THE DEVELOPMENT AND APPLICATION OF THE SELF-ADAPTIVE GRID CODE, SAGE

Carol B. Davies
Ames Research Center
Moffett Field, California

INTRODUCTION

Solution-adaptive grid methods have become an important tool in the field of Computational Fluid Dynamics for accurately computing flow solutions. The adaption procedures are applied to the grid after an initial flow solution has been computed, using the solution to produce a more rational distribution of grid points. There are two basic adaption methods: redistribution and refinement. In redistribution schemes, points are intelligently moved to more appropriate locations; in refinement schemes, points are added to the existing grid. In either case, solution errors are reduced by minimizing grid discretization errors.

The approach used in the development of the Self-Adaptive Grid codE, SAGE, is based on a grid point redistribution scheme. Gnoffo (ref. 1) first introduced a one-dimensional method (1-D) method for the redistribution of grid points based on local flow gradients. This method is analogous to finding the equilibrium position of a system of springs that connect adjacent nodes with spring (or tension) forces that are proportional to the local error or weight function. This spring analogy, with the proper choice of weight function, results in a simple system of algebraic equations. Nakahashi and Deiwert (ref. 2) formulated an appropriate weight function that not only was proportional to local flow gradients but also provided grid control through user-specified minimum and maximum grid spacing, thus introducing the "self-adaptive" nature to the process. This method was extended to two and three dimensions (ref. 3) by approximating the resulting system of equations as a series of 1-D problems. However, the associated adapted grid was not smooth between the adapted lines and Nakahashi and Deiwert therefore introduced grid-smoothing functions which are analogous to torsion springs. This procedure was found to be efficient and fast and it allowed the user to control the quality of the grid while performing the solution adaption.

The user-friendly SAGE has been developed using this method (ref. 4) and applied to a wide variety of 2- and 3-dimensional flow problems (refs. 5 and 6). The present paper describes the development of the basic adaptive procedure as utilized by the code and the application of the code to a variety of flow problems. Results are shown that clearly demonstrate the ability of the adaptive grid scheme to enhance the solutions of both 2- and 3-D flows.

FORMULATION OF THE ADAPTIVE GRID SCHEME

The adaption procedure is analogous to applying tension and torsion spring forces proportional to the local flow gradient at every grid point and finding the equilibrium position of the resulting system. The multidimensional problem of grid adaption is split into a series of 1-D problems along the computational coordinate lines. The reduced 1-D problem is solved as a tridiagonal system to find the location of grid points along a given coordinate line. Multidirectional adaption is achieved by the sequential application of the 1-D method in each coordinate direction.

The tension forces direct the redistribution of points to the strong gradient regions. The torsion forces relate information between the family of lines adjacent to one another, in order to maintain smoothness and a measure of orthogonality of grid lines. These smoothness and orthogonality constraints are direction dependent, since they relate only the coordinate lines that are being adapted to the neighboring lines that have already been adapted. This implies that the solutions are nonunique and depend on the order and direction of adaption.

The adaption procedure is illustrated in figure 1: three constant k planes of an initial grid are shown in figure 1(a) for which a flow-field solution has been obtained. The points in this grid are now adapted to the computed flow solution, starting on the first line $j = 1$ on the lower plane $k = 1$. In figure 1(b), the first plane has already been adapted and the second plane is the current adaption plane. The current adaption line j is shown, with previous lines already adapted and subsequent lines awaiting adaption. The third plane is still in its original form. Adaption is performed in this line-by-line within a plane-by-plane manner until all requested planes are complete. It is then possible to perform an adaption in a second direction, adapting "on top" of the already adapted grid. The number and order of adaptions are arbitrary and depend on the type of flow problem and the purpose of the adaption. However, for clarity, the analysis in this report assumes that all adaptions are performed in the order shown in figure 1.

Figure 2 shows a segment of the current adaption line in more detail. The lower plane has already been adapted and the upper plane is currently being adapted. Four forces control the redistribution of

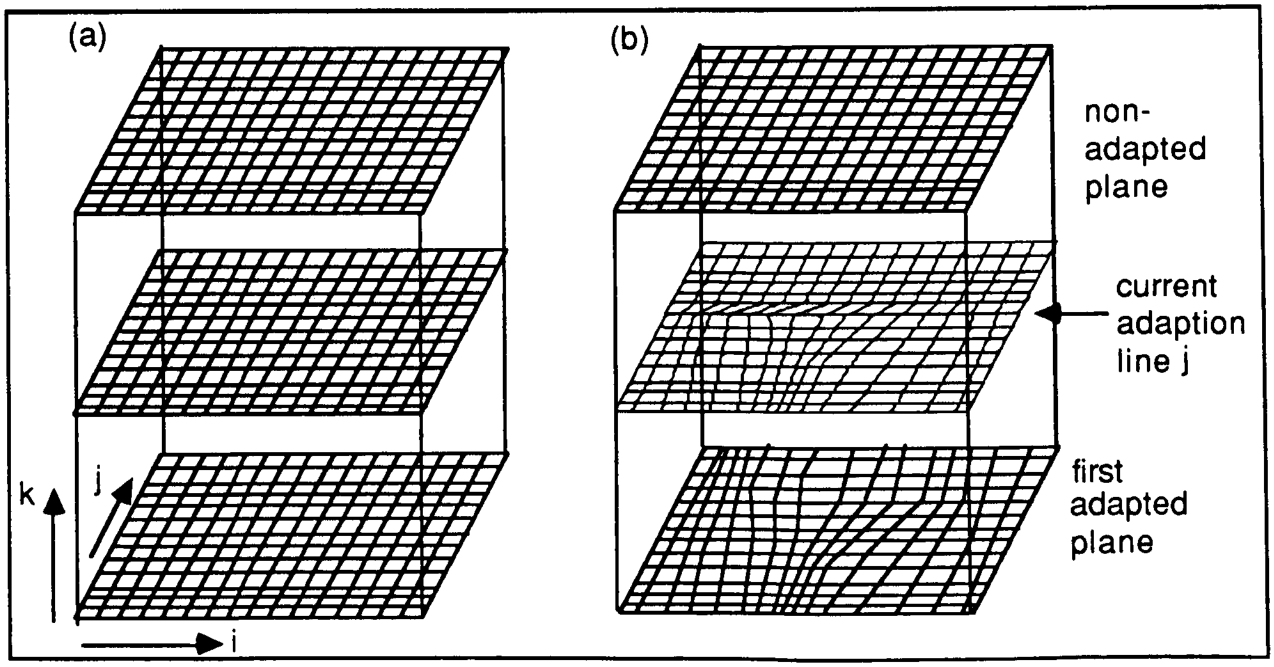

Figure 1. 3-D adaption. (a) Initial grid, (b) first directional adaption.

