NASA
Technical
Paper
3526

1994

# An Assessment of the Adaptive Unstructured Tetrahedral Grid, Euler Flow Solver Code FELISA

M. Jahed Djomehri
*Calspan Corporation*
*Moffett Field, California*

Larry L. Erickson
*Ames Research Center*
*Moffett Field, California*

National Aeronautics and
Space Administration

**Ames Research Center**
Moffett Field, California 94035-1000

# An Assessment of the Adaptive Unstructured Tetrahedral Grid, Euler Flow Solver Code FELISA

M. Jahed Djomehri* and Larry L. Erickson**

Ames Research Center

## SUMMARY

A three-dimensional solution-adaptive Euler flow solver for unstructured tetrahedral meshes is assessed, and the accuracy and efficiency of the method for predicting sonic boom pressure signatures about simple generic models are demonstrated. Comparison of computational and wind tunnel data and enhancement of numerical solutions by means of grid adaptivity are discussed. The mesh generation is based on the advancing front technique. The FELISA code consists of two solvers, the Taylor–Galerkin and the Runge–Kutta–Galerkin schemes, both of which are spatially discretized by the usual Galerkin weighted residual finite-element methods but with different explicit time-marching schemes to steady state. The solution-adaptive grid procedure is based on either remeshing or mesh refinement techniques. An alternative geometry adaptive procedure is also incorporated.

## INTRODUCTION

The degree of flow field complexity surrounding high-speed aircraft is primarily related to the aircraft's geometry and speed. The flow field may consist of one or more complex, highly nonlinear regions containing shocks, shear layers, concentrated vortices, etc., and zones of interactions among these.

One major obstacle in the application of CFD to realistic aircraft is proper discretizaton of the physical space, known as grid generation—a procedure that has recently been characterized as a bottleneck overall in CFD applications (ref. 1). Despite the vast effort that has been made to develop various sophisticated structured and unstructured grid packages in recent years (ref. 2), the time required for grid generation still makes up a large portion of a typical CFD application. For instance, with the state-of-the-art multiblock structured-grid-generation software package (ref. 3), the time required to generate a suitable grid for a complex geometry is on the order of several months. Alternatively, recent methods based on unstructured grids have shown promise of reducing the time to weeks. These approaches have not yet become practical in generating cells of high aspect ratio (ref. 1), such as those required in boundary layers and wake regions of viscous flow calculations where flow gradients are strongly one-dimensional (1-D), with large Reynold numbers (differences of several orders of magnitude in normal and streamwise discretization). Nevertheless, they have demonstrated strong potential in handling complex geometries for inviscid problems. Many important and interesting features of the flow about high-speed aircraft, such as shock formation and flow separation from sharp edges, are essentially inviscid, rotational physical processes.

To resolve the grid requirement for integrated viscous and inviscid calculations, some recent works (refs. 4–7) have adopted the strategy of combining the structured and unstructured grids such that local regions of the structured grid are embedded within a global unstructured environment. Interfaces between regions have either one-to-one or overlapped connections. Both cases require special procedures to transfer flow variables across interfaces which are known as the internal boundary conditions. For overlapped connections, interpolatory procedures are used. For one-to-one connections, flow quantities at the interface computed from the structured grid region are simply passed to the unstructured grid zone.

The increased flexibility offered by unstructured grids in achieving automated grid software should not be the only motive behind their use. They can also be used to achieve enhanced solution algorithms by techniques known as solution adaptive grids. These methods (refs. 8–12) have been the subject of active

*Calspan Corporation.

**Aerodynamics Division.

research in recent years and have proved to be practical tools in many two- and three-dimensional problems. They enhance solution algorithms without excessively increasing the global number of grid points and/or employing further complex high-order resolution schemes (refs. 13–16).

Structured grid methods are essentially limited by the effort to preserve orthogonality and smoothness of the grid lines and by the lack of ability to refine or unrefine the grid locally. The latter feature is crucial in dealing with complex flows around realistic configurations. The unstuctured grid data system easily lends itself to development of such schemes.

This work provides an assessment of the unstructured grid code FELISA. It allows 3-D modeling of inviscid flows about complete aircraft. The code was initially developed at the University College of Swansea and Imperial College and has been supported through grants with the NASA Langley and Ames Research Centers. The package consists of a grid generator, a flow solver, and a solution-adaptive grid generator. A brief description of the method follows in subsequent sections of this report. The code has been tested on geometrically simple 3-D problems ranging from transonic to supersonic speeds. Five applications are discussed in this report. Three of these test cases are generic configurations that have been used since the study, in the early 1970s, of sonic boom problems where the computational data are compared with wind tunnel results. These configurations consist of a low-aspect-ratio rectangular wing with parabolic sections, a cone-cylinder, and a delta wing-body with a leading-edge sweep of 69 deg with double wedge sections. This work was motivated by national interest in the study of the feasibility of supersonic transports.

## GRID ALGORITHM

Among various unstructured grid-generation algorithms, two categories predominate: those based on the Voronoi/Delauny technique (refs. 11 and 12) and those based on the Advancing Front technique (refs. 9 and 10). The major differences are that the former is more efficient and easily used but does not exploit the benefit of the irregularity of the grid as much as the latter does. That is, with Voronoi/Delauny types, it is assumed that points of triangulation have been given in advance by a set of nonrandom points. These points usually form an ordered set similar to the structured grid data, whereas with the Advancing Front types, new grid points are generated according to user-input grid-spacing parameters, allowing variable grid spacing and better control.

The grid-generation scheme here is based on the Advancing Front method. This method is discussed in detail by Peraire, et al. (ref. 9). It uses the concept of a background grid to define the spatial variation of the element base length, $\delta$, the stretching factor, $s$, and the stretching direction, $\alpha$. The background grid is a coarse grid formed by four noded linear tetrahedra covering the whole domain, and includes the far field of interest. The user should supply the values of the parameters at every node of the background grid; indeed, for every direction, $\alpha$, there are two associated orthogonal directions and two related stretching parameters whose magnitudes are proportional or equal to the factor, $s$. These parameters within the domain are obtained by linear interpolation over the pertinent nodal values of tetrahedra supplied by the background grid. Grid clustering can also be furnished by means of functional representations, such as exponentials defined in the neighborhood of user-specified points or line segments. The coordinate information pertinent to these points or lines is supplied through background grid data.

The solution domain is bounded by a collection of "boundary surfaces" that define the confining flow field boundaries. These surfaces define the geometry of the model, the symmetry planes, and the far-field boundary surfaces. Each boundary surface is an $m$ by $n$ network of topologically rectangular nonself-intersecting surfaces (networks). The points are ordered in a fashion that defines a clockwise direction as the positive orientation on the surface, with its normal pointing into the field. Boundaries of each network consist of finite curves defined between two endpoints by a number of points that naturally impose a positive orientation along the curve. One boundary edge in each network can be collapsed to length zero. Each boundary curve, also called "boundary edge," is a smooth curve (i.e., it contains no "kinks") whose line segments are common between two networks. Every boundary edge always identifies an abutment between two and only two networks. Care must be taken that the data defining each network consist of nonskewed, topologically subrectangular cells with aspect ratio no larger than ten. A sufficient number of points must be given to properly represent the curvatures of the edges and surfaces at areas containing bends, twists, etc. Depending on the complexity of the model, the process of preparing the input geometry data file could be time consuming. This task could be automated, because of the similarity between the geometrical format used by FELISA to represent the model and that of the standard CAD/CAM systems.

Analytical representations of boundary curves and surfaces are accomplished by composite cubic and bicubic (a tensor product) interpolation. The surface triangulation process begins with the discretization of the boundary according to local grid spacings obtained from the background grid. The initial front for each surface region is formed by the segments joining two successive boundary points. The smallest segment on the front is selected to construct a new triangle constrained by the parameters $\delta$, $s$, and $\alpha$. The front, then, is updated accordingly. The process is repeated until there is no segment left on the front. Once surface triangulations are completed, their assembly forms the initial front for generating the new tetrahedral points in the field. The procedure continues (ref. 17) in the same fashion as the triangulations.

Although the choice of grid parameters for the Advancing Front method offers considerable freedom in achieving variable grid spacing in the field, the numeric values should be chosen to avoid skewness in the triangulation process. Discretization of the surface should properly represent scales and curvatures of the model; deviation could introduce spurious wavelets and pressure distributions which could severely perturb near and far field flow characteristics.

In general, there is no rigorous approach for predicting the grid spacing parameters. The considerable effort by which a proper grid can be generated may present quite an inconvenience. Repeated trials are necessary to set these parameters. Unsmooth distribution of parameters on the nodes of the background grid often leads to physically impossible conditions so that the code cannot complete the surface/volume discretization. The fronts become corrupted and/or the procedure falls into infinite loops of deleting and generating points in an attempt to correct itself. The procedure is highly sensitive to the adjustment of directional stretching parameters. These parameters are initially set to unit vectors for almost all applications in this work, with the stretching parameters about unity. With this approach, generation of smooth grids with variable grid spacing directionally have been found challenging. Some stumbling blocks such as those mentioned above, which are fatal to grid completion, could be easily alleviated by local treatment procedures. One could incorporate a measure (not currently included) to detect the quality of the elements after or any time during grid generation, and then disassemble bad elements and a few layers of their closest neighbors, leaving a cavity in the grid with a new front. Then the Advancing Front procedure could be reinvoked to patch the grid. The procedure could be reiterated several times if neccessary: each time, further neighbors should be disassembled and local grid-spacing parameters should be smoothed in order to avoid creation of the same elements.

