AN ADAPTIVE MULTIGRID METHOD FOR THE EULER EQUATIONS

Marsha J. Berger

Courant Institute of Mathematical Sciences
251 Mercer Street
New York University
New York, NY 10012

Antony Jameson

Princeton University
Dept. of Mechanical and Aerospace Engineering
Princeton, NJ 08544

1. INTRODUCTION

It is well known that an accurate solution to the Euler equations requires more resolution in some parts of the flow field than others. Just where these regions are, and how fine the mesh spacing must be, depends on characteristics of the solution to be computed, and changes with different flow parameters, such as Mach number, angle of attack, etc. Therefore, it cannot be known in advance of the computation. A solution technique which gets comparable accuracy over the entire flow field, thus using fewer points in smoother regions of the solution and more elsewhere, would clearly be optimal.

In this talk, we describe a method of local adaptive grid refinement for the solution of the steady Euler equations in two dimensions, which automatically selects regions requiring mesh refinement by measuring the local truncation error. Our method of refinement uses locally uniform fine rectangles which are superimposed on a global coarse grid. Possibly several nested levels of refined grids will be used until a given accuracy is attained. The fine grid patches are in the same coordinate system as the underlying coarse grid. All the data management is done in the computational plane, where, since we use rectangular grids, the data structures and bookkeeping can be very simple (see figure 1). Furthermore, the same data structures and control flow needed for adaptive grid refinement are also required for multigrid convergence acceleration. This can therefore be added with little additional cost.

Other adaptive mesh strategies have previously been proposed. Many of them are moving grid point methods, where a logically rectangular mesh is distorted to put more grid points in region where the solution error is large. Brackbill and Saltzman [3] do this by having their mesh minimize a functional which includes terms measuring the solution error, grid smoothness, and grid orthogonality. Rai and Anderson [4] attract the grid points into regions with high error by determining grid point speeds. These methods seem to work well in the examples in the literature, and they do not suffer from the difficulties with conservation that a patched grid method has at the grid interfaces. However, controlling the grid skewness in these methods is very difficult, and will be more so in three dimensions. Our method does not have this drawback. Furthermore, in a moving grid point method, it is still desirable to have a mechanism to add new points if necessary. Recent work by Murman and Usab [5] presents a similar approach to grid adaption using nested grid patches, but does not include a procedure for automatic control of the error. We add here that our adaptive method is a general purpose approach (and program) which has successfully been applied to computing other problems besides transonic flow (see e.g. [6]).

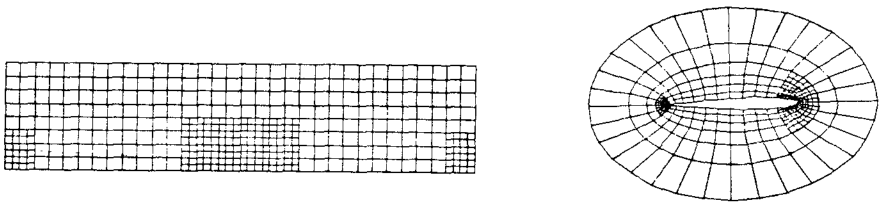

Figure 1. Fine grids at the leading and trailing edges.

The adaptive algorithm consists of four main steps. First, an automatic error estimator decides where the coarse grid accuracy is insufficient. This is based on estimates of the local truncation error, obtained by a method similar to Richardson extrapolation. In the second step, the local fine grid patches are created in the regions of high error. (This typically occurs at the leading and trailing edges of the airfoil and at a shock). Third, the solution on this multiple grid structure is integrated to steady state using the same integrator in the interior of each grid. Because of the regular grid structure, we have been able to use an existing method (FLO52) on each grid. However, special boundary conditions are needed at the interface of the fine and coarse grids to insure that the overall scheme is conservative and stable. In the separately described fourth step, the convergence of the solution is accelerated using a multigrid method with the modifications needed for this grid structure. We briefly describe each step, and show computational examples. A more detailed description of the adaptive algorithm can be found in [1], and of the underlying integrator and multigrid accelerator in [2].

