1995122344

N95-28765

MULTILEVEL ELLIPTIC SMOOTHING OF LARGE THREE-DIMENSIONAL GRIDS

C. Wayne Mastin*
Mail Stop 125
NASA Langley Research Center
Hampton, VA 23681

SUMMARY

Elliptic grid generation methods have been used for many years to smooth and improve grids generated by algebraic interpolation schemes. However, the elliptic system that must be solved is nonlinear and convergence is generally very slow for large grids. In an attempt to make elliptic methods practical for large three-dimensional grids, a two-stage implementation is developed where the overall grid point locations are set using a coarse grid generated by the elliptic system. The coarse grid is then interpolated to generate a finer grid which is smoothed using only a few iterations of the elliptic system.

INTRODUCTION

Elliptic grid generation methods have become less applicable to large scale problems due to the time required to solve the elliptic system of partial differential equations. The equations themselves are nonlinear and are difficult to solve efficiently even using the traditional multigrid methods. Although there are some variations of the equations, this report assumes the elliptic system is of the form

$$g^{11}(r_{\xi\xi} + P\ r_{\xi}) + g^{22}(r_{\eta\eta} + Q\ r_{\eta}) + g^{33}(r_{\zeta\zeta} + R\ r_{\zeta}) + g^{12}r_{\xi\eta} + g^{13}r_{\xi\zeta} + g^{23}r_{\eta\zeta} = 0$$

where $r = (x, y, z)$, g^{ij} are the contravariant metric tensor components and the functions P, Q, and R are used to control the distribution of grid points. The objective of this report is to demonstrate that in many cases it is not necessary to solve the elliptic system to generate a smooth grid with the required grid point distributions. If a coarse grid is first generated by solving the elliptic system, then this grid can be interpolated to generate a finer grid and the fine grid can be smoothed with only a few iterations of the elliptic difference equations. If this procedure is to work in practice, it is essential that the interpolated grid be smooth and give a good approximation of the final solution of the elliptic system on the fine grid. Thus, the fine grid iterations are primarily used to eliminate interpolation errors which are local and of high frequency. The actual residual on the fine grid may not be close to zero. This multilevel approach is efficient if only a few fine grid iterations are to give a smooth grid. It is well known that if the initial grid deviates greatly from the final elliptic grid, the first few iterations may generate large scale oscillations which decay very slowly. The multilevel approach can also be used to generate grids with specified boundary orthogonality and spacing. Coarse grid computations can be used to generate good initial approximations of control functions which may be fixed or further adjusted during the fine grid iterations.

*This work was performed while the author held a National Research Council - NASA Langley Research Center Research Associateship.

The techniques described in this report can be easily implemented in existing software packages. Coarse grids solutions have been used to generate starting values for the elliptic system in the GRIDGEN code of Steinbrenner and Chawner (ref. 1) and the 3DGRAPE code of Sorenson (ref. 2). Eisman (ref. 3) has also employed a similar concept in using elliptic systems to distribute control points for his algebraic grid generation scheme in the the GRIDPRO code.

It should be noted that the proposed scheme is not a true multigrid scheme. The scheme here is a simple grid sequencing scheme progressing from a coarse grid to progressively finer grids. There is no cycling between coarse and fine grids with the objective of eventually achieving a higher rate of convergence to the solution of the elliptic system of partial differential equations on the finest grid. Multigrid methods have been shown to increase convergence rates, but there have been no applications to large systems with moving boundary points and adaptive control functions. Some difficulties that may arise are discussed in the paper by Stüben and Linden (ref. 4).

COARSE GRID ITERATIONS

The first step in the procedure is to coarsen the initial algebraic grid. This scheme that has been used here is to remove every other grid point in each coordinate direction. This refinement may proceed to several levels as long as the grid dimensions in each direction are odd integers. For large grids, this reduces the size of the problem by approximately eight. A second coarsening would reduce the original problem by a factor of approximately 64. Complementing the reduction in problem size is an increase in the rate of convergence when going to a coarser grid. The coarse grid iterations should be optimized for rapid convergence. For example, all coarse grid calculations here have used locally optimal acceleration parameters. Any control function and boundary condition options should be implemented so that the converged coarse grid has all of the desired characteristics of the final grid.