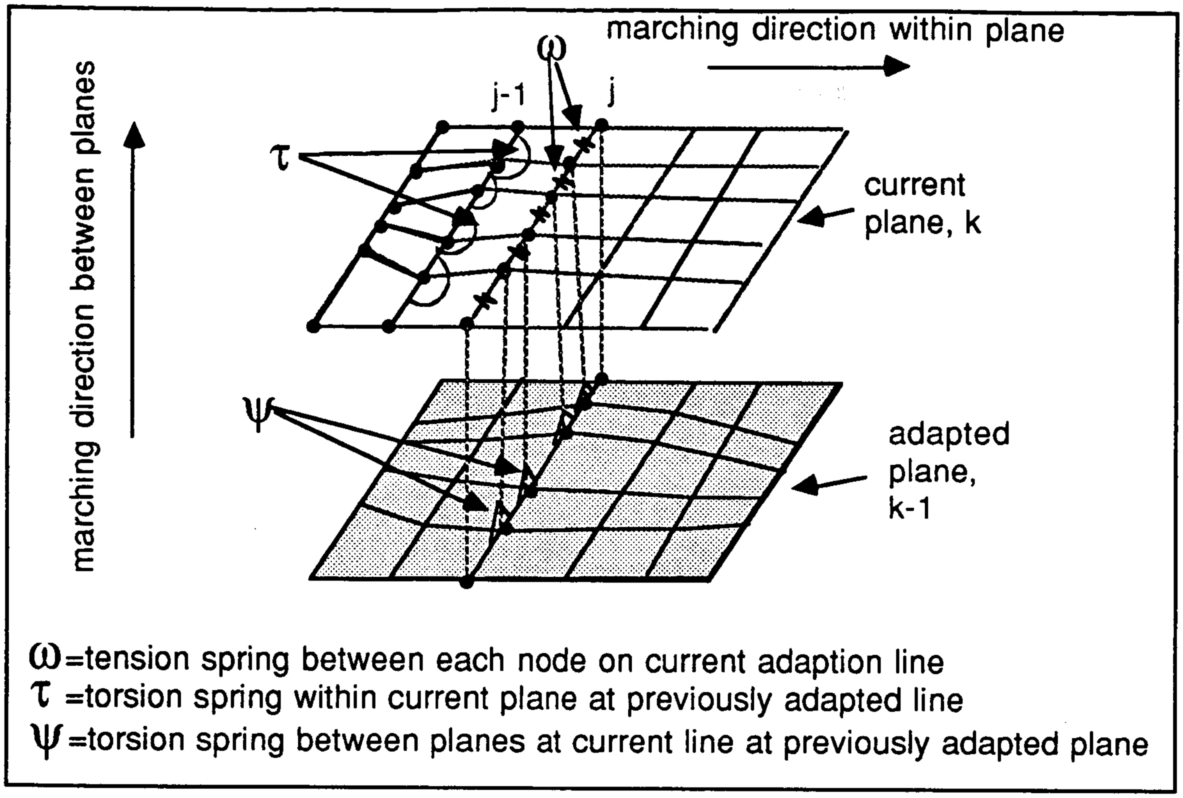
ω
marching direction within plane
j-1
j
τ
current plane, k
marching direction between planes
ψ
adapted plane, k-1
ω=tension spring between each node on current adaption line
τ =torsion spring within current plane at previously adapted line
ψ=torsion spring between planes at current line at previously adapted plane

Figure 2. Line-by-line adaption, showing tension and torsion forces.

a point along a line: the two tension springs that act on each side of a node, and the two torsion springs that control the smoothness of the grid. The tension forces, ω, have the effect of clustering the redistributed points into the high gradient regions.The torsion forces (τ and ψ) maintain continuity between sequentially adapted lines. As shown in figure 2, the τ force acts from the previously adapted line within the current plane and the ψ force acts from the previously adapted line in the preceding plane.

The following analysis describes the development of the multi-dimensional adaption from the simple 1-D algorithm, to the added complexities of two and three dimensions. Although the description is segmented for clarity, SAGE is multi-dimensional and can adapt 1-, 2- and 3-D grids.

ONE-DIMENSIONAL ANALYSIS

The first step in the formulation of the adaption algorithm is to consider the adaption of a single line where torsional constraints do not exist. This 1-D analysis and example are given to illustrate the approach; in practice this method is used only along the initial line of a 2- or 3-D problem.

Figure 3 shows a line, where the arc length at P, (i.e. s_i), along the current adaption line, *j*, is defined as

$$s_{i+1} = s_i + \Delta s_i \qquad (1)$$

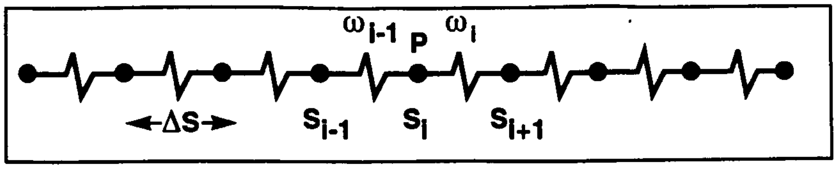

Figure 3. 1-D adaption line, showing initial spacing and tension springs.

A tension force, ω, is defined to act between each node such that

$$\omega_i \Delta s_{i+1} = \text{constant} \tag{2}$$

where ω_i is a function based on the normalized flow gradient $\bar{f}$, and

$$\omega_i = 1.0 + A\,\bar{f}^{\,B} \tag{3}$$

The constants A and B are directly related to the grid spacing and maintain the grid intervals to within the requested minimum and maximum limits (Δs_{MIN} and Δs_{MAX}). Put simply, equation (2) states that the larger the flow gradient, the denser the mesh spacing. This equation is written for each node on the line, giving a 1-D formulation that can be solved directly for Δs_i. Taking the sum of both sides of equation (2) gives

$$\sum \Delta s_i = s_{max} = K \sum 1/\omega_i \text{ giving } K = s_{max} \Big/ \sum 1/\omega_i \tag{4}$$

Substituting back in equation (2), we obtain

$$\Delta s_i = s_{max} \Big/ \left(\omega_i \sum 1/\omega_i\right) \tag{5}$$

and from equation (1), we can obtain the value of each s_i using an iterative procedure.