2. DESCRIPTION OF THE ADAPTIVE ALGORITHM

The solution procedure begins by iterating on a single global grid, or possibly several nested grid levels, if multigrid is already being used. We wait until the residual of the solution is approximately 10^{-2} before applying the error estimator and subsequent adaptive strategy, so that the error due to lack of convergence is less than the error due to discretization. The local truncation error in the solution is estimated using an automatic procedure similar to Richardson extrapolation. In this approach, the same fine grid integrator is used for one step on a coarsened grid of mesh spacing twice the fine grid spacing. This coarsened residual is proportional to the local truncation error of the solution. Those grid points where the error estimate exceeds the specified tolerance are flagged as needing to be in a grid with finer mesh spacing. The grid generation algorithm creates rectangular grids, so that every flagged point is contained in a fine grid. We emphasize that these fine grids are not patched into one global grid, but are kept independently, each with its own solution vector. It is important that these grids are rectangles in the computational plane. This means that regardless of what method is chosen, the same integrator can be used to advance the solution of each grid, and it will vectorize on each grid. This also keeps the data structures fairly simple. The overhead in this approach is that the solution will be computed and stored on the entire coarse grid, even in those regions where it is not needed and the fine grid solution is used instead. However, we feel this is simpler than fragmenting the coarse grid and using pointers to indicate where each row and column begin.

Given the newly created fine grid structure, the solution on each grid is initialized by interpolation from the coarser grids, and the time stepping continues. For each step on the coarse grid, one (or more) steps on each fine grid are taken. When all grids have been advanced, the fine grid solution is used to update all coarse grid points which are lying underneath a fine grid. This is done by replacing the coarse grid solution value using a volume weighted interpolation from the fine grid to maintain conservation. The final step in the solution process is the specification of the boundary values for the fine grids by interpolation from coarser grids. This must be done in a way such that overall conservation in the solution is maintained at the interface between the fine and coarse grids. This is the only part of the adaptive strategy that is not independent of the chosen method of integration. Since the scheme we use to integrate the solution (FLO52) is based on a finite volume scheme, it is possible to devise a stable and conservative scheme such that the flux into the fine grid balances the flux into the coarse grid. This is described in more detail in [1].

These calculations are supported by a general purpose computer program that handles the storage and data flow management problems for multiple grids, (see [7] for details). This makes possible multiple levels of refinement with the fine grids themselves further refined in regions requiring very high resolution. These same routines support the multigrid algorithm described below. A similar approach can also be taken for calculations involving independently generated component grids.

3. DESCRIPTION OF THE MULTIGRID ALGORITHM

The method chosen to integrate the Euler equations uses a finite volume spatial discretization, augmented by dissipation terms, and a multi-stage time stepping procedure to integrate the resulting ODE's. If we write the equations as

$$\frac{d}{dt}(hw) + Qw - Dw = 0,$$

where w is the solution vector $(\rho,\rho u,\rho v,e)^T$, h the cell area, Q the terms from the Euler equations, and D the dissipative terms, the time stepping sequence looks like

$$w^{(1)} = w^{(0)} - \Delta t\alpha^{(0)}\left(\frac{Qw^{(0)}-Dw^{(0)}}{h}\right)$$

$$w^{(2)} = w^{(0)} - \Delta t\alpha^{(1)}\left(\frac{Qw^{(1)}-Dw^{(0)}}{h}\right)$$

etc. The $\alpha^{(j)}$ are chosen using criteria such as maximizing the stability region, or in the case of multigrid, maximizing the damping of the higher frequencies. Notice that the part of the right hand side in parentheses is just the residual, henceforth denoted $R_h(w)$ when on a grid with mesh spacing h.

A full multigrid acceleration strategy has been devised for use with FL052 on global grids (see [2]). In this scheme, a forcing function is defined so that on a coarser grid, with the solution obtained by conservative interpolation from the fine grid denoted by w_{2h}, the equations being solved are

$$w^{(1)}_{2h} = w^{(0)}_{2h} - \Delta t\alpha^{(0)} (Rw^{(0)}_{2h} + P_{2h})$$

$$w^{(2)}_{2h} = w^{(0)}_{2h} - \Delta t\alpha^{(1)} (Rw^{(1)}_{2h} + P_{2h})$$

where $P_{2h} = \sum R(w_h) - R(w^{(0)}{}_{2h})$. The sum is taken over the four fine cells comprising one coarse cell. This formulation insures that when the residual is zero on the fine grid, the solution on the coarse grid does not change. After one iteration on a coarse grid, the correction is interpolated back to the fine grid using bilinear interpolation.

To use an adaptive multigrid strategy, we start by iterating on a coarse grid. The Richardson error estimates are used to determine how fine a grid (global or local) is necessary to obtain a certain accuracy in the solution. Refinement based on error estimates has also been proposed in the multigrid literature. To use multigrid on fine grids that are not global means that when iterating on the global coarse grid, some cells have a forcing function from a finer grid and some do not. This is easily arranged, since the adaptive strategy outlined above already accounts for the fact that some coarse solution values are updated from a finer grid, and some are not. This updating corresponds to the restriction operation in the multigrid literature. The only new part of the algorithm that needs to be added to the adaptive code is a prolongation routine to add the correction back to the fine grid. In using multigrid on patched grids one must be careful to make sure that the solution procedures for boundary values and interfaces are compatible, so that a steady state solution is possible. For example, the far field boundary conditions for a fine grid are extrapolated for outflow boundaries using the entropy in the far field. Coarser grids however should not use this far field boundary specification, since coarse grid solution values obtained by interpolation from the fine grid give a different entropy value. Thus, using this to obtain the far field solution on the coarse grid would change a converged solution on a fine grid. We add here that although our preliminary results indicate that multigrid is effective in accelerating the solution of the Euler equations on a patched grid system, some of the assumptions of the standard multigrid optimality theory [8] are no longer valid, and it remains an open question whether a solution can be obtained in a fixed number of iterations independent of the number of grid points.