INTERPOLATION

The interpolation scheme is a critical component in this procedure. Therefore, a tricubic Hermite interpolation procedure has been developed to generate a smooth grid. The slope information is calculated using central differences on the coarse grid. There are two options for the bounding surface grids. Either the original surface grids can be used or the surface grids can be redefined using the same interpolation scheme used at the interior points. The choice can have a significant effect on the success of the coarse grid solution in reducing the amount of work needed to obtain a usable grid. If the coarse grid accurately resolves the surface so that there is little change is curvature or spacings between grid points, then one can generally use the original fine surface grids and still have a reasonably smooth grid to start the elliptic system on the fine grid. On the other hand, if there are significant changes in surface or grid properties between the interpolated grid and the original surface grids, then using the original grids on the boundary surfaces will result in large changes in grid spacings and angles at the boundary, and possibly even some places where the interpolated grid folds over the boundary. This would result in a poor starting grid for the elliptic system and the main purpose in using the coarse grid solution would be lost.

The ability to prescribe grid distributions on the final grid is very dependent on the control functions of the elliptic system. There are two options for calculating control functions which seem to work equally well. The control functions can either be calculated based on the initial fine grid and then restricted to each coarser grid, or the control functions can be calculated on each grid from restrictions of the initial grid to that level. It is generally recommended that the control functions not be computed on the coarsest grid and interpolated to the finer grids. This will often result in a loss of distribution on the finer grid. The only time that control functions have been interpolated to finer girds is when using the control functions to control boundary spacing an orthogonality. Even then, the interpolated control functions are blended with the control functions from the initial grid so that the interpolated values are only effective near the boundary.

There is one important fact that should be emphasized when transferring control functions between fine and coarse grids. When the control functions are restricted to a coarser grid, then they should also be multiplied by a factor of two. This scaling factor is necessary so that the same elliptic system is approximated on both the coarse and fine grids. Conversely, if the control functions are transferred from a coarse to a finer grid, then the control functions should be interpolated and divided by two. This assumes that the grid coarsening is done by removing every other point in each coordinated direction. Other coarsening schemes would result in different scaling factors. For example, if only every third point was retained in generating the coarse grid, then the factor of two would be replaced by three.

FINE GRID ITERATIONS

At this point it is assume that the coarse grid iterations and interpolation procedure has resulted in a smooth grid that has the desired distribution of grid points. In many cases this grid would be good enough to compute a CFD solution. However, there may be a few ripples in the grid due to the Hermite interpolation. If the original boundary surfaces are maintained, the grid may need some additional smoothing near the boundary. Since the objective here is only to smooth the grid and not to obtain convergence of the elliptic system, there should be a change in the relaxation parameter so that the iterations are underrelaxed. Another effective way of smoothing the grid while maintaining the existing distribution of grid points is to introduce a time derivative into the partial differential equations, and solve the resulting parabolic system using a time marching method.

EXAMPLES

Three sample grids will be considered to test the concepts of this report. All three initial grids were generated using transfinite interpolation. All three also are obviously not suited for CFD computations because of negative Jacobians or extremely skewed cells. After applying the elliptic smoothing, in two of the three cases the final grid was free of negative Jacobians. In the other case, a few negative Jacobians remained after using the elliptic methods, even when convergence of the elliptic system was attempted on the original fine grid.

The first example is the grid in the interior of a duct. The duct is plotted in Figure 1(a). The cross section is very irregular. An initial grid was constructed with dimensions of 33 by 33 by 65. There are a large number of grid points which fall outside of the duct as can be seem in the plot of an interior grid surface in Figure 1(b). This grid can be improved using elliptic methods, so that no negative Jacobians appear. In fact, negative Jacobians can be eliminated without resorting to grid coarsening in only seven iterations. However, it takes many more iterations to give a smooth grid. The negative Jacobians in the initial grid cause large values of the residual in the solution of the elliptic system. Nonelliptic phenomena like ripples and waves can develop in the numerical solution. These perturbations may be of large magnitude and decay very slowly during the iterative solution. For example, the plot in Figure 1(c) is the same interior grid surface after ten iterations of the elliptic system. Ripples in the grid appear as triangular shaped cells in the lower part of the surface grid. Now consider the case where the original grid is coarsened to give an 17 by 17 by 33 grid. The elliptic system can be converged on this grid in less than sixty iterations, which is equivalent to less that eight iterations on the final grid. This coarse grid is interpolated and further smoothed to eliminate any rough spots near the boundary. For comparison with Figure 1(c), ten iterations on the fine grid was performed after convergence on the coarse grid, and the resulting grid plotted in Figure 1(d). As a further comparison, the original grid was used to calculate a converged solution of the elliptic system and it took about twice as many iterations to converge on the fine grid as it did on the coarse grid. Thus, when considering both the size of the grid and the convergence rate, it can be concluded that the use of coarse grid iterations resulted in a reduction in work by a factor of five over what would be needed to generate a converged elliptically generated grid. Zero control functions were used in all of these calculations. The grid points were fixed on the walls of the duct, but were allowed to slide along the two circular end caps.