One-Dimensional Application

Example 1. 1-D sine wave– Figure 4 shows a simple example of this 1-D process. The flow function is given as a sine wave with 51 points on the initial line that are equally spaced and displayed by the filled symbols. The user-requested spacing-control parameters, Δs_{MAX} and Δs_{MIN}, are given to be 0.1 and 0.002 respectively (i.e., a ratio of 50/1). The 1-D adaption algorithm intelligently redistributes the points so that clustering occurs in the high gradient regions. These redistributed points are presented as the open symbols on the constant line in the figure.

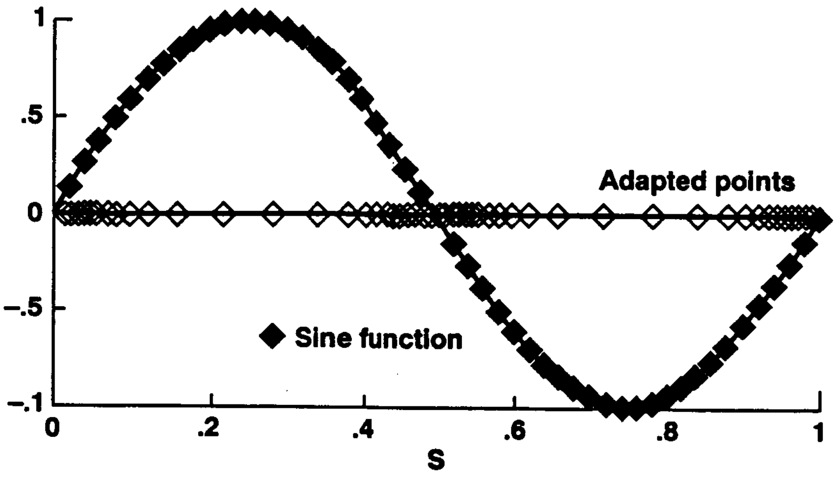

Figure 4. Adaption of a 1-D sine wave.

TWO-DIMENSIONAL ANALYSIS

The continuation of the 1-D approach for successive line-by-line adaptions does not create a mesh that is sufficiently smooth for input into computational flow-field codes. To ensure a more reasonable grid, the redistribution of points (driven by tension springs) is constrained by torsion springs. Within a 2-D plane, a torsion parameter τ is defined that represents the magnitude of the torsion force that maintains smoothness and orthogonality between the node *(i,j,k)* and the nodes *(i,j – 1,k)* and *(i,j – 2,k)*. This torsion force is evaluated as $\tau_i(s'_i - s_i)$ where τ defines the magnitude and s' the direction (i.e., orthogonality and smoothness) of the torsion force. To introduce the torsion forces to the system of equations, equation (2) can be rewritten to represent the force balance, i.e.,

$$\omega_i(s_{i+1} - s_i) - \omega_{i-1}(s_i - s_{i-1}) = 0 \qquad (6)$$

Adding the torsion term gives

$$\omega_i(s_{i+1} - s_i) - \omega_{i-1}(s_i - s_{i-1}) + \tau_i(s'_i - s_i) = 0 \qquad (7)$$

which can then be rearranged to give

$$\omega_{i-1} s_{i-1} - (\omega_i + \omega_{i-1} + \tau_i) s_i + \omega_i s_{i+1} = -\tau s'_i \qquad (8)$$

This equation is written for each interval along the adaption line, producing a system of n – 1 equations that can be expressed as a tridiagonal matrix.

Two-Dimensional Applications

To maintain the integrity of the adapted grid, the user has the choice of several grid quality control variables. These include: the extent of the adaption domain; the maximum and minimum mesh spacings; the proportion of straightness to orthogonality of the grid lines and the magnitude of the torsional effect. In addition, boundary spacing and grid matching controls are available to accommodate patched, zonal and multiple grids. Finally, there is the choice of stepping direction. As described earlier, the adaption process is a line-by-line stepping process and for 2-D applications, there will be four choices of adaption direction: stepping in the *i* direction, the *j* direction, or both (in either order). Each of these options will produce a different adapted grid, however there is no way to quantify which of these will produce the "best" adapted grid. It has been shown (ref. 6) that limiting the adaption to only one direction can produce adequately adapted grids. However, for problems with distinct two-directional flow features, experimenting with the order of adaption may be necessary.

The following two examples of 2-D adaption demonstrate both these cases: the supersonic inlet problem, despite the cross-flow nature of the features, is adequately adapted in one direction only. The more complex case of an axisymmetric nozzle plume flow requires a limited adaption in the second direction.

Example 2: Supersonic inlet– The first 2-D illustration of the adaption procedure is a supersonic channel flow with shock reflection and expansion features. Supersonic viscous flow through a variable-width channel was simulated by Abrahamson (ref. 7) using the thin-layer Navier-Stokes equations. The initial grid is shown in Figure 5(a) and the resulting density contours in Figure 5(b). Initial clustering has been imposed in the streamwise direction at the corners of the upper wall and in the transverse direction near the channel walls. Flow is from left to right, creating a shock from the upstream upper wall corner that reflects off the lower wall. An expansion fan emanates from the downstream upper wall corner and interacts with the reflected shock downstream.

The initial grid is adapted to the density flow gradients, stepping in one direction only, producing the adapted distribution shown in figure 6(a). The adaption procedure has moved the grid points so that the dense mesh spacing now closely follows the high gradient regions across the shocks. This adapted grid, along with the interpolated flow variables, were re-input to the flow solver, producing a new flow-field solution shown in figure 6(b). The improvement in the resolution of the incident and reflected shocks is considerable when compared to the initial solution.