The current method has been most successful in the presence of uniform and slowly varying parameters where one must exercise less tuning, but the result is a relatively larger number of tetrahedral elements. This, of course, is evident in view of the fact that algorithmic procedures of the Advancing Front method are inherently based on a uniform triangulation procedure. For configurations with highly variable geometrical scales and curvatures—such as surfaces with relatively small thicknesses, leading and trailing edges, slender conical bodies with small half-cone angles—one should ensure that grid-spacing parameters are sufficiently small to allow proper discretization so that the characteristic element length varies slowly across regions of differing scales. To generate a satisfactory initial grid, preserving the curvature of the geometries and ensuring a smoothly varying grid across the field, users will need to develop expertise in the use of the various parameters in the code.

Various ad hoc approaches have been used to construct the background grid. One relatively easy approach is to specify grid parameters on a planar mesh of triangular elements lying on a plane orthogonal to the Cartesian coordinate axis $x$, $y$, or $z$. Then, by moving the plane parallel to itself or rotating it about a coordinate axis, one generates layers of 6-noded prisms, easily subdivided into three tetrahedral elements, which cover the entire domain. The grid-spacing parameters are specified on each node of the planar mesh and identically copied to translated/rotated nodes of tetrahedra. This approach is a simple extension of 2-D parameters to 3-D. Although quite efficient, it will not offer the desired grid spacing for many 3-D geometries. To alleviate this problem, one can redefine the nodal grid parameters for those nodes of the background grid whose coordinates fall in certain user-specified regions of the space. The regions may simply be defined by the maximum/minimum of $(x, y, z)$ coordinates confining the region and/or limiting conical angles in space.

Another simple approach implemented in this application is the construction of a coarse background grid by subdividing a box which inscribes the domain using a structured Cartesian grid where hexahedral elements are subdivided into six tetrahedra. The default grid parameters at each node of the structured grid are modified either locally or according to the user-specified region as described above. Pirzadeh (ref. 18) has introduced another interesting approach for background grid construction that is based on uniform Cartesian meshes where tetrahedral elements can be formed as described but the grid parameters are determined by solutions of Poisson equations with specified discrete source terms, as in heat conduction problems. This approach gives a smooth variation of grid parameters and has proved practical.

## ADAPTIVE METHODS

Solution adaptive methods are efficient schemes to enhance solution algorithms; they have been used extensively in recent years. Three major categories of techniques that have been applied to 3-D flow problems are mesh movements (ref. 8), mesh refinement (ref. 19) and remeshing (ref. 20). The main procedure in all these approaches is (1) to identify an error or an adaptivity criterion as a measure of the solution error and (2) to use the criterion as a driving mechanisim to concentrate or delete grid points in areas of high or low gradient.

Refinement and remeshing techniques are both used in the current FELISA code. These procedures are applied in a batch-mode fashion for steady state problems. That is, at prescribed time intervals, the computed solution data on the current grid are passed to the adaptive procedure, where a new solution adaptive grid is generated. The flow solver again is applied and a new solution on the new grid is obtained. This procedure may be repeated several times until a satisfactory solution is reached. For problems with an initially large number of elements, only one or two repetitions of the procedure can be practically exercised, because of restrictions on CPU time, memory, and access to the system.

The adaptive criteria for the remeshing approach follows the 1-D concept of equidistribution of the error. That is, the product of a measure of the second-order derivative of a prescribed key physical variable, $\sigma$, such as density, and a measure of the element length at each grid point is set to be constant over the entire field. One can then predict the element length for each cell of the adapted mesh locally in a 1-D sense. For 3-D problems, the matrix of quadratic coefficients (i.e., the matrix of second-order derivatives) at each grid point is computed and the magnitude of eigenvalues and the directions of the corresponding eigenvectors of this matrix are used to implement the above equidistribution concept locally. A new background grid with elements and grid points equal to those of the initial grid is generated. The local directional stretching parameters of the new background grid at each point are computed in the local principal coordinate directions, i.e., the eigenvectors mentioned.

The adaptive criteria for the refinement approach are much simpler than for the remeshing one. They are merely based on the subdivision of the marked tetrahedral elements. A measure of the gradient that is based on the magnitude of divided differences of a key variable is computed over all edges of the elements. Subsequently the elements are marked for possible subdivision if their magnitudes fall above (or below) a prescibed threshhold value. This procedure is not based on the equidistribution concept described earlier. However, the repeated application of the program would eventually lead to equidistribution of the error in the field. Depending on the number of edges marked per element, each element is subdivided into eight, four, or two elements. This approach has the tendency to increase the number of elements by a factor of eight of the number of marked elements. For example, an initial grid of 500,000 elements that has only one-tenth of its elements marked for refinement would yield a final grid of nearly 1,000,000 elements. The number of elements in overall application could be optimized if further accurate measures of error and/or efficient derefinement schemes were used. The code does not currently have a derefinement capability.

The current adaptive programs as applied to some of the examples shown in this work have frequently failed to complete the procedure. For instance, for two examples, the slender cone-cylinder and the wing-body problem, the remeshing approach failed to regenerate the new grid for the various prudent ranges of parameters specified by the authors. The root of the problem seemed to rest with the nonsmooth parametrizations of the background grid which consequently generated nonsmooth surface triangulations of the body and/or tetrahedralization of the field by the Advancing Front method. Application of the refinement approach was also unsuccessful when applied to the wing-body example because the code miscalculated coordinates of the boundary points. Here a problem arises for proper calculation of the coordinates of the subdivision points that are supposed to lie on the surface of the model. Coordinates of subdivision points, which are normally located at the midpoints of tetrahedral edges, can be computed by a simple arithmetic average of coordinates of the corresponding two vertices of the edge. For the calculation of midpoint coordinates of boundary edges, however, additional precautions are necessary to assure that they lie on the proper boundary surfaces. The parametric representation of these surfaces, which has been employed in surface grid generation procedures, is used to determine the coordinates of new boundary points. This portion of the refinement algorithm has been found erratic when applied to the wing-body example above.

Regions consisting of weak shocks are difficult to detect properly by the usual error measures, particularly in the presence of stronger shocks and/or expansions. In solving cone-cylinder and wing-body problems we have observed that the conical weak shocks at the apex of the cone had not been effectively detected in either of the above adaptive programs. Numerical experiments with relevant parameters that are designed to control the density of grid points in the critical regions had failed to properly cluster grid points in this region. To alleviate this problem, the user naturally would have to adjust the tolerance of the relevant parameters in a particular range in order to recover the region of weak shocks, but this would adversely affect the cluster grid points in a large portion of the domain. Thus, in such circumstances, this approach has a tendency to excessively increase the total number of grid points in the field.

Adaptive grid results for the problem mentioned have been obtained by the simple implementation of the geometry-adaptive grid concept in conjunction with the refinement approach. The geometry-adaptive grid approach is simple and practical only for those cases in which the user has a priori knowledge of the critical regions of the field, and in which the bounding surfaces of these regions are smooth and can be readily characterized by simple geometrical definitions, such as for annular or angular regions. For instance, the conical shock regions at the apex of the cone example can easily be defined by two limiting conical regions specified by the pertinent conical angles about the axis of symmetry. One can then single out all the elements that fall in this region to be refined. Geometrical features of simple critical regions can be obtained based on a posteriori knowledge of shock positions predicted by solutions obtained on initially nonadapted coarse grids.

An alternative procedure in the geometry-adaptive grid approach can be used. The functional representation mechanism for grid clustering, discussed in the beginning of the previous section, is exploited by the geometry-adaptive grid concept. That is, positions of points and lines in the background grid can be determined in accordance with the geometrical shape of the critical regions. For example, by revolution of a directed line segment about the symmetry axis of the body, one can define a finite number of uniformly distributed lines in space for the background grid. These lines can then be used by the grid generator to concentrate grid points in a conical-type region about the specified line segments.

The geometry-adaptive approach can be extended to cluster grid points in other geometrically complex, critical regions if the limiting surfaces associated with these regions can be easily identified. An approximate shape of the critical regions can be predicted by the solution data obtained on an initially coarse grid. This approach should be thought of as a complementary procedure to overcome certain shortcomings of the solution adaptive programs, not as a replacement for these programs.

## FLOW EQUATIONS

The mathematical flow model used here is the conservative law form for inviscid compressible fluids referred to as the Euler Equations. In 3-D space these equations are expressed as follows:

$$\frac{\partial U}{\partial t} + \sum_{i=1}^{3} \frac{\partial F_i(U)}{\partial x_i} = 0 \qquad (1)$$

where $U = (\rho, \rho u, \rho v, \rho w, \rho\epsilon)^T$, and

$$F_1 = \left| \begin{array}{c} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(\rho\epsilon + p) \end{array} \right|, \quad F_2 = \left| \begin{array}{c} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(\rho\epsilon + p) \end{array} \right|,$$

$$F_3 = \left| \begin{array}{c} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(\rho\epsilon + p) \end{array} \right| \tag{2}$$

where $\rho$ is the density, and the Cartesian velocity components of velocity field $\vec{V}$ are $(u, v, w)$ in the $(x, y, z)$ direction, respectively. The equivalent notations for velocity vector field $(u_1, u_2, u_3) \equiv (u, v, w)$ and coordinates $(x_1, x_2, x_3) \equiv (x, y, z)$ are also implied. With the ideal gas assumption, the pressure $p$ and total energy per unit volume $e = \rho\epsilon$ can be expressed as

$$p = (\gamma - 1)(e - \frac{1}{2}\rho|\vec{V}|^2) \tag{3}$$

with $\gamma$, the ratio of specific heat, equal to 1.4 in air.