The second example is a grid for a region about an aircraft wing. The initial grid is a C-grid constructed using transfinite interpolation. The grid was constructed as a four block grid each of which was 33 by 53 by 33 for a total of slightly over 230 thousand grid points. The edges of the blocks are illustrated in Figure 2(a) There are no negative Jacobians in this grid, but the grid is highly skewed in regions near the wing tip as can be seen in Figure 2(b). Several options have been exercised in this example which tend to reduce the rate of convergence for the elliptic system. The control functions have been interpolated from the block boundaries (using the Thomas - Middlecoff technique) to maintain the interior grid point distribution during the solution of the elliptic system. The control functions have been allowed to adjust near the surface of the wing (as in the GRAPE code) to generate orthogonal grid lines at the wing surface. Finally, the grip points have been allowed to float along the two planar boundary surfaces intersecting the ends of the wing. An indication of the slow rate of convergence is evident in Figure 2(c). After 10 iteration, starting with the original algebraic grid, there is little difference between the grid generated using the elliptic difference equations and the original algebraic grid. There is some indication of boundary orthogonality. However, there is considerable difference when examining Figure 2(d) which was generating using 100 coarse grid iterations followed by 10 fine grid iterations. The control functions were treated as described above. The control functions were first interpolated from the coarse grid to the fine grid and divided by two, since the coarse grid was generated by taking every other point of the fine grid. These control functions were then blended with the control functions computed from the initial interpolated grid. For this example, the original surfaces were used with the fine grid. It was therefore necessary to continue to adjust the control functions during the fine grid iterations to correct the slight skewness at the boundary which resulted from the interpolation.

The final example is included to demonstrate that this method can be applied to a grid with highly nonuniform spacing. The initial grid was again constructed using transfinite interpolation. The grid was to be used to compute viscous flow about an HSCT configuration. The wing/fuselage configuration is plotted in Figure 3(a). Since the HSCT configuration was symmetric, only half of the body was used to generate the grid. The grid was constructed in two blocks, each of which was 177 by 81 by 61. Thus the total grid consists of nearly one and three-quarter million points.

There are a large number of negative Jacobians. A comparison of the initial grid and the elliptic grid after 100 coarse grid iterations appears in Figures 3(b) and 3(c). For clarity, only the coarse grid points are plotted. The elliptic method was able remove most of the negative Jacobians and still maintain the spacing at the boundary. However, it could not remove all negative Jacobians and generate a grid which would be suitable for CFD calculations. While this may be considered a failure of elliptic methods, it is a successful application of the multilevel method. Knowing when and where elliptic methods fail to generate a suitable grid will allow the user to proceed with his efforts in redefining the topology or redistributing points.

CONCLUSIONS

The discussion and examples contained in this report should give the grid generator a guide for using coarse grid iterations for smoothing and improving computational grids for CFD applications. The main point is that the coarse grid solution should be converged to some specified tolerance. After that, only a few fine grid iterations are needed. It is also important to treat the control functions correctly to generate the desired distribution of grid points along boundary surfaces. Simply interpolating these functions from the coarse grid to the fine grid is generally not sufficient.

There is one area where further study is needed. There should be some way of projecting an interpolated volume grid onto the original boundary surfaces without effecting the overall smoothness and orthogonality of the grid. One possible approach would be to include boundary information into the interpolation scheme.