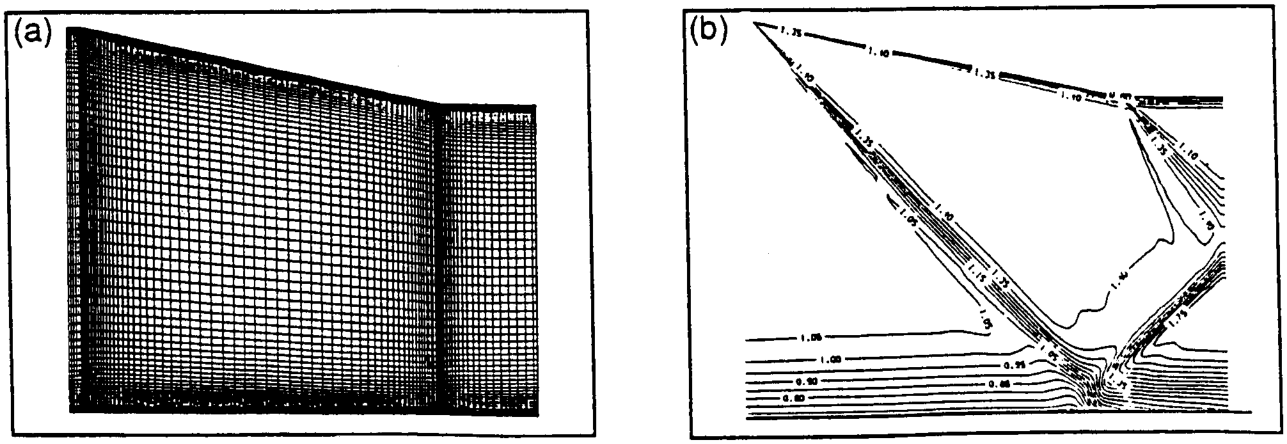

Figure 5. 2-D flow through a supersonic inlet. (a) Initial grid, (b) density contours.

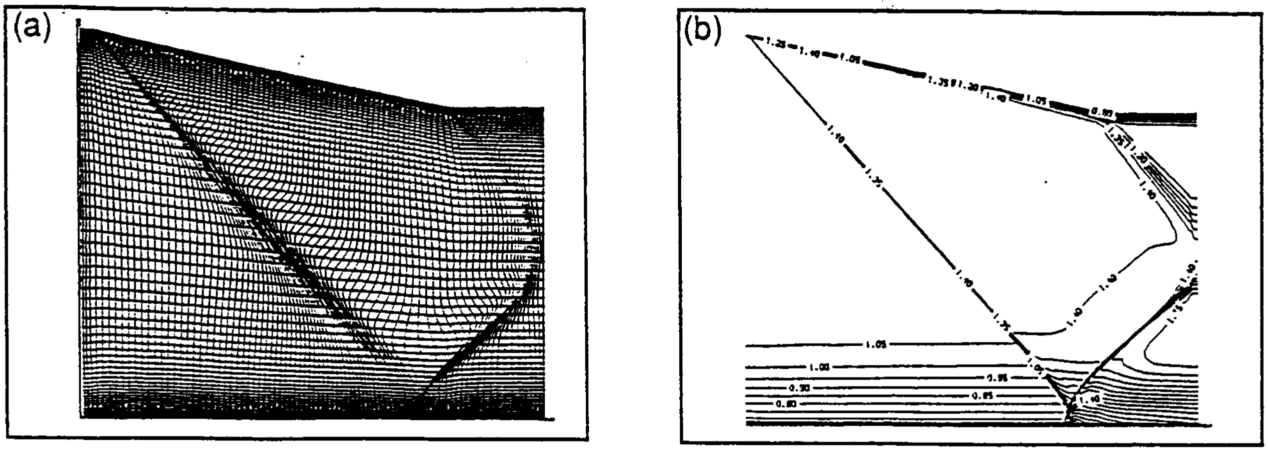

Figure 6. 2-D supersonic inlet. (a) Adapted grid, (b) solution computed on adapted grid.

Example 3: Axisymmetric plume flow– The second 2-D example is an axisymmetric nozzle plume flow (ref. 8). Supersonic flow exits from the nozzle on the lower left side of the grid and interacts with the surrounding quiescent air, causing complex plume features to develop. The upper part of figure 7(a) shows the initial grid used to compute this flow; below, in mirror image, are the Mach contours computed using this initial grid. Because of the lack of grid points in the appropriate regions, this solution does not clearly capture the flow features seen in the experimental results (ref. 8).

Several iterations through the grid solver and SAGE were required to create the final adapted grid and the significantly improved flow solution shown in figure 7(b). Two-directional adaption was required in order to capture the Mach disk whose gradient is in the opposite direction to the other features. The accuracy of the solution is illustrated with a comparison to experimental data: the picture in figure 7(c) is a shadowgraph of the actual experiment and is almost mirrored by the computed solution shown below.

THREE-DIMENSIONAL ANALYSIS

For the extension of the algorithm to three dimensions, a second torsion parameter, ψ is introduced to constrain the movement between the current node *(i,j,k)* and the nodes *(i,j,k – 1)* and *(i,j,k – 2)* on adjoining computational planes. This second torsion term is defined as $\psi_i(s^*_i - s_i)$ and is added to equation (7). After rearrangement, the final equation becomes

$$\omega_{i-1}s_{i-1} - (\omega_i + \omega_{i-1} + \tau_i + \psi_i)\, s_i + \omega_i s_{i+1} = -\tau s'_i - \psi s^*_i \qquad (9)$$

In SAGE, this is the final equation that is solved for all cases. When a 2-D grid is indicated, the ψ terms are simply set to zero. The full details of the solution method can be found in reference 4.