4. NUMERICAL RESULTS

Figure 2 shows a typical calculation for transonic flow over a NACA0012 airfoil with a Mach number of 0.8. Using an error tolerance of .004, the adaptive algorithm automatically generates 5 refined grids. One grid patch is at the leading edge, two are at the trailing edge (since the grid is periodic with a cut at the trailing edge), and two patches are at the top and bottom of the airfoil, surrounding the shock. Figure 3 shows the pressure coefficient computed on this grid. A total of 300 iterations were done in this calculation but only 200 of them included the five finest grids. This is approximately a factor of 6 faster than without using multigrid. The combined area of these grids was only 30% of a uniformly refined grid with 128 by 32 grid points, and so the cost of an iteration was only 1/3 the cost of an iteration of a uniformly refined grid. Faster rates of convergence have been obtained by multigrid applied to global grids without embedded patches [2], but the patched method yields savings in computational effort owing to the reduced cost per iteration.

5. REFERENCES

[1] M. Berger and A. Jameson, "Automatic Adaptive Grid Refinement for the Euler Equations", NYU Report No. DOE/ER/03077-202, October, 1983. To appear in AIAA Journal.

[2] A. Jameson, "Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method", MAE Report NO. 1613, June 1983, Applied Math. and Computation, 13 (1983), 327-356.

[3] J. Saltzman and J. Brackbill, "Applications and Generalizations of Variational Methods for Generating Adaptive Meshes", in Numerical Grid Generation, J. Thompson, ed., North-Holland, 1982.

[4] M. Rai and D. Anderson, "Grid Evolution in Time Asymptotic Problems", J. Comp. Phys. 43 (1981), 327-344.

[5] W. Usab, Jr. and E. Murman, "Embedded Mesh Solutions of the Euler Equation Using a Multiple-grid Method", AIAA Paper 83-1946-CP. Presented at the 6th AIAA Computational Fluid Dynamics Conference, Danvers, Mass. July 1983.

[6] P. Collela and H. Glaz, "Numerical Computation of Complex Shock Reflections in Gases". These proceedings.

[7] M. Berger and J. Oliger, "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations", J. Comp. Phys. 53 (1984), 484-512.

[8] W. Hackbusch, "On The Multi-Grid Method Applied to Difference Equations," Computing, 20 (1978), 291-306.

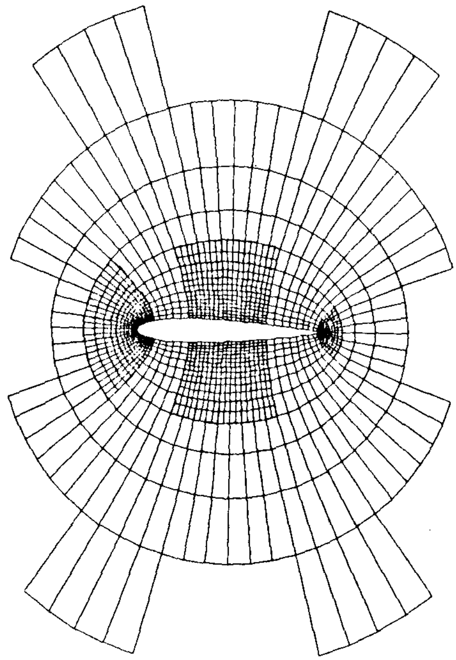

Figure 2(a)
Grid 64 x 16

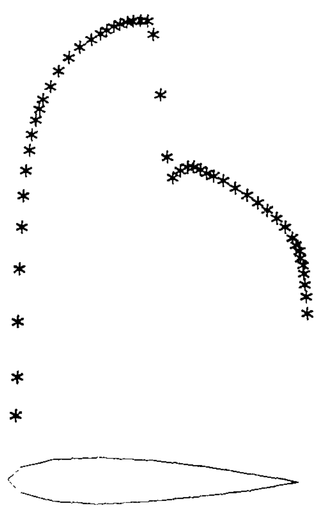

Figure 2(b)
MACH 0.800 ALPHA 0.0
CL 0.0000 CD 0.0086 CM 0.0000