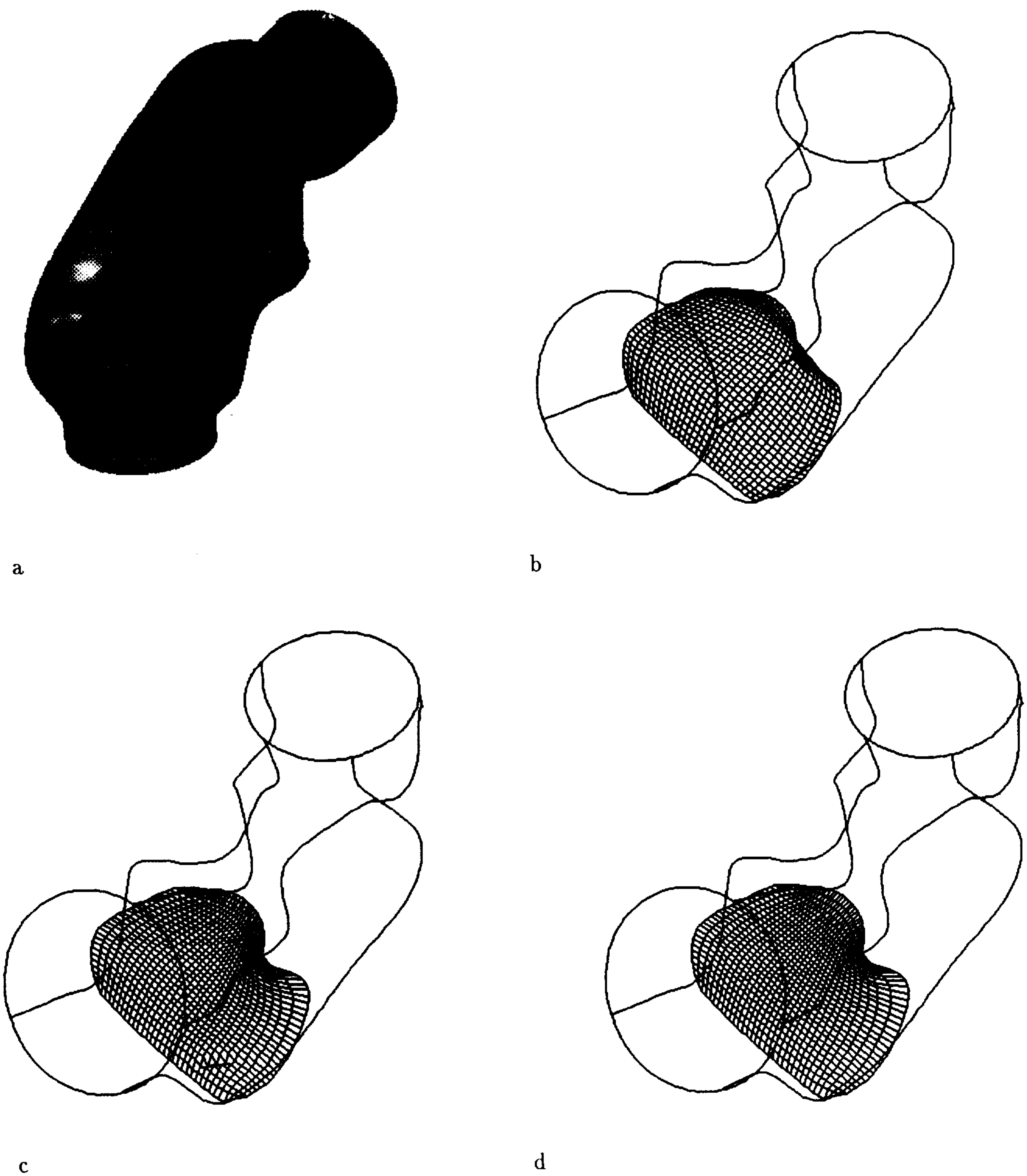
REFERENCES

1. Steinbrenner, J. P.; and Chawner, J. R.: " Recent Enhancements to the GRIDGEN Structured Grid Generation System," in *Software Systems for Surface Modeling and Grid Generation*, NASA CP-3143, NASA Langley Research Center, April 1992, pp. 253-271.

2. Sorenson, R. L.: *The 3DGRAPE Book: Theory, Users' Manual, Examples*, NASA TM-102224, NASA Ames Research Center, July, 1989.

3. Eisman, P. R.; Lu, N.; Jiang, M.; and Thompson, J. F.: "Algebraic - Elliptic Grid Generation", in *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, Weatherill, N. P.; Eisman, P. R.; Häuser, J.; and Thompson, J. F. (eds.), Pineridge Press, Swansea, 1994, pp. 37-48.