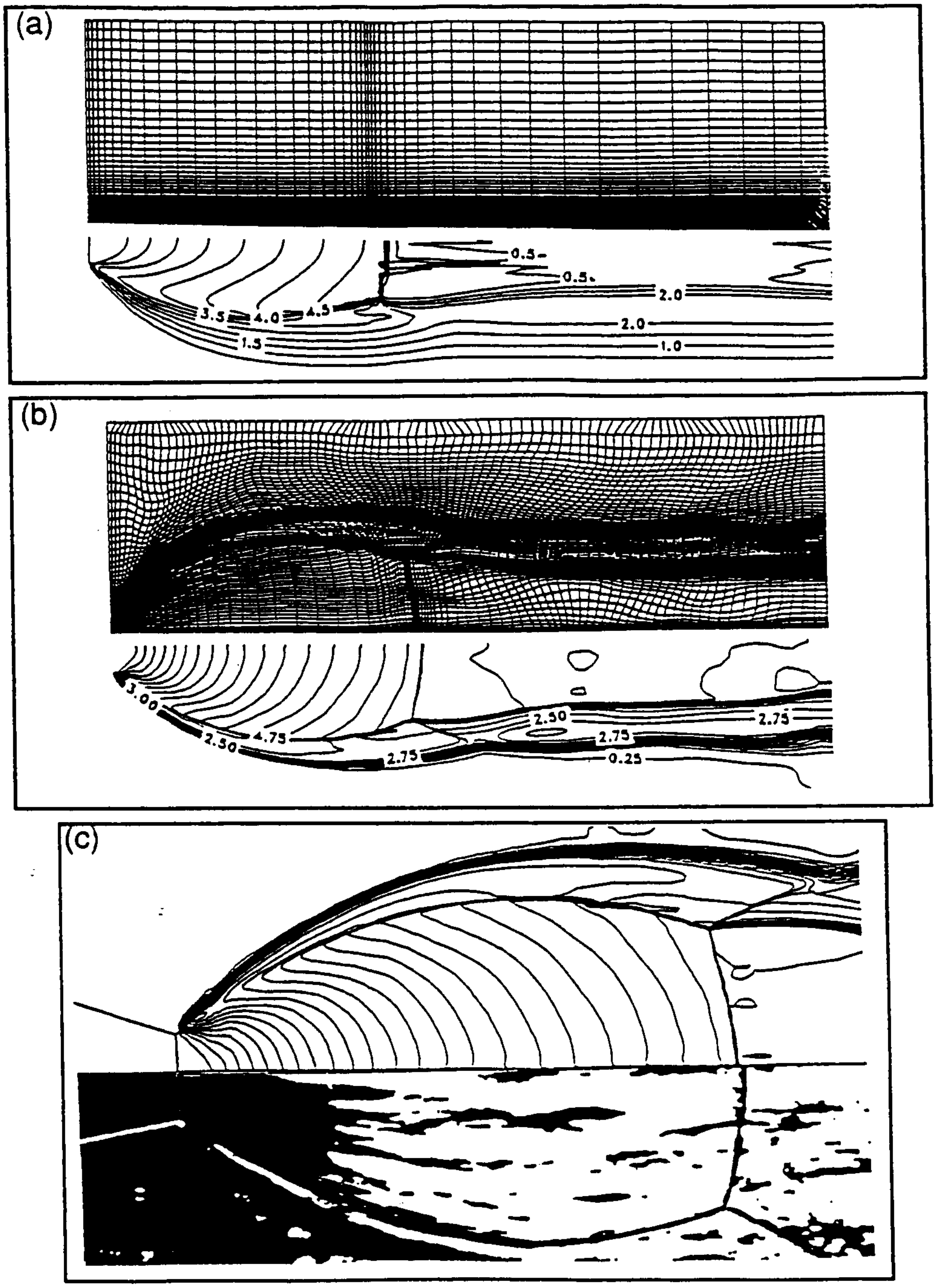

Figure 7. Axisymmetric plume flow. (a) Initial grid and undeveloped solution, (b) final adapted grid and solution, (c) comparison of computed axisymmetric plume flow with experimental shadowgraph result.

Three-Dimensional Applications

For 3-D adaption, the choice of adaption control parameters is expanded to include the direction and magnitude of the torsion force acting between planes. However, the most difficult of the expanded choices is that of the direction, number and order of adaptions. With more complex flow structures, the apparent need for two- or even three-directional adaption increases. However, even a one-directional adaption will significantly change the grid in all three directions as demonstrated by this first example.

Example 4: Supersonic air injection– The problem shown in figure 8 is the University of Virginia combustor-flow experimental test (ref. 9). The figure shows a Mach 2 flow over a backward-facing step, with two sonic transverse air jets behind the step. An outline of the grid used for this computation is shown in the figure by dotted lines. The flow data was provided by Dr. Jong H. Wang of Rockwell International. Figure 8(b) shows three selected planes *i,j,k* from the initial grid that have been separated for clarity in the remaining figures. Figure 8(c) shows the corresponding Mach contours from the initial flowfield solution that were used as the adaption variable. Adaption takes place on constant *j* planes with marching in the *i* direction within the plane. This implies that the points were redistributed along *k* lines. Figure 8(d) shows the adapted planes, and although only one adaption pass has been made, both the *i* and *j* planes are adapted with respect to Mach number, with these planes retaining their original "flat" 3-D surfaces. However, the *k* plane shows a different effect; points have not moved within the plane, but the 3-D surface has deformed, since points have shifted in the *k* direction through the adaption.

It is also interesting to look at the effect of two-directional adaption. Figure 8(e) shows the result of a second adaption, this time adapting in the *i* direction and marching in *k* lines, "on top" of the adaption already shown in figure 8(d); points have been redistributed in the *i* direction as well as the *k* direction and both the *i* and *k* planes are now curved surfaces. This *i* redistribution can also occur by marching in *j* within the constant *k* planes. This alternate adapted grid (not shown) gives a very similar redistribution on the *j* plane, even though that was not the adaption plane. Other combinations were also tested that confirmed that the choice of adaption order may not be crucial to the creation of an acceptable adapted grid. In addition, we have seen that both one- and two-directional adaptions can produce an adapted grid that sufficiently represents the complex 3-D flow features.

Example 5: Supersonic flow over a 3-D swept ramp– The geometry of a 3-D ramp is shown in figure 9(a), where the dotted lines outline the computational grid. The recompression shock at the ramp corner and the viscous-inviscid interaction with the boundary layer results in a large separation that is three-dimensional in nature. Solutions have been computed for two ramp angles, 15° and 30°, and compared to experimental oil flow patterns (ref. 10). For the 15°-ramp-angle case, the experimental and the computed results agreed well. However, the results obtained for the 30° case, computed on the initial grid shown in figure 9(b), were not as close. This is demonstrated in figure 9(c), where the computed oil-flow patterns along the ramp surface show a significant difference in the size of the separation zone from the experimental values (shown by the open symbols). This discrepancy was suspected to be caused by local grid inaccuracies, which suggested utilizing the adaption procedure. The adapted grid, shown in figure 9(d), was able to more accurately capture the interaction between the shocks from the separated region and the wall boundary layer. This improved interaction generated a larger separation zone than before, which is seen in the recomputed oil-flow patterns shown in figure 9(e). This example demonstrates how a one-directional adaption pass influences the entire 3-D flow solution. The ramp surface plane was not adapted directly, but nevertheless, the flow on this surface changed considerably.

