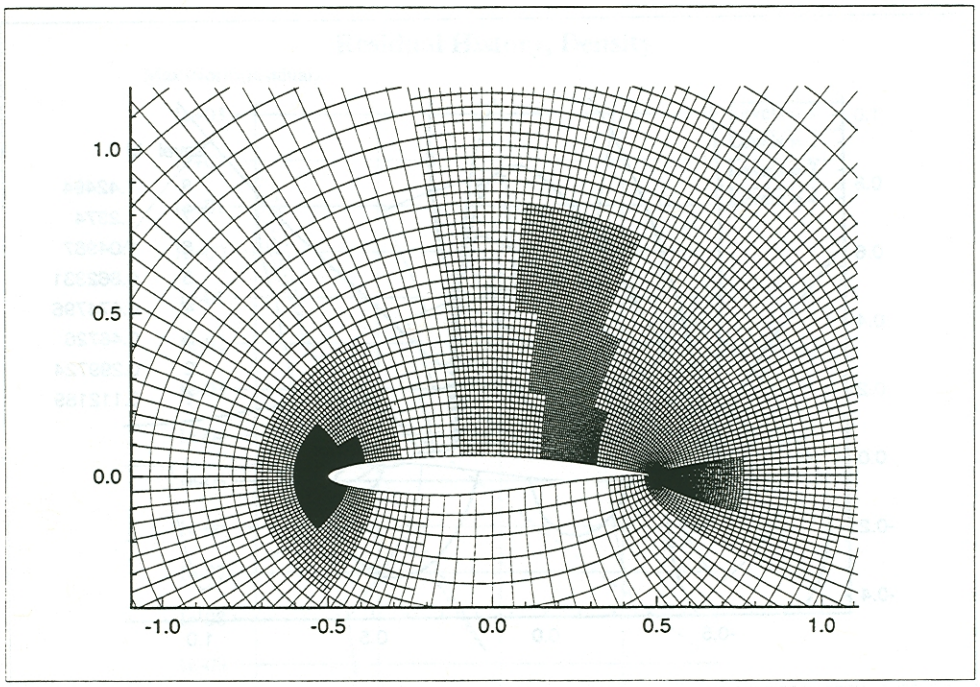


Figure 13: Four-Level Hierarchy, RAE 2822, $\alpha = 3°$, $M_\infty = 0.75$

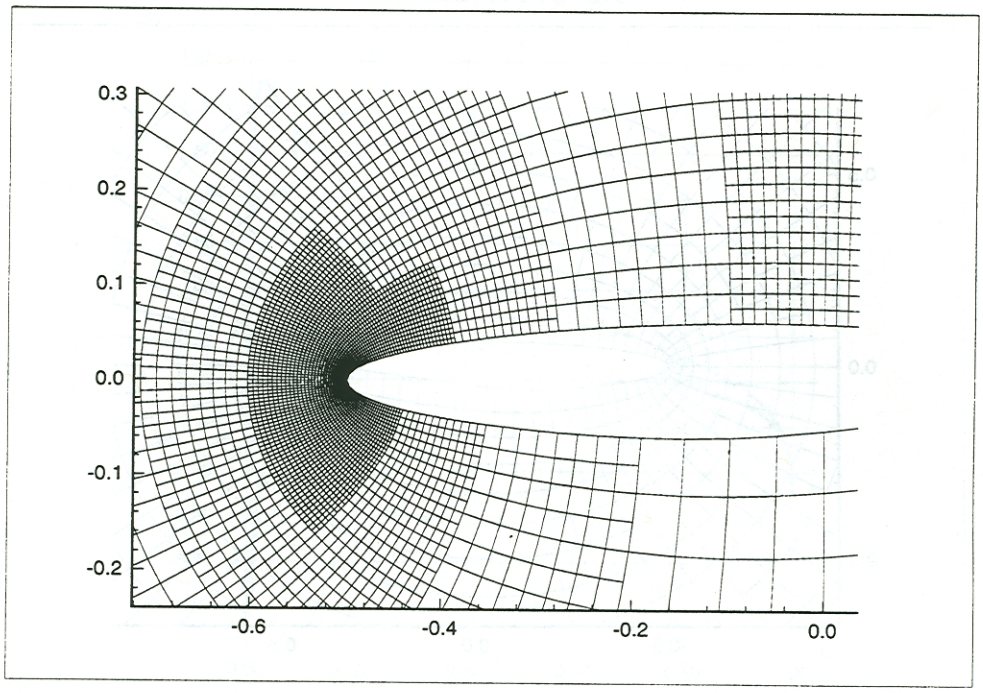


Figure 14: Four-Level Hierarchy, RAE 2822 Leading Edge, $\alpha = 3°$, $M_\infty = 0.75$