## SOLUTION SCHEMES

The flow algorithms for unstructured grid data can be classified in two major groups—finite element (refs. 16, 17, 21, and 22) and finite volume (refs. 14, 15, 23, and 24) schemes. No pronounced advantages or disadvantages in the application of either of these techniques to practical multidimensional problems have been demonstrated as yet. Further specific studies and comparisons of computed solutions with experimental data are required. The FELISA code consists of two solvers, the Taylor–Galerkin (TG) and the Runge–Kutta–Galerkin (RK) schemes. Both are spatially discretized by the usual Galerkin weighted residual finite-element methods but with different explicit time marching schemes to steady state.

### Taylor–Galerkin Scheme

Some prominent features of the Taylor–Galerkin technique can be outlined as follows. The physical domain of interest, $\Omega$, consists of an assembly of nonoverlapping tetrahedral elements, which constitute the finite-element mesh. The vertices of the tetrahedra are referred to here as nodes. The interpolation (or shape) functions $N_j$ associated with nodes $j$, which take the value unity at $j$ and zero at the other nodes, are considered as the space of test or weight functions. In FELISA, the shape functions are to be piecewise linear for all the flow variables $(\rho, \rho u, \rho v, \rho w, \rho\epsilon)$. The Galerkin weighted residual statement of the flow equations, equation (1), is simply the integral expression

$$\int_\Omega \left( \frac{\partial U}{\partial t} + \sum_{i=1}^{3} \frac{\partial F_i}{\partial x_i} \right) N_j d\Omega = 0,$$

$$j = 1, \ldots, NODES \tag{4}$$

where $NODES$ is the total number of nodes in the field. This is known as the weak form of the equations. With the use of the Gaussian divergence theorem it can be restated as

$$\int_\Omega \frac{\partial U}{\partial t} = \int_\Omega \left( \sum_{i=1}^{3} F_i \frac{\partial N_j}{\partial x_i} \right) d\Omega$$

$$- \int_\Gamma \left( \sum_{i=1}^{3} F_i n_i \right) N_j d\Gamma \tag{5}$$

where $n \equiv (n_1, n_2, n_3)$ denotes the outward unit normal to the boundary surface $\Gamma$ of domain $\Omega$.

Linear finite-element interpolatory expressions $U^*$ and $F_i^*$ for the solution and flux vector are given by

$$U^*(x, t) = \sum_{k \in \Omega} U_k^*(t) N_k(x), \text{ and}$$

$$F_i^* = \sum_{k \in \Omega} F_{ik} N_k(x) \tag{6}$$

where $F_{ik} \equiv F_i(U_k^*)$. The expression $k \in \Omega$ reads as "every node $k$ in $\Omega$." A semidiscrete system of equations for nodal values $U_k^*$ would result, upon insertion of these expressions in equation (5), in

$$\sum_{k \in \Omega_j} \left( \int_{\Omega_j} N_j N_k d_\Omega \right) \frac{dU_k^*}{dt}$$

$$= \int_{\Omega_j} \left( \sum_{i=1}^{3} F_i^* \frac{\partial N_j}{\partial x_i} \right) d\Omega - \int_\Gamma \left( \sum_{i=1}^{3} F_i^* \right) N_j d\Gamma$$

$$\tag{7}$$

where $\Omega_j$ is the subdomain of all elements containing node $j$. This is apparent from definition of the shape functions; the summation also accounts for the elements surrounding node $j$. These equations form a system of nonlinear ordinary differential equations whose temporal discretizations are discussed later. It should also be noted that the disappearance of the derivatives of flux terms in this variational formulation permits exploitation of the important class of discontinuous constants or linear approximations for computation of the fluxes. Because of the nature of the explicit time integrations, the procedure does not lead to the assembly of the so called "stiffness matrix," as is the case for the finite-element approximation of the elliptic differential equations. The solution vectors on the right side of equation (7) are known from the previous time-level calculation. The volume integral terms are assembled locally on each element, with a single loop over each element (ref. 17) sending its contribution to each node.

## Taylor–Galerkin Time-Marching Scheme

The Taylor–Galerkin discretization is similar to an explicit, central-difference (midpoint quadrature rule), finite-element-based method. Application of the midpoint quadrature rule to equation (7) and elimination of the asterisks gives

$$\sum_{k\in\Omega_j} M_{jk}\frac{U_k^{n+1}-U_k^n}{\Delta t} = \int_{\Omega_j}\left(\sum_{i=1}^{3} F_i^{n+\frac{1}{2}}\frac{\partial N_j}{\partial x_i}\right)d\Omega - \int_\Gamma\left(\sum_{i=1}^{3} F_i^{n+\frac{1}{2}} n_i\right)N_j d\Gamma \qquad (8)$$

where the superscript $n$ denotes time level $t = t^n$, and

$$M_{jk} \equiv \int_{\Omega_j} N_j N_k d\Omega = \sum_{e\in\Omega_{jk}} \int_{\Omega_e} N_j^e N_k^e d\Omega \qquad (9)$$

The quantities $M_{jk}$ are known as entries of the standard (or consistent) finite-element mass matrix. The sub- and superscripts $e$ on $\Omega$ and the shape functions are used to emphasize restriction to this element. The symbol $\Omega_{jk}$ denotes the subdomain of all elements containing nodes $j$ and $k$. The first term on the right side of equations (7) and (8) leads to the expression known as the stiffness matrix, $K_{jk}$. With an approximation for the flux term, as in equation (6), this term can be recast as

$$\sum_{k\in\Omega_j}\left(\sum_{i=1}^{3} F_{ik}K_{jk}\right) \qquad (10)$$

where $K_{jk} \equiv \int_{\Omega_j} N_j \frac{\partial N_k}{\partial x_i} d\Omega$. Computations are performed in a two-step Taylor–Galerkin manner similar to a two-step family of Lax–Wendroff schemes.

The first step requires an approximation for the intermediate value of the solution vector $U^{n+\frac{1}{2}}$, at time level $t^{n+\frac{1}{2}}$. A truncated Taylor expansion of $U$ up to the first order is used, and the time derivative, $\partial U/\partial t$, at $t^n$ is replaced by equation (1), yielding

$$U^{n+\frac{1}{2}} = U^n - \frac{1}{2}\Delta t \sum_{i=1}^{3} \frac{\partial F_i^n}{\partial x_i} \qquad (11)$$

The spatial discretization of this equation is achieved by the Galerkin scheme. Here the piecewise linear approximations for the dependent values $U^n$ and $F_i^n = F_i(U^n)$ are used. The piecewise constant interpolation functions, $P_e$, associated with element $e$, defined as having the value unity on $e$ and zero on the others, are employed for the approximation of $U^{n+\frac{1}{2}}$. The piecewise constant functions are used as weighting functions in the Galerkin formulation

$$U^n = \sum_j U_j^n N_j, \quad F_i^n = \sum_j F_{ij}^n N_j,$$
$$U^{n+\frac{1}{2}} = \sum_{e\in\Omega} U_e^{n+\frac{1}{2}} P_e \qquad (12)$$

where $F_{ij}^n = F_i(U_j^n)$. For piecewise constant weighting functions $P_E$ associated with elements $E$, the variational relation takes the form

$$\sum_{e\in\Omega}\left(\int_\Omega P_e P_E d\Omega\right)U_e^{n+\frac{1}{2}} = \sum_j\left(\int_\Omega N_j P_E d\Omega\right)U_j^n - \frac{1}{2}\Delta t\sum_j\left(\int_\Omega \frac{\partial N_j}{\partial x_i}P_E d\Omega\right)F_{ij}^n \qquad (13)$$

Because $P_E$ is only nonzero on $E$, and the partial derivatives $\partial N_j/\partial x_i$ are constant over each element, equation (11) reduces to

$$V_{\Omega_E} U_E^{n+\frac{1}{2}} = \sum_{j\in\Omega_E} \left(\int_{\Omega_E} N_j d\Omega\right) U_j^n - \frac{1}{2}\Delta t V_{\Omega_E} \sum_{j\in\Omega_E}\left(\sum_{i=1}^{3} \frac{\partial N_j}{\partial x_i} F_{ij}^n\right) \qquad (14)$$

The term $\Omega_E$ restricts the domain of integration over the element $E$, and $V_{\Omega_E}$ denotes its volume. The elemental values $U_E^{n+\frac{1}{2}}$ can then be readily calculated.