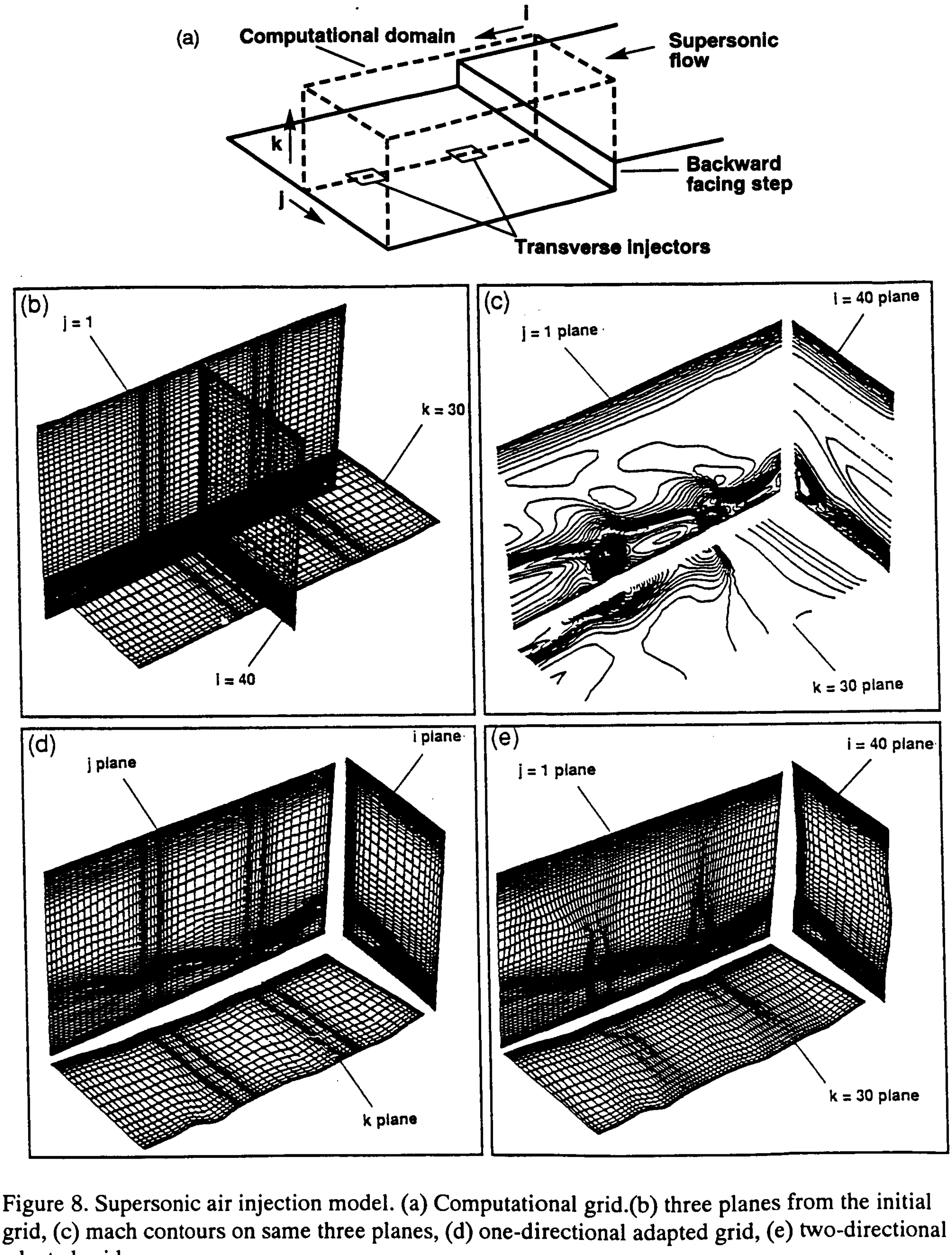

Figure 8. Supersonic air injection model. (a) Computational grid.(b) three planes from the initial grid, (c) mach contours on same three planes, (d) one-directional adapted grid, (e) two-directional adapted grid.