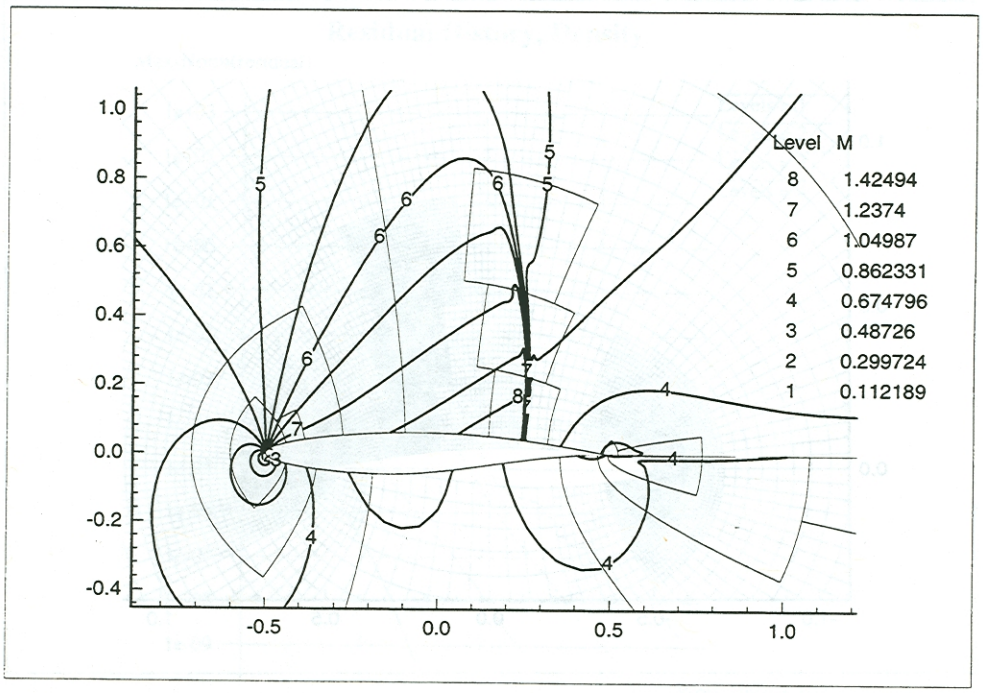


Figure 15: Mach Number (15 Contours, Min = 0.1122, Max = 1.4249), RAE 2822, $\alpha = 3°$, $M_\infty = 0.75$

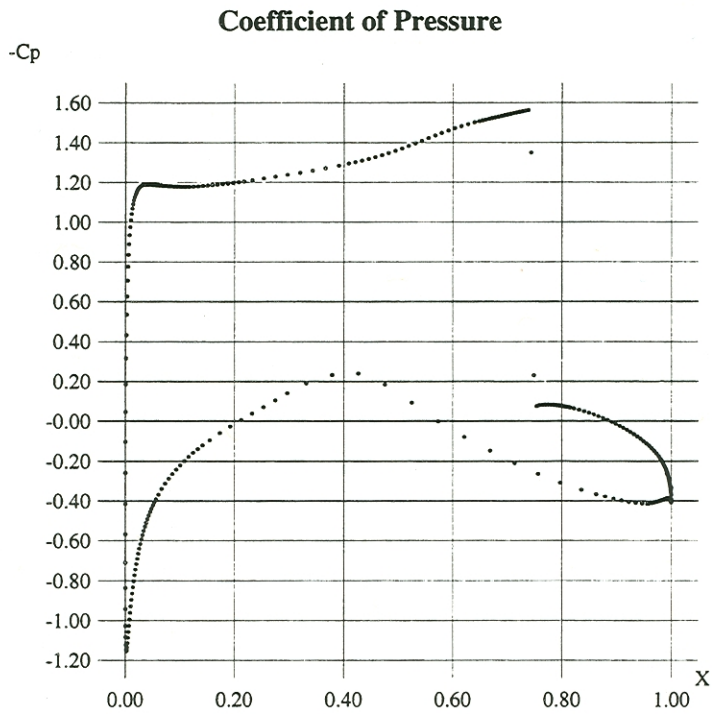


Figure 16: RAE 2822, $\alpha = 3°$, $M_\infty = 0.75$

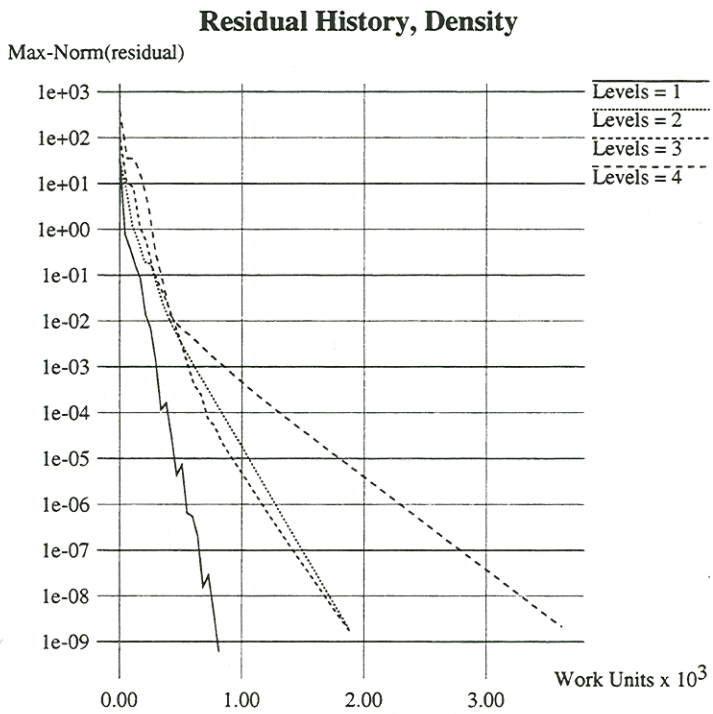


Figure 17: RAE 2822, $\alpha = 3°$, $M_\infty = 0.75$