These algorithmic procedures have been implemented in the FELISA code in a slightly different and more simple manner. An approximate value of the intermediate solution is sought in terms of the known piecewise linear representation at time $t^n$ and the piecewise constant approximation for the increment $\delta U$, i.e., $\hat{U}^{n+\frac{1}{2}} = U^n + \delta U^{n+\frac{1}{2}}$. An approximation for $U^{n+\frac{1}{2}}$ can be obtained from the same Galerkin procedure when it is applied to the relation $\delta U = -\frac{1}{2}\Delta t \sum \frac{\partial F_i}{\partial x_i}$, with the piecewise constant and linear approximations $\delta U^{n+\frac{1}{2}} = \sum_e \delta U_e^{n+\frac{1}{2}} P_e$ and $F_i^n = \sum_j F_{ij}^n N_j$ for the incremental values and flux terms, respectively. These relations are an interpretation of equation (9). The elemental values of the incremental solution are then obtained, as before, by the Galerkin approach, thus:

$$\delta U_E^{n+\frac{1}{2}} = -\frac{1}{2}\Delta t \sum_{j\in\Omega_E}\left(\sum_{i=1}^{3} \frac{\partial N_j}{\partial x_i} F_{ij}^n\right) \qquad (15)$$

A piecewise linear discontinuous representation of the intermediate solution $\hat{U}_E^{n+\frac{1}{2}}$ within each element $E$, and its pertinent nodal values $\hat{U}_j^{n+\frac{1}{2}}$ are expressed as

$$\hat{U}_E^{n+\frac{1}{2}} = \sum_{j\in\Omega_E} U_j^n N_j + \delta U_E^{n+\frac{1}{2}} \text{ and}$$

$$\hat{U}_j^{n+\frac{1}{2}} = U_j^n + \delta U_E^{n+\frac{1}{2}}, \text{ for each } j \in \Omega_E \qquad (16)$$

The second step requires an approximation of the solution at time level $t^{n+1}$. The Galerkin formulation is applied to a Taylor expansion of $U$ that consists of a first-order remainder term in time interval $\Delta t = t^{n+1} - t^n$. The expansion is expressed as

$$U^{n+1} = U^n - \Delta t \sum_{i=1}^{3} \frac{\partial \hat{F}_i^{n+\frac{1}{2}}}{\partial x_i} \qquad (17)$$

The piecewise linear interpolation functions are used for weighting functions as well as approximations for $U^n$ and $U^{n+1}$. Flux terms at the intermediate time level, however, are produced by the piecewise linear discontinous approximations

$$\hat{F}_{iE}^{n+\frac{1}{2}} = \sum_{j\in\Omega_E} \hat{F}_{ij}^{n+\frac{1}{2}} N_j, \quad \hat{F}_{ij}^{n+\frac{1}{2}} = F_i(\hat{U}_j^{n+\frac{1}{2}}) \qquad (18)$$

where the nodal values of the flux terms, $\hat{F}_{ij}^{n+\frac{1}{2}}$, at the intermediate time level $t^{n+\frac{1}{2}}$ are computed by the values of $\hat{U}_j^{n+\frac{1}{2}}$ obtained in the first step.

The desired nodal values of the solution $U_j^{n+1}$ can be calculated from the Galerkin expression, equation (8), with $F_i^{n+\frac{1}{2}}$ replaced by $\hat{F}_i^{n+\frac{1}{2}}$. The integrals on the right hand side of this expression, denoted by $RES_j$, represent the residual vector of element contributions to node $j$ and can be evaluated explicitly. The expression leads to an implicit algebraic relation for the time increment value

$$\Delta U_k^n = U_k^{n+1} - U_k^n, \sum_{k\in\Omega} M_{jk}\Delta U_k^n = RES_j \qquad (19)$$

and can be solved by a Jacobi iteration procedure originally proposed by Donea (ref. 25). This is done by the simple decomposition $M_{jk} = M_{jk}^L + (M_{jk} - M_{jk}^L)$, where $M_{jk}^L$ denotes the entries of the lumped (diagonal) mass matrix and takes the value of zero if $j \neq k$, and $\sum_{s\in\Omega} M_{js}$ if $j = k$. Equation (19) then takes the form

$$M_{jk}^L\Big|_{j=k} \Delta U_j^{n(r)} = RES_j - \epsilon \sum_{k\in\Omega} \times \left(M_{jk} - M_{jk}^L\Big|_{j=k}\right)\Delta U_k^{n(r-1)} \qquad (20)$$

where $(r)$ denotes the current iteration number and $\epsilon = 1$. Starting with $\Delta U^{n(0)}$, three iterations are usually sufficient for convergence. This concludes the time-marching procedure for one time interval.

Boundary conditions are implemented in the second step of the Taylor–Galerkin scheme through the second integral term—the boundary integral—on the right side of equation (8). The intermediate values calculated in the first step are corrected by applying a characteristic analysis of the linearized Euler equations in a coordinate system tangential and normal to the boundary. The corrector step consists of enforcing subsonic/supersonic inflow, outflow, and solid wall conditions near the boundary. The solver currently accounts for no explicit enforcement of a Kutta condition or other special treatment at the trailing edge. A detailed discussion of the characteristic analysis can be found in a recent book by Hirsch (ref. 26). The resulting pointwise corrected values at boundary points influence the nodal values of the intermediate fluxes. The modification procedure is applied in a post-processing manner where the boundary integral calculations are invoked. This approach incorporates the effect of boundary conditions only in an average, or so-called weak, sense. The inviscid wall boundary condition of zero normal velocity is imposed by projection. That is, the pointwise value of the predicted velocity at the wall is replaced by its projection vector on the wall.

Artificial viscosity terms are added to the right side of equation (8), explicitly, in order to control oscillations, and overshoots in the vicinity of the steep gradients such as shock waves, discontinuities, and vortex sheets, where dissipation effects or shear stresses take place in very thin layers of the flow. A detailed discussion of the numerical dissipation can be found in references 17 and 27. The method has implemented the McCormack modification of Lapidus artificial viscosity, proposed for finite difference, in conjunction with the finite-element solver. The diffusion term here is further simplified to avoid computation of the second-order derivative pressure terms in the linear finite-element approximations, which precludes the expensive process of performing a variational recovery (ref. 27).

The concept of smoothing the solution $U^{n+1}$ at the end of each time step by means of the above Galerkin approximation of the diffusion equation gives

$$U_j^{n+1}\Big|_{\text{smooth}} = U_j^{n+1} + \Delta t_j C_s \Big( \sum_{k\in\Omega_j} M^L_{jk}|_{j=k} \Big)^{-1} \times \Big( \sum_{k\in\Omega_j} (\frac{S_e}{\Delta t_e})\Big|_{e\in\Omega_{jk}} (M_{jk} - M^L_{jk}\,|_{j=k}) U_k^{n+1} \Big) \tag{21}$$

where $C_s$ is a user-specified constant. $S_e$ is the element pressure switch coefficient, whose value is given in terms of the mean of the element nodal values $S_i$:

$$S_i = \sum_{k\in\Omega} \frac{[(M_{ik} - M^L_{ik}\,|_{i=k})P_k]}{|(M_{ik} - M^L_{ik}\,|_{i=k})P_k|} \tag{22}$$

where $P_i$ is the nodal value of pressure at node $i$, and $\Delta t_e$ and $\Delta t_i$ are local elements and nodal time steps whose values (ref. 27) are determined in accordance with Courant-type stability criteria. The node time step, which is used in the second step of the Taylor–Galerkin scheme, is calculated by averaging the element time steps surrounding each node. The element time steps are known from the first step of the Taylor–Galerkin scheme.

Local time-stepping methods are known as valuable tools for accelerating solutions to time-asymptotic steady states. These methods are particularly important in the context of the solution adaptive approach. Meshes of small grid spacings that are concentrated in the critical regions require a very small time step limited by the local Courant number. This requirement would then limit the global time stepping if a spatially fixed time step needed to be used. The reader should, however, be aware of some controversial issues regarding local time stepping which have not been thoroughly proven. Reference 28 reports an anomalous behavior of solutions obtained by the use of spatial-varying time steps. The discussion in reference 28 is of a precautionary nature. For certain applications, it was observed that solutions can lead to nonphysical transients which may eventually converge to a nonphysical asymptotic solution. It was not concluded that this phenomenon is independent of the particular recipe for local varying of the time step, nor were the effects of the computational grids and numerical dissipation terms strongly addressed. In the results presented here, we did not observe this phenomenon, and no attempt was made to study the aforementioned effect pertinent to the FELISA code. A further relevant possibility is that, from the computation of a fictitious transient at each time step, errors could arise which might propagate through the entire solution domain. This effect might slow the convergence rate.

From Fourier analysis of numerical errors (ref. 26), it can be argued that, since local time steps are essentially computed in the close vicinity of the stability limit (unit Courant number everywhere in the domain), greater solution accuracy will be obtained than with the spatially global fixed time step. The latter case, which enables the true transient calculation because of the existence of drastically varying element sizes in the domain, is restricted to computing with Courant numbers much smaller than unity, which is known to render less accurate steady-state solutions. The code has not been provisioned to calculate the real transient or time-accurate solutions, and therefore is only of use for the solution of steady-state problems.

The residual averaging method (ref. 23) has also been implemented in the code to increase the time step. The increment $\Delta U_k$ in equation (20) is replaced by a weighted average of increments at its neighboring nodes and $\epsilon$ is now a small negative number ($\approx -0.4$). Using the weighted average smooths out the residuals in a Laplacian fashion and leads to an increase in the permissible time step.

In the absence of grid structure, the computer implementation of the method encounters certain compexities, such as memory storage, indirect addressing (i.e., gather–scatter operations), and access of memory vectorization, which require particular handling. The geometrical data, as well as the usual $x, y, z$ coordinate arrays per node, include an element-node connectivity array consisting of four integers associated with each tetrahedral element. It also includes a boundary array described by three integers associated with the nodes of a triangular facet at the boundary and integers indicating the adjacent element and the boundary condition marker, such as the far field, the wall, etc. More storage space for geometrical quantities such as volume and partial derivatives of shape functions per element (appearing in the $K_{jk}$ terms) is required in order to avoid recomputation of these quantities at each time step. The overall memory requirements of the code add up to 100 words per node. The CRAY Y-MP CPU time spent per iteration and per node is a fraction of $10^{-5}$ sec. Usually three to five thousand iterations are required to attain a converged solution starting from the initial free-stream conditions; and at such an event the mean residual values would drop about three to four orders of magnitude. It has been found that for grids consisting of a large number of elements with substantial spatial variation of the element length, further iterations would decrease the residue only a little. This is due to integrated numerical errors introduced by the quality of the grid, or by the explicit nature of time integrations, or by the artificial viscosity effects, where parasitic numerical solutions or possibly false transients are not damped out properly. This problem, according to our knowledge of other 3-D explicit codes, is not unique to this code.