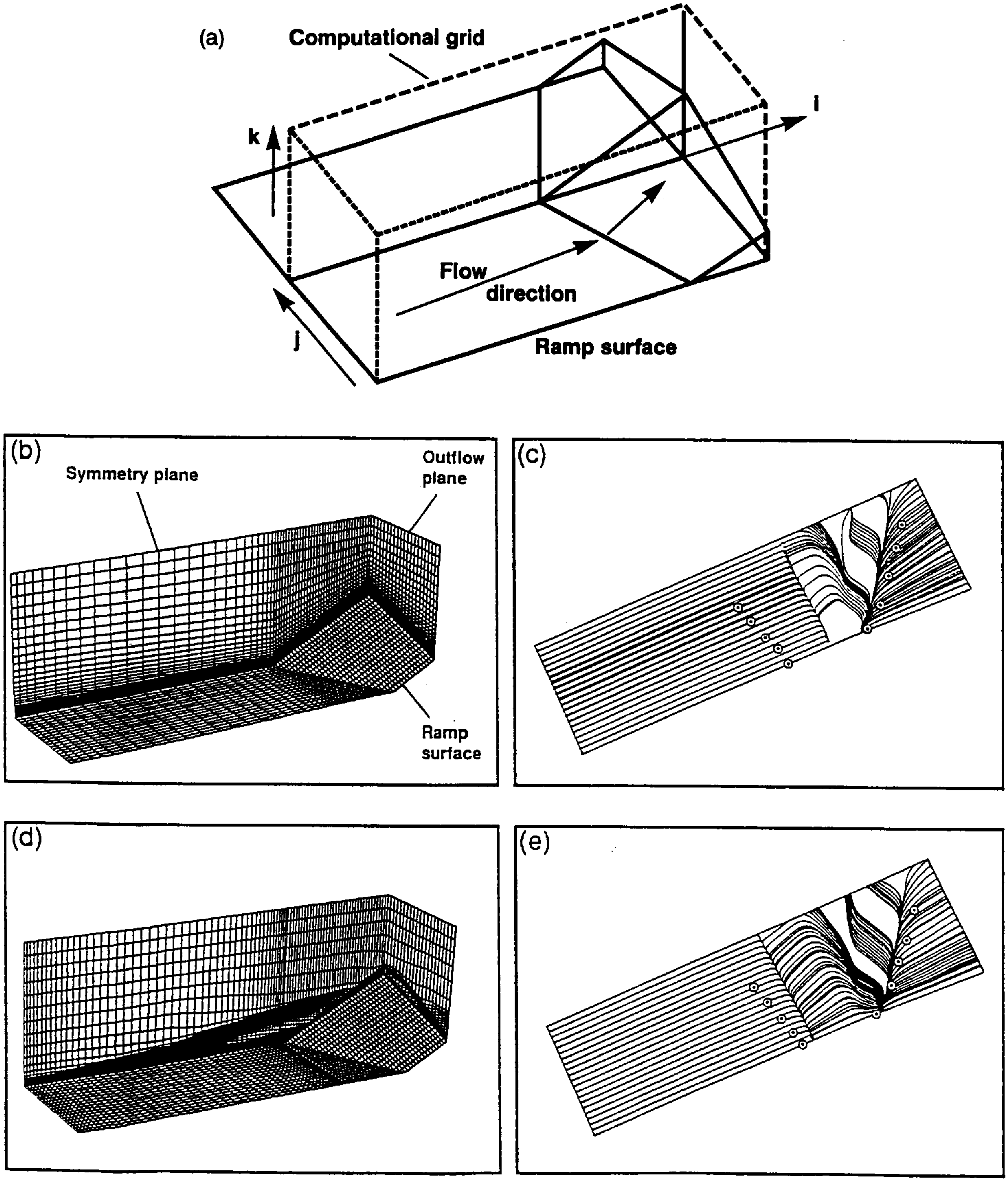

Figure 9. 3-D 30° swept ramp. (a) Experimental model, (b) initial grid, (c) oil-flow patterns, comparing computed and experimental separation regions, (d) adapted grid, (e) oil-flow patterns from adapted solution.

Example 6: Hypersonic NASP nozzle-plume flow– Ideal gas solutions have been obtained using a multiple, 3-D grid topology for the supersonic flow around a generic National Aero-Space Plane (NASP) nozzle configuration known as the single exhaust ramp nozzle (SERN) (ref. 11). The 3-D model is shown in figure 10 along with an outline of the computational zone in the plume region. This grid is one of the eight separate overlapping grids that were required to model this problem and this example demonstrates the flexibility of using SAGE for adapting segments of multiple grids. Figure 11(a) shows three planes (the symmetry plane, the outflow plane and the plane containing the ramp surface) from the original grid that covered the plume region. Figure 11(b) shows the Mach contours from the flow-field solutions obtained on this grid (ref. 11). It is clear that the interaction of the plume shocks with the ramp and the edge flow is complex and three-dimensional in nature. Although most of the flow features were observed, they were not all well-defined and thus the adaption procedure was invoked. The initial grid was adapted in two stages: first the points were redistributed in two directions, based on Mach number; then points were moved from the outer, constant region to the more complex interior flow region.

The resulting adapted grid is shown in figure 11(c). The inflow plane (not shown) and the lower ramp plane were not adapted, to maintain continuity between this grid and the unadapted grids surrounding it. Smoothness between the nonadapted planes and the internal adapted planes was maintained by the merging process described in reference 4. The flow solver used this adapted grid to obtain the result shown in figure 11(d). It can be seen that the flow features have sharpened considerably. Figure 12 compares the experimental shadowgraph (ref. 12) to the computed Mach contours along the symmetry plane downstream of the nozzle exit. The external and internal plume shocks are captured well and the agreement between the computation and the experiment is seen to be good.

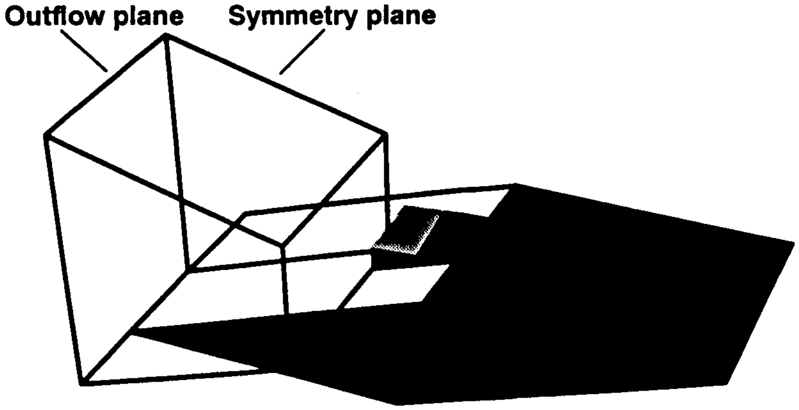

Figure 10. SERN experimental model showing computational grid in plume region.

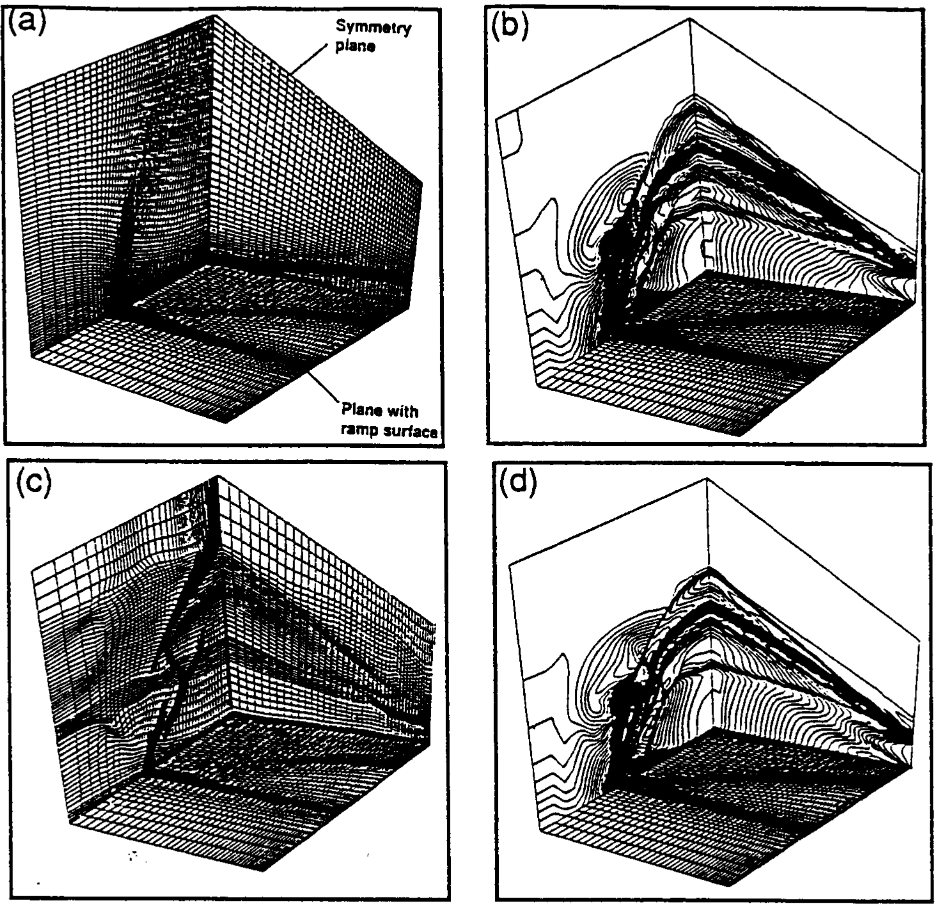

Figure 11. Nozzle plume flow. (a) Initial grid showing symmetry plane, outflow plane, and ramp surface, (b) initial Mach contours, (c) adapted grid, (d) adapted Mach contours.