4. Stüben, K.; and Linden, J.: "Multigrid Methods: An Overview with Emphasis on Grid Generation Processes", in *Numerical Grid Generation in Computational Fluid Dynamics*, Häuser, J.; and Taylor, C. (eds.), Pineridge Press, Swansea, 1986, pp. 483-509.



a b c d

Figure 1. (a) Duct geometry, (b) initial surface grid, and surface grids after 10 iterations starting with (c) initial grid and (d) converged coarse grid solution.

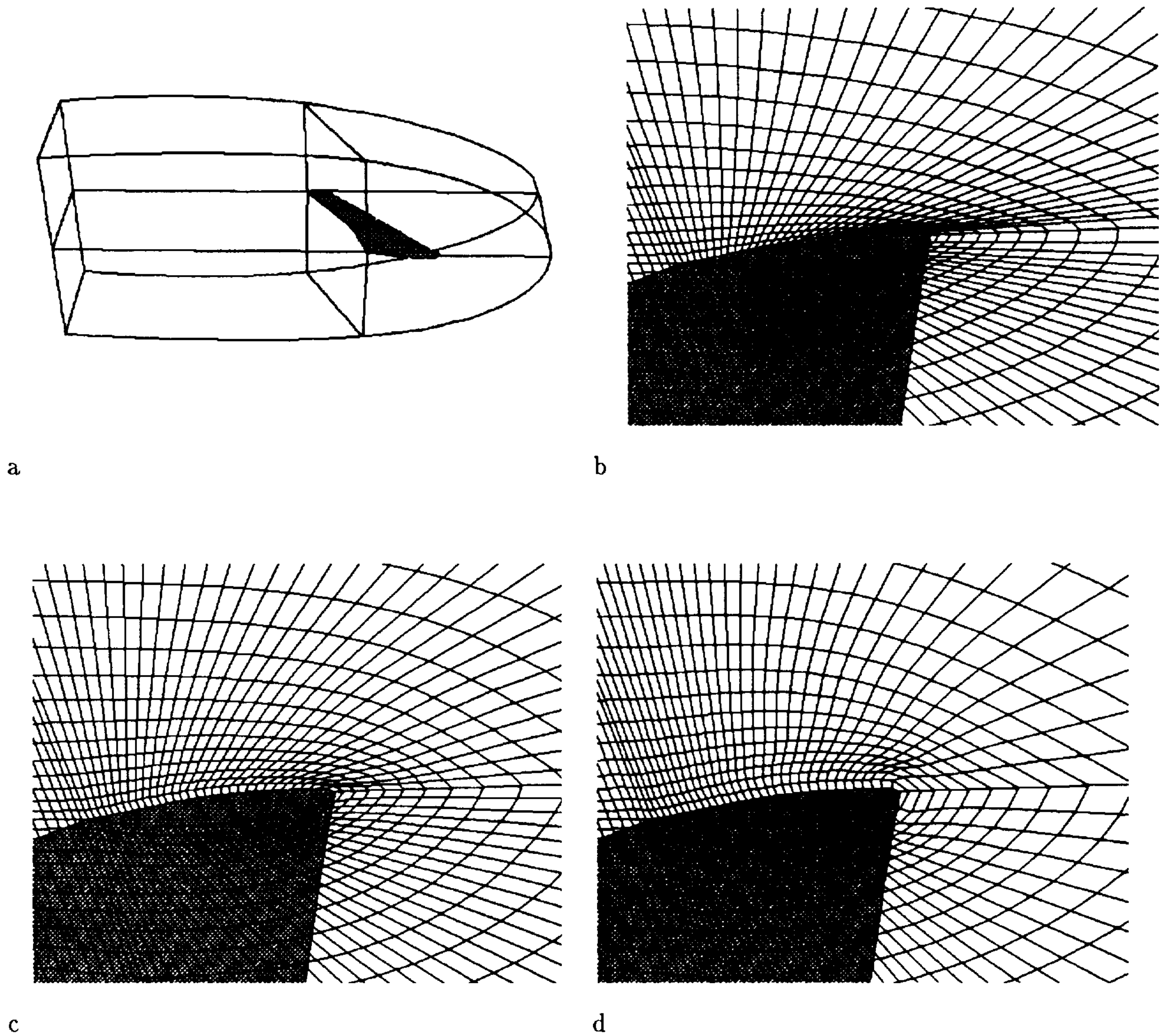
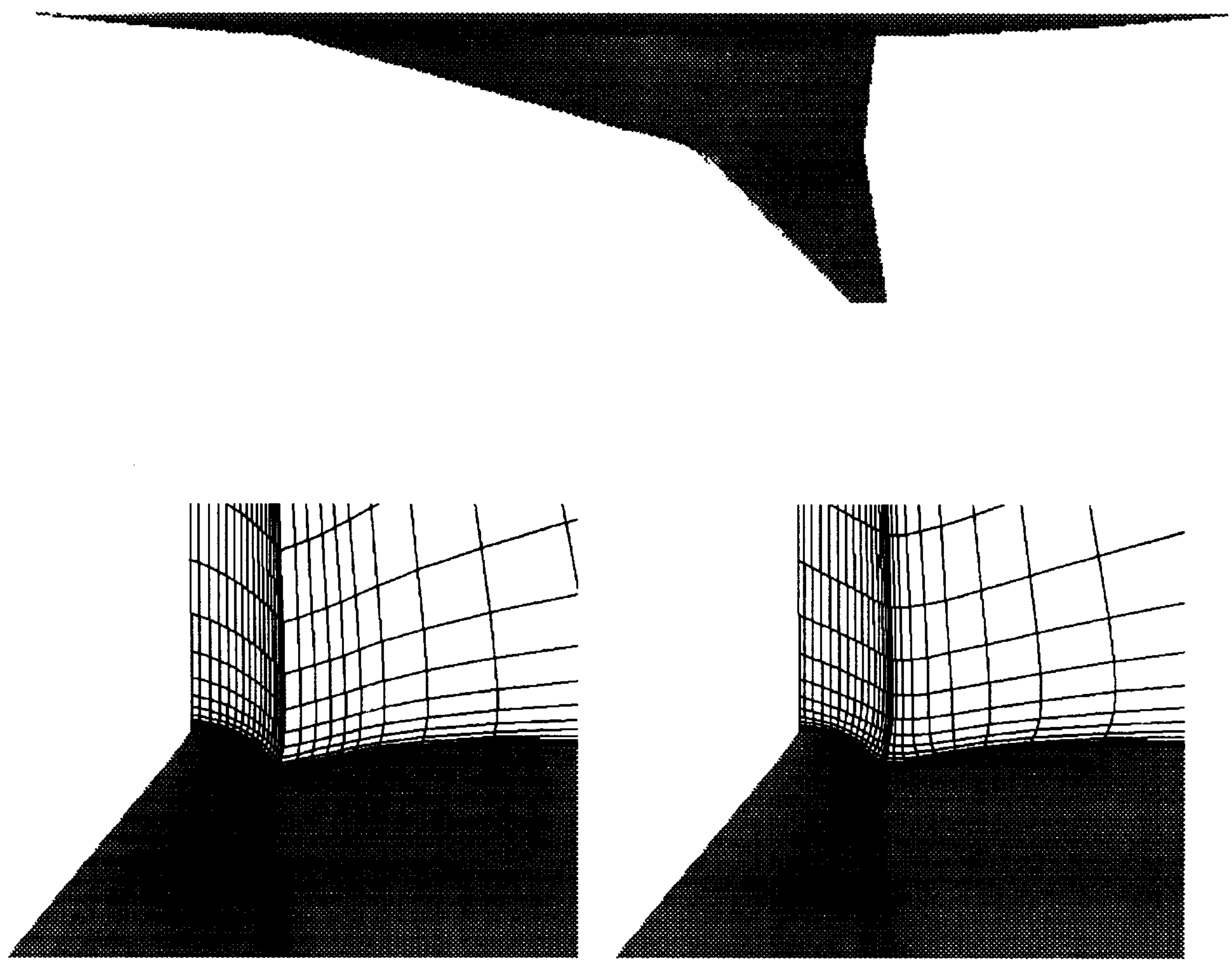


Figure 2. (a) Region about wing, (b) initial surface grid at wing tip, and surface grids after 10 iterations starting with (c) initial grid and (d) converged coarse grid solution.



a

b

c

Figure 3. (a) HSCT configuration, and (b) initial and (c) elliptic coarse grid at wing fuselage junction.