## Runge–Kutta Time-Marching Scheme

The Galerkin statement (eq. (7)) for a single forward explicit time step and with the consistent mass matrix $M_{jk}$ replaced by the lumped (diagonal) mass matrix $M^L_{jk}|_{k=j}$ can be written as

$$M^L_{jk}|_{k=j} \Delta U^n_j = \Delta t \; Q_j \qquad (23)$$

where $Q_j$ is the right side of equation (7) evaluated at $t^n$. The Runge–Kutta scheme implemented in FELISA has adopted an alternative procedure for discretization of $Q_j$. Here edge-based computations, rather than the usual standard finite-element procedure of looping over the individual elements and sending element contributions to each appropriate node, are considered. It is shown in reference 29 that the discretization of $Q_j$ can be interpreted by means of edge contributions to the pertinent nodes. The associated data structure would consist of the list of the nodes $j$ and $k$ for each side in the mesh. The memory storage requirement reduces to about 70 words per node, which is the most compact of the various possible alternatives.

It can also be shown that this data structure efficiently lends itself to the computation of the artificial viscosity in a manner similar to that used in the Taylor–Galerkin scheme. Equation (23) with inclusion of the artificial viscosity term denoted by $D_j(U)$ at a general node $j$ can be expressed as

$$\Delta U^n_j = \Delta t \; \left( M^L_{jk}|_{k=j} \right)^{-1} R_j \qquad (24)$$

where $R_j(U) = Q_j(U) + D_j(U)$. A multistage time discretization of equation (24) can be written as

$$
\begin{aligned}
U_j^{(0)} &= U_j^n \\
&\vdots \\
U_j^{(p)} &= U_j^{(p-1)} - \alpha_p \, \Delta t \\
&\quad \times \left( M_{jk}^L |_{k=j} \right)^{-1} R_j(U_j^{(p-1)}) \\
&\vdots \\
U_j^{n+1} &= U_j^{m-1}
\end{aligned}
\qquad (25)
$$

where $p = 1, \ldots, m-1$ and with the parameters $\alpha_p$ assigned appropriate values. A variable time step close to the stability limit is imposed by the Courant number, CFL, at each mesh point. The time-marching scheme in this code has only been considered for nontransient problems, and convergence to steady state is accelerated by local time stepping. Far-field boundary conditions, as in the Taylor–Galerkin scheme, are imposed pointwise at each step based on a linearized characteristic analysis in the direction normal to the boundary. Inviscid wall boundary conditions are imposed likewise in the Taylor–Galerkin scheme.

## RESULTS

Calculations were performed to assess the overall performance of the code FELISA. Several applications to geometrically simple 2- and 3-D problems ranging from transonic to supersonic are discussed. The 2-D test problems have been geometrically modeled as 3-D problems so that FELISA could be applied; the boundary conditions are chosen so that the 2-D flow field is simulated. The models considered are a NACA 0012 and a double-wedge profile. The 3-D problems consist of three wind tunnel models—a low-aspect-ratio wing, a cone cylinder, and a wing-body configuration at supersonic Mach numbers near 2.0—for which wind-tunnel-measured off-body pressure signatures exist. These models are generic configurations used for sonic boom prediction. Unless otherwise specified, most of the computational results discussed below are obtained by the Taylor–Galerkin scheme in FELISA.

## NACA 0012

The NACA 0012 transonic wing is one of the standard test cases frequently used for bench marking many CFD algorithms in 2-D. For the current application, a 3-D nonvarying-cross-section wing was generated from the 2-D NACA 0012 profile definition. The wing at both ends is mounted between two parallel planes in the $y$-coordinate (spanwise) direction. With the aid of reflection wall boundary conditions (symmetry type) at the mounting planes, a 2-D flow-field solution results. Several flow conditions and meshes have been considered. For free-stream Mach number, $M_\infty$, equal to 0.85 and 0 deg incidence, a 3-D tetrahedral mesh has been generated about the model which consists of a relatively dense grid in the region of the shocks. This mesh has been geometrically adapted to the region of expected shocks. For reasons discussed herein, the solution adaptivity programs in FELISA, remeshing or refinement techniques, failed to produce new grids for the several trial runs made. However, because of the geometrical simplicity of the shock regions, whose locations were easily predicted from an analysis of the solution obtained on a relatively uniform grid, adaptation of the grid was readily set by defining a slab region about the shock. The grid was clustered about the leading and trailing edges set by means of exponential distribution functions in the code. The 3-D mesh consists of 99,000 tetrahedral elements and 23,000 grid points.

Figure 1(a) shows an expanded view of the mesh on one of the planes with symmetry-type boundary condition. Figure 1(b) displays surface pressure coefficients, $C_p$, along the chord, and figure 1(c) shows a partial view of the computed Mach contours on the symmetry plane. Regions of supersonic flows in the upper and lower surfaces are fairly symmetric and the shock is captured relatively smoothly. Figure 1(d) shows a partial view of the Mach contour solution at $M_\infty = 0.85$ and 1 deg incidence, and figure 1(e) displays the corresponding surface pressure. Some slight oscillation is observed behind the shocks. Solutions have also been obtained at $M_\infty = 0.95$ and 0 deg incidence on the same mesh displayed in figure 1(a). Mach contour solutions are shown in figure 1(f): here we can see that the fishtail-like shock formed at the trailing edge is fairly spread out. This is an expected result of the coarseness of the grid, since the clustering of the grid points is not in the region where the shocks developed. Solution adaptive programs were tried again but failed to concentrate the grid in the critical regions because of program bugs, as before. A similar geometry adaptive grid approach was used to cluster the grid in the fishtail-like shock regions. The new grid consists of approxmately 118,000 elements and 27,000 points. Figure 1(g) shows an expanded view of this geometry adaptive grid on the symmetry plane. Figure 1(h) displays a partial view of the corresponding Mach contour solutions; here we can see that shocks are captured much more crisply than before. Predicted locations of shocks and surface pressure distributions for these ranges of Mach numbers and angles of attack agree closely with those predicted by the advanced CFD techniques that use dense structured grids about the NACA 0012 profile in 2-D simulation (ref. 30). Steady-state solutions were obtained within 3,000 time iterations for which the $L_2$ norm of the residue had dropped three orders of magnitude. Further iterations did not significantly change the residue.

## Double Wedge

The second 2-D test case used to assess the FELISA code was a supersonic flow past a 6-deg double-wedge profile at $M_\infty = 1.75$ and 0 deg incidence. A 3-D geometric model of this profile that simultates 2-D flow conditions was modeled in exactly the same fashion as for the NACA 0012 profile. The inital coarse grid consists of approximately 43,000 elements and 10,000 grid points. Figure 2(a) shows an expanded view of the initial mesh on the symmetry plane. Figure 2(b) displays a partial view of the computed Mach contours on the same plane, and figure 2(c) gives surface pressure plots along the chord. The characteristics of the flow include leading- and trailing-edge shocks and expansion waves at the wedge. Because of the coarseness of the grid, shocks are smeared out, and consequently the accuracy of the solution is rather poor in that area. Figure 2(c) shows a comparison of the computed surface pressure with the theoretical shock-expansion results; some oscillations in the solution are observed in regions of shocks and the expansion fan. The width of these regions is spread out over several grid cells. An enhancement of the solution can be achieved by the clustering of more grid points in the shock regions. This procedure is successfully accomplished by the FELISA remeshing, solution adaptive program, in two iterations. First, the solution obtained on the initial coarse grid is used in conjunction with the remeshing procedure; then the Mach number is selected as the "key" variable for the error indication or the driving mechanisim for grid redistribution. Figure 2(d) shows an expanded view of the first solution adaptive grid; here, one can see that the grid points are relatively concentrated in the shock regions, as expected. Using this new grid, FELISA was run again for the same flow conditions and a new solution was obtained. The new solution was again used in the same manner in conjunction with remeshing to cluster further grid points in the critical regions. Figure 2(e) shows the second solution adaptive grid on the symmetry plane. Here, the grid points are not only more dense in critical regions, but they trace the shock footprints more crisply than the first solution adaptive grid. This mesh consists of 114,000 elements and 26,000 grid points. Figure 2(f) shows the partial view of the Mach contours on the second solution adaptive grid. Shock waves are resolved more crisply and clearly (with less noise on the contour lines) compared with the solution pertinent to figure 2(b). Finally, figure 2(g) depicts predicted surface pressures obtained on the second solution adaptive grid in comparison with the theoretical solution. Results are noticeably enhanced in that the shock is less spread out. Wiggles at the leading and trailing edges due to shocks, and at the wedge due to expansion, although still present, are smaller than in the solution obtained on the initial grid. For all the applications in this example, the code was run for 4,000 time steps and the $L_2$ norm of the residue dropped four orders of magnitude.

## SONIC BOOM EXAMPLES

The problems discussed here will show the extent of the applicability of FELISA to the sonic boom examples listed. Off-body pressure signatures at certain distances from the model are extracted from the computational results and in some cases have been extrapolated to distances farther away from the body for comparison with the pertinent experimental data. The extrapolation algorithm is based on the Witham theoretical method known as the F function, developed for $N$-wave-like propagation. For detailed information on the geometry and for alternative CFD/experimental correlations related to off-body pressure signatures of these models, see reference 31.

### Low-Aspect-Ratio Wing

The first sonic boom model is a 0.5-aspect-ratio rectangular wing with parabolic sections. Computed solutions are obtained at $M_\infty = 2.01$ and 0 deg angle of attack. Because of geometrical symmetries, only one quadrant of the flow field has been discretized. Flow features consist of leading- and trailing-edge shocks and expansion waves generated along the convex surface of the wing. Figures 3(a) and 3(b) display a sectional view of the nonadapted initial surface grid on the $xy$ and $xz$ planes of symmetry and a quadrant of the wing surface. The grid consists of 245,000 elements and 44,000 points. Figure 3(c) shows a sectional view of Mach contours on the surfaces described above. The computed pressure signatures, $\Delta p/p_\infty$, $\Delta p = p - p_\infty$, along a line on the $xz$ symmetry plane one chordlength above the wing (visible in fig. 3(a)) are shown in figure 3(d) compared to corresponding wind tunnel results. Signatures are plotted against $x/c$, where $x = 0$ is at the wing leading edge and $c$ denotes the chord. It can be seen that the computed solution overestimates the pressure rise caused by both the leading-edge shock and the pressure fall from expansion. The pressure rise resulting from the trailing-edge shock is better. Discrepancies are attributed to the coarseness of the initial grid in the critical regions.

Remeshing procedures are used to enhance several solutions, using solution adaptive programs. Two of these are discussed here. Mach number has been used as a "key" variable for grid redistribution. Figures 3(e) and 3(f) show a sectional view of the solution adaptive grids on the symmetry planes and an expanded view of the wing. Both figures are enmeshed in a denser grid, although the volume grid here globally consists of fewer grid points—95,000 elements and 19,000 points. Grid clustering follows the oblique shocks attached to the leading and trailing edges. Figure 3(g) shows Mach contour solutions for the same surfaces; in comparison with figure 3(c), shocks appear more crisp. Computed and experimental pressure signatures shown in figure 3(h) are also reasonably improved and are in fair agreement. Here, the leading-edge shock is slightly underestimated as a result of the degree and quality of grid concentrations.

Another solution adaptive grid application that has been successfully pursued is shown in figures 3(i)–3(k). Figure 3(i) shows a solution adaptive grid that traces footprints of leading- and trailing-edge shocks in a more pronounced fashion. Mach contour solutions in figure 3(k) are more crisp than those in figure 3(g). The volume grid here consists of more grid points: 171,000 elements and 34,000 points. Again, as depicted in figure 3(l), computed and experimental solutions are in reasonable agreement. The pressure rise as a result of the trailing-edge shocks is improved compared to that of the previous solution adaptive case, and the solution is generally smoother. However, the pressure rise resulting from the leading-edge shock and the following expansions are slightly underestimated. Discrepancies in the results are probably due to density adjustments or perhaps from some skewness of the pertinent 3-D grids. All of the solutions are obtained within 4,000 to 6,000 time iterations.

Several other solution adaptive results were compiled with similar outcomes, and several attempts failed completion for some selected parameters in the code. Our experiences here with solution adaptivity on unstructured grids, and on structured (ref. 8) grid approaches for 3-D problems, indicate that the adaptivity does play an important role in the considerable enhancement of the solution. However, the main issues here are not merely grid clustering but the initial capability to detect the critical regions of the solution properly, and at the same time generate nonskewed grids that are dense enough to progressively enhance the solution. Tuning several code parameters to achieve these goals without failure of the program is currently a formidable task. Further rather fundamental work must be done in order to alleviate the present ad hoc guessing procedures for setting code parameters. Our results were affected by these types of limitations.

## Cone Cylinder

The second sonic boom model is a sharp cone (conical spike) with a cylindrical attachment at its base representing the sting. The cone surface is linear and its half-angle is 3.24 deg. A quadrant of the model has been considered for discretization of the flow field. Because of the small cone angle, this model has been a particularly challenging geometry for the grid generator. Surface grid data near the cone apex were regenerated so that the aspect ratio of the $(m - 1)$ by $(n - 1)$ quadrilateral patches was maintained at about five to eight, and so that neighboring patch areas were smoothly changed on departure from the apex. The flow field solution is obtained at 0 deg angle of attack for $M_\infty = 1.68$. The main feature of the flow field is an attached weak bow shock at the cone apex and an expansion wave at the cone and cylinder section; the geometry-adaptive grid method is used to concentrate the grid points in the critical regions. Because of the simplicity of the problem here, the conical shock inclination may be approximated by the Mach angle

$\mu$ pertinent to the Mach wave characteristic. This approximation is then used to generate a set of lines lying on a conical surface with half-angle $\mu$ that can be input in the background grid file for concentrating the grid points in the critical regions at the apex and the cone cylinder.

Figures 4(a) and 4(b) display the sectional view of the geometry adaptive grid on the $xy$ and $xz$ symmetry planes and the model surface. The grid consists of 320,000 elements and 61,000 points. Solutions have been obtained for 5,000 time iteratons where the residue has dropped three orders of magnitude and a steady state is achieved. Mach contours are shown in figure 4(c). The pressure signature data are obtained half a body length away from the surface (visible in fig. 4(a)) in the free-stream flow direction. These data are then extrapolated to the wind tunnel measurement distance of ten cone lengths off-body. The comparison between predicted and measured results is shown in figure 4(d). (It should be noted that in figs. 4–9, $x = 0$ does not correspond to $x = 0$ of the body coordinate system). This figure also contains a multiblock grid solution by the structured grid solver TEAM (ref. 31). Results appear to be in agreement near the shock but differ slightly near the expansion wave. The inconsistency is believed to be due to the coarseness of the grid to the right of the expansion wave.

## Delta Wing-Body

The third model is a slender delta wing-body with a delta-wing planform with a leading-edge sweep of 69 deg and a diamond-shaped double-wedge airfoil. Grid generation about this model, which has a slender forebody with small vertex cone angle and a body–sting connection, has also been a challenging task. Because of symmetry at the constant $y = 0$ coordinate plane, the flow field about only half of the model is considered. Primary flow features consist of shocks and expansions of the forebody at leading and trailing edges and at the base of the body–sting regions. This model has been used in conjunction with our numerical experiments for several purposes: for comparison of flow algorithms in FELISA, to test solution adaptivity, to study the geometrical effect of the body–sting step- and ramp-function connections, and, finally, for computations at nonzero angles of attack at different Mach numbers.

An initial relatively coarse grid is generated which consists of 423,000 elements and 79,000 points. In the results that follow, our intention has been to focus our effort on the flow features developed in regions between the body nose and the base. The sting extends back about two body lengths and has been truncated as a conical surface. No attempt has been made to resolve shocks at the artificial end of the sting. An expanded view of the symmetry-plane triangulation and an isometric view of the surface triangulation of the model are shown in figures 5(a) and 5(b). Two flow solvers in FELISA, the Taylor–Galerkin and Runge–Kutta schemes, are used and computed solutions by each of the solvers at 0 deg angle of attack for $M_\infty = 1.68$ are compared with the wind tunnel data. The near-field pressure signature data are obtained on a line away from the body parallel to the body axis of symmetry ($x$ axis), and on the symmetry plane with altitude ratio $h/l = 0.3$. Here, $h$ is the distance of the line from the $x$ axis and $l$ is the reference body length. The $h/l = 0.3$ solution is extrapolated (ref. 31) to the experimental measurement distance of $h/l = 3.6$. Extracting off-body pressure signatures from the unstructured solution data is accomplished by a simple interpolation procedure. A data acquisition line of finite length, with $n$ sample points uniformly distributed between its two ends, is defined. The numerical values at each sample point are computed by averaging the solution data on the closest grid points; the results are plotted in figures 5(c) and 5(d). The computed results obtained by the Taylor–Galerkin scheme (fig. 5(d)) overall are in better agreement with the experimental data than are those obtained by the Runge–Kutta scheme (fig. 5(c)). The Runge–Kutta solver underestimates the pressure rise as a result of shocks at the apex and leading and trailing edges; similarly, expansions are underestimated. The noticeable disagreement between computed and experimental data toward the end of the signature plots for both methods is thought to be caused by the flow circulation at the base of the body where the backward-facing step connection between the body and the sting is a factor. The discrepancy becomes noticeably reduced when a 12-deg ramp body/sting connection is used (see also ref. 31). Furthermore, the Runge–Kutta solver appears to be more dissipative; shocks and expansions are more spread out than with the Taylor–Galerkin scheme. Some nonsmoothness of the plot lines is related to the interpolation procedure.

To enhance the solution, solution adaptive programs were applied. The remeshing technique was tried, but it failed as a result of program bugs, as before. The refinement technique with local Mach number chosen as the key variable could not properly detect and concentrate the grid at critical regions of shocks and expansions. Tuning the tolerance parameters to refine more tetrahedra would result in almost global refinement of the mesh and yield unacceptably large mesh sizes. This approach not only would exceed the practical limits of our resources but also would not serve the purpose of the concept of solution adaptivity.

The shortcoming of the adaptivity procedure was overcome by the geometry-adaptive grid program. A conical region about the apex and a slab region about the leading and trailing edges were specified, and all tetrahedral elements falling in these regions were then refined. Figures 5(e) and 5(f) display an isometric view of the adapted grid on the symmetry plane and the model, and the surface of the model alone. The grid consists of 930,000 elements and 169,000 points. Figures 5(g) and 5(h) display pressure signature results obtained by the Runge–Kutta and Taylor–Galerkin schemes, comparing them with the experimental data. The Runge–Kutta solution does not show an improvement compared with the previous solution (fig. 5(c)). A possible explanation here is insufficient grid refinement with the Runge–Kutta algorithm. The Taylor–Galerkin solution, on the other hand, has slightly improved prediction signatures caused by expansion waves over the wing.

Several other combined applications of the refinement technique with the geometry-adaptive grid program have been tried. A grid consisting of 1,150,000 elements and 169,000 points has been generated. Figures 5(i) and 5(j) show isometric views of the grids and figure 5(k) shows the comparison of the pressure signatures. Here we notice an improvement with the Runge–Kutta solver; nevertheless, the dissipative nature of this solver still persists. It should be noticed that the dissipative effects experienced with this solver of FELISA by no means should indicate similar behavior for other solvers based on the Runge–Kutta time integrations.

To study the effects of the geometry at the body–sting connection and for angles of attack larger than zero, the geometry-adaptive grid program was used to generate more efficient grids. All the solutions below are obtained by the Taylor–Galerkin solver. In one solution, a grid consisting of 780,000 elements and 142,000 points was generated for the rearward-facing step-type connection geometry (as in the previous model). The second solution, using a grid of almost the same size, was generated for a 12-deg ramp-type geometry. This grid consists of 796,000 elements and 146,000 points. Figure 6(a) displays grid adaptation for expected features of the solution in the symmetry $xz$ plane, and figure 6(b) displays an isometric view of the model on the symmetry plane.

The near-field pressure signature data are obtained on the same line as before, i.e., with $h/l = 0.3$, and extrapolated to $h/l = 3.6$. The comparison of results between prediction and measurement is shown in figures 7(a) and 7(b) for step- and ramp-shape geometries, respectively. The bow shocks and expansion signatures at the forebody and the expansions at the wing wedge region are in close agreement, although the wing leading- and trailing-edge shocks are slightly underestimated. The difference is attributed to grid clustering, in which a slightly denser grid is required. It should also be noted that numerical dissipative terms in the Euler flow solvers, which are automatically accounting for the wake flow behavior at the trailing edge, are another source of discrepancies between the computed and measured data.

The solution for the ramp shape plot, figure 7(b), shows noticeable improvement at the body–sting base when compared with the step geometry results. It appears that the ramp geometry introduces a pseudo-boundary to account for flow recirculation in the case of rearward step geometry. The choice of this ramp angle was first suggested by Whitham (ref. 32), who provided a detailed discussion of this matter. The presence of a slight inconsistency of the computed body–sting base expansions and recompression shocks and the experimental data is still thought to be related to grid effects and flow behavior at the base, which did not quite disappear as a result of the choice of the ramp geometry function.

The ramp-shape body/base intersection model, consisting of 796,000 elements, was used to obtain solutions at $M_\infty = 1.68$ and angles of attack of 2.53 deg and 4.74 deg corresponding to lift cofficients of 0.08 and 0.15, respectively. The CFD procedure for $\alpha > 0$ calculations here does not rotate the grid but does rotate the free-stream velocity vector. The CFD data acquisition line starts at $h/l = 0.3$, positioned below the body nose, and is parallel to the free-stream direction; $h/l$ at the other end of the acquisition line becomes smaller as $\alpha$ gets larger. Computed results are extrapolated to $h/l = 3.6$ along a line also parallel to the free-stream direction and are generally in close agreement. Results are displayed in figures 8(a) and 8(b). The strengths of the leading- and trailing-edge shocks in both figures are slightly over- and underestimated, respectively. It should be noted, particularly for additional results to be discussed, that the adaptive grid used here was made for $M_\infty = 1.68$ and $\alpha = 0$ deg. It should not be expected to be the best grid distribution for obtaining the solution at different Mach numbers and at angle incidences larger than zero.

Finally, our last computational results are a comparison of solutions obtained at $M_\infty = 2.7$ and $\alpha = 0$ deg, 3.47 deg, and 6.52 deg. Figures 9(a)–9(c) display these results. Computed solutions are obtained by interpolation, as before, on the data acquisition line of $h/l = 0.3$ at the nose of the model, extrapolated to $h/l = 3.1$. The grid is the same adaptive grid consisting of 796,000 elements with ramp-shape body/sting intersection. Consequently, the computed results show loss of accuracy for larger angles of attack. For $\alpha = 0$ deg, the computed solution agrees satisfactorily with experimental data; the leading-edge shock is not adequately predicted, but the bow and tail shocks are captured more sharply than by experiment. For $\alpha = 3.47$ deg and 6.52 deg (figs. 6(b) and 7(c), respectively) we observe interaction between the bow and wing shocks at the leading edge. In the latter case, shocks have nearly coalesced. The overall strength of the shocks predicted are in close agreement with experiment, although discrepancy resulting from interaction is particularly pronounced for the case $\alpha = 3.47$ deg. In addition, the trailing-edge shock and signatures at the base/sting intersections are not well resolved.

The following comments should be considered in relation to these results. (1) The grid used for the above calculations, as mentioned earlier, was generated for $\alpha = 0$ deg and $M_\infty = 1.68$ and would not be expected to be entirely appropriate for $M_\infty = 2.7$ calculations. The Mach wave angle for the larger Mach number is about 21 deg compared to 36 deg for the smaller Mach number. The difference in Mach wave angles indicates that the direction of concentrations of grid points for the 36-deg case is not suitable for the 21-deg case. (The higher the Mach number, the higher the propagation speed of disturbances and the higher the strength of shocks; grid clustering, then, must be properly done, particularly for larger angles of attack.) (2) The dissipation term used in the code may be unsuitable for shock interaction cases and/or wake computations. (3) The case with the higher Mach number would be more sensitive to the body/sting intersection model and the actual geometry of the sting itself. This could possibly be responsible for the signature discrepancy at the downstream tail shock. (4) Finally, it should be noticed, as discussed earlier, that for nonzero angles of attack, the extraction of data from the computed solution along the data-acquisition line for interpolation has less altitude ratio at the downstream end of the line. For instance, for the case $\alpha = 6.52$ deg, $h/l \approx 0.03$ about two body lengths away from the nose compared to $h/l \approx 0.3$ at the nose of the model. The difference in $h/l$ ratio suggests that some 3-D phenomena are not yet fully developed, and this would explain the poor correlation with experiment downstream of the tail shock. In addition, the experimental data acquisition per se may be for a different angle of attack and may be more sensitive in cases of higher Mach numbers. Experimental data need more resolution for such regions with bow and wing shock interactions.

## CONCLUSIONS

An assessment of an unstructured-grid, finite-element-based explicit code, FELISA, has been described. The code can generate spatially smooth, varying tetrahedral grids for complex geometries. The preparation of the geometrical data exchange, depending on the complexity of the model, could be quite time consuming; however, because of the coherent definition used in representing the geometry, the procedure could be easily automated. Lack of well tuned control parameters could generate skewed elements on or away from the surface; the program may frequently abort without provisioned recovery capability. The code, however, has the potential to incorporate local treatments to alleviate this problem.

The flow solver based on the Taylor–Galerkin method has produced reliable results for steady state, inviscid compressible flow problems, and has been tested on problems ranging from transonic to supersonic. The solution-adaptive programs need to be improved and the mechanism for detecting critical regions of the domain must be modified. Remeshing is a means of optimizing the adaptation procedures where the density and total number of elements can come under control. The rate of success with the remeshing program might be greatly improved if local alleviation of the gridding problems is incorporated. The refinement approach is simple to use but should be accompanied by the pertinent unrefinement program to avoid excessive increase in the total number of elements.

Despite existing problems with the adaptive programs, innovative users can find ways of tailoring them to apply to a particular problem using geometry-adaptive techniques. The original developers of the code have demonstrated qualitatively impressive solutions for complex geometries, and, in some cases, their predicted results have been favorably compared with experimental data. The overall results we have obtained with this code have been reliable within the scope of the problems reported here and for some which have not been included. Further tests on more geometrically complex configurations are neccessary to establish the practicality of the code and the reliability of the predicted solutions.

## REFERENCES

1. Lomax, H.: CFD in 1980s from One Point of View. AIAA Paper 91-1526. AIAA 10th Computational Fluid Dynamics Conference, June 1991.
2. Numerical Grid Generation in Computational Fluid Dynamics and Related Fields. A. S. Arcilla, J. Hauser, P. R. Eiseman, and J. F. Thompson, eds., North-Holland, 1991.
3. Steinbrenner, J. P.; and Chawner, J. R.: Enhancements to the GRIDGEN System for Increased User Efficiency and Grid Quality. AIAA Paper 92-0662, Jan. 1992.
4. Shaw, J. A.; Georgala, J. M.; Peace, A. J.; and Childs, P. N.: The Construction, Application and Interpretation of Three-Dimensional Hybrid Meshes. Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, A. S. Arcilla, J. Hauser, P. R. Eiseman, and J. F. Thompson, eds., North-Holland, 1991, pp. 887–898.
5. Hwang, C. J.; and Liu, J. L.: Locally Implicit Hybrid Algorithm for Steady and Unsteady Viscous Flows. AIAA J., vol. 30, no. 5, May 1992, pp. 1228–1236.
6. Nakahashi, K.; Nozaki, O.; Kikuchi, K.; and Tamura, A.: Navier–Stokes Computations of Two- and Three-Dimensional Cascade Flowfields. J. Populsion and Power, vol. 5, no. 3, 1989, pp. 320–326.
7. Tavella, D. A.; Djomehri, M. J.; Kislitzin, K. T.; Blake, M. W.; and Erickson, L. L.: A Concurrent Hybrid Navier–Stokes/Euler Approach to Fluid Dynamic Computations. AIAA Paper 93-0789, Reno, Nev., Jan. 11–14, 1993.
8. Djomehri, M. J.; and Deiwert, G. S.: Solution-Adaptive Program SADAP3D. AIAA Paper 91-2903, Aug. 1991.
9. Peraire, J.; Vahdati, M.; Morgan, K.; and Zienkiewicz, O. C.: Adaptive Remeshing for Compressible Flow Computations. J. Comp. Phys., vol. 72, no. 2, Oct. 1987, pp. 449–466.
10. Gumbert, C.; Lohner, R.; Parikh, P.; and Pirzadeh, S.: A Package for Unstructured Grid Generation and Finite-Element Flow Solvers. AIAA Paper 89-2175, July 1989.
11. Baker, T. J.: Three Dimensional Mesh Generation by Triangulation of Arbitrary Point Sets. Proc. of the AIAA 8th Comp. Fluid Dyn. Conf., AIAA Paper 87-1124, June 1987.
12. Mavriplis, D. J.: Adaptive Mesh Generation for Viscous Flows Using Delauny Triangulation. J. Comp. Phys., vol. 90, no. 2, Oct. 1987, pp. 271–291.
13. Yee, H. C.: A Class of High-Resolution Explicit and Implicit Shock-Capturing Methods. NASA TM-101088, Feb. 1989.
14. Barth,T. J.; and Frederickson, P. O.: Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction. AIAA Paper 90-0013, Jan. 1990.
15. Szema, K. Y.; Chakravarthy, S. R.; Peppi, K.; Rowell, C. M.; Burman, R. G.; and Schultz, R. E.: Application of a UNIVERSE-Series Code for Inviscid Flow Over Complex 3-Dimensional Configurations. AIAA Paper 92-0150, Jan. 1992.
16. Demkowicz, J. T.; Oden, J. T.; Rachowicz, W.; and Hardy, O.: Toward a Universal h-p Adaptivity Finite Element Strategy, Part 1, Constrained Approximation and Data Structure. Comp. Methods in Appl. Mech. Eng., vol. 77, no. 1/2, 1989, pp. 79–112.

17. Peraire, J.; Peiro, J.; Formaggia, L.; Morgan, K.; and Zienkiewicz, O. C.: Finite-Element Euler Computations in Three Dimensions. Int. J. Num. Methods Eng., vol. 26, no. 10, 1988, pp. 2135–2159.

18. Pirzadeh, S.: Structured Background Grids for Generation of Unstructured Grids by Advancing Front Method. AIAA Paper 91-3233, Sept. 1991.

19. Lohner, R.; and Baum, J. D.: Numerical Simulation of Shock Interaction with Complex Geometry Three-Dimensional Structures Using a New Adaptive H-Refinement Scheme on Unstructured Grids. AIAA Paper 90-0700, Jan. 1990.

20. Morgan, K.; Peraire, J.; Thareja, R. R.; and Stewart, J. R.: An Adaptive Finite-Element Scheme for the Euler and Navier-Stokes Equations. AIAA Paper, 87-1172, 8th Computational Fluid Dynamics Conf., Honolulu, Hawaii, June 1987.

21. Chalot, F.; John, Z.; Mallet, M.; Ravachol, M.; and Roge, G.: Development of a Finite-Element Navier-Stokes Solver with Applications to Turbulent and Hypersonic Flows. AIAA Paper 92-0670, Jan. 1992.

22. Lohner, R.; Morgan, K.; Peraire, J.; and Zienkiewicz, O. C.: Finite Element Methods for High-Speed Flows. AIAA Paper 85-1531, July 1985.

23. Jameson, A.; Baker, T. J.; and Weatherill, N. P.: Calculation of Inviscid Transonic Flow over a Complete Aircraft. AIAA Paper 86-0103, 1986.

24. Frink, N. T.; Parikh, P.; and Pirzadeh, S.: A Fast Upwind Solver for the Euler Equations on Three-Dimensional Unstructured Meshes. AIAA Paper 91-0102, Jan. 1991.

25. Donea, J.: A Taylor-Galerkin Method for Convective Transport Problems. Int. J. Numer. Methods Eng., vol. 20, 1984, pp. 101–119.

26. Hirsch, C.: Numerical Computation of Internal and External Flows, Vol. 1: Fundamentals of Numerical Discretization, and Vol. 2: Computational Methods for Inviscid and Viscous Flow Models, John Wiley & Sons, 1989, 1990.

27. Morgan, K.; and Peraire, J.; Finite-Element Methods for Compressible Flows. Von Karman Institute for Fluid Dynamics, Lecture Series 1987-04, 1987. Computational Fluid Dynamics, March 2–6, 1987.

28. Chakravarthy, S. R.; and Ota, D. K.: Numerical Issues in Computing Inviscid Supersonic Flow over Conical Delta Wings. AIAA Paper 86-0440, Jan. 1986.

29. Peraire, J.; Peiro, J.; and Morgan, K.: A 3D Finite-Element Multigrid Solver for the Euler Equations. AIAA Paper 92-0449, Jan. 1992.

30. Pulliam, T. H.; and Barton, J. T.: Euler Computations of AGARD Working Group 07 Airfoil Test Cases. AIAA Paper 85-0018, Jan. 1985.

31. Cliff, S .E.; and Thomas, S. D.: Euler/Experiment Correlations of Sonic Boom Pressure Signatures. AIAA Paper 91-3276, Baltimore, Md., Sept. 23–25, 1991.

32. Whitham, G. B.: The Flow Pattern of a Supersonic Projectile. Commun. Pure Appl. Math., vol. V, no. 3, Aug. 1952, pp. 301–348.

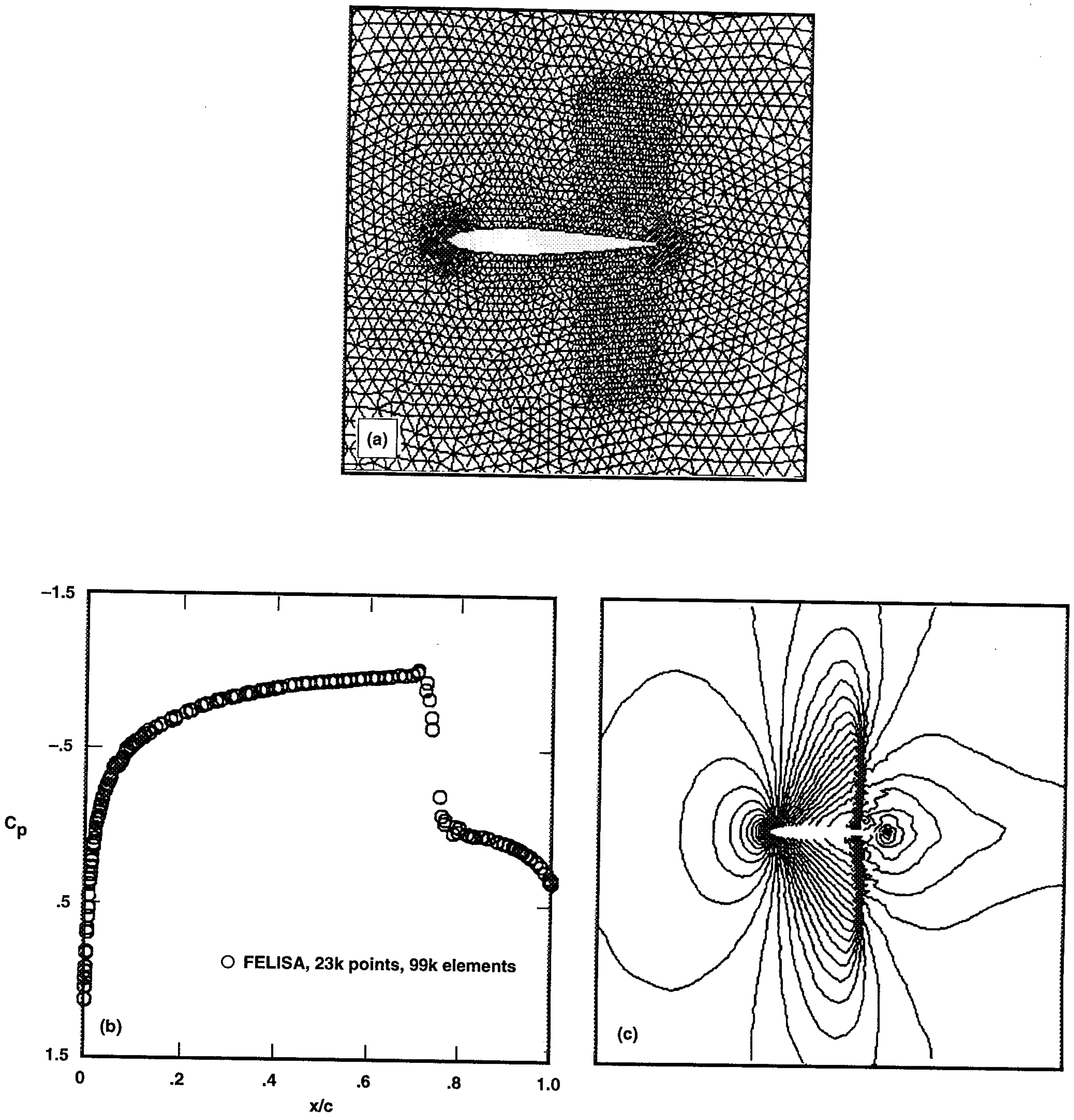

Figure 1. NACA 0012 unstructured grid—99,000 elements and 23,000 points, Mach contour solution, and surface pressure distributions. (a) Expanded view of grid on symmetry plane. (b) Surface pressure along chord. (c) Partial view of Mach contours at $M_\infty = 0.85$ and $\alpha = 0$ deg.