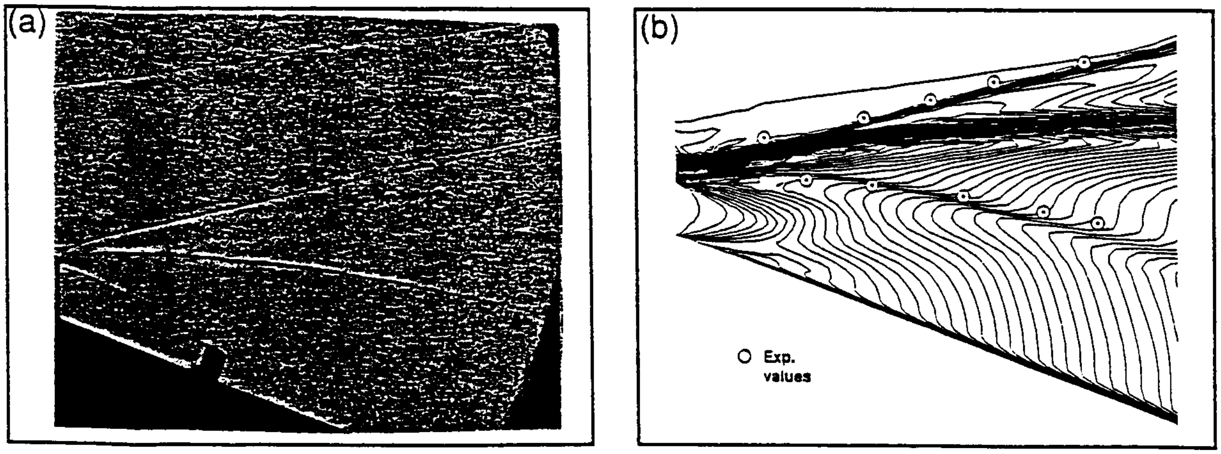

Figure 12. Comparison along the symmetry plane of the SERN nozzle flow. (a) Experimental shadowgraph, (b) computed Mach contours.

CONCLUSIONS

The multidimensional self-adaptive grid code, SAGE, has proven to be a flexible and useful tool in the solution of complex flow problems. Both 2- and 3-D examples given in this report show the code to be reliable and to substantially improve flowfield solutions. Since the adaptive procedure is a marching scheme the code is extremely fast and uses insignificant CPU time compared to the corresponding flow solver. The SAGE program is also machine and flow solver independent. Significant effort was made to simplify user interaction, though some parameters still need to be chosen with care. It is also difficult to tell when the adaption process has provided its best possible solution. This is particularly true if no experimental data are available or if there is a lack of theoretical understanding of the flow. Another difficulty occurs if local features are important but missing in the original grid; the adaption to this solution will not result in any improvement, and only grid refinement can result in an improved solution. These are complex issues that need to be explored within the context of each specific problem.

REFERENCES

1. Gnoffo, P.: A Vectorized, Finite-Volume, Adaptive Grid Algorithm Applied to Planetary Entry Problems. AIAA Paper 82-1018.

2. Nakahashi, K. and Deiwert, G. S.: A Practical Adaptive Grid Method for Complex Fluid Flow Problems. NASA TM-85989, 1984.

3. Nakahashi, K. and Deiwert, G. S.: A Three-Dimensional Adaptive Grid Method. AIAA Paper 85-0486.

4. Davies, C. B. and Venkatapathy, E.: The Multi-Dimensional Self-Adaptive Grid Code, SAGE. NASA TM-103905, 1992.

5. Davies, C. B. and Venkatapathy, E.: Application of a Solution Adaptive Grid Scheme, SAGE, to Complex Three-Dimensional Flows. AIAA Paper 91-1594, 10th Applied Computational Fluid Dynamics Conference, Honolulu, 1991.

6. Deiwert, G. S.; Venkatapathy, E.; Davies, C. B.; Djomehri, J.; and Abrahamson, K.: Application of a Self-Adaptive Grid Method to Complex Flows. NASA TM-102223.

7. Abrahamson, K. W.: Numerical Investigation of a Mach 3.5 Axisymmetric Inlet with Multiple Bleed Zones. AIAA Paper 88-2588, 6th Applied Aerodynamics Conference, June 1988, Williamsburg, Va.

8. Venkatapathy, E. and Feiereisen, W. J.: Computational Studies of Hard-Body and 3-D Effects in Plume Flows. AIAA Paper 89-0129, 1989.

9. Fletcher, D. G. and McDaniel, J. C.: Quantitative Characterization of a Non-Reacting, Supersonic Combustor Using Laser-Induced Iodine Fluorescence. AIAA Paper 89-2565.

10. Venkatapathy, E., "Computational Results for 2-D and 3-D Ramp Flows with an Upwind Navier-Stokes Solver," to appear in the Proceedings of the Hypersonic Workshop, Springer-Verlag.

11. Ruffin, S. and Venkatapathy, E., "Computational Design Aspects of a NASP Nozzle/Afterbody Experiment," AIAA Paper 89-0446, 1989.

12. Ruffin, S. M., Venkatapathy, E., Lee, S. H., Keener, E. R., and Spaid, F. W., "Single Expansion Ramp Nozzle Simulations," AIAA Paper 92-0387, Reno, Jan. 1992.

BIOGRAPHY

Carol B. Davies was born and educated in England. She attended Reading University and received her B.S. in Mathematics in 1966. After postgraduate studies in Meteorology she emigrated to the USA and started work at NASA Ames as a contractor in 1968. From 1968 to 1976 she worked on a variety of projects, including analyzing experimental data from the Pioneer Space Project and the development of an Inlet Boundary Layer code. In 1976, she moved to the Republic of Singapore and joined the University of Singapore. There, she taught Statistics and was a consultant in the Computer Science Dept. On her return to the USA in 1980, she rejoined NASA Ames as a contractor with Sterling Software. She has since worked on several projects including particle spallation from the Galileo Space Probe and the development of Aero-assisted Space Transfer Vehicles (ASTV), for which she received a patent and a Space Act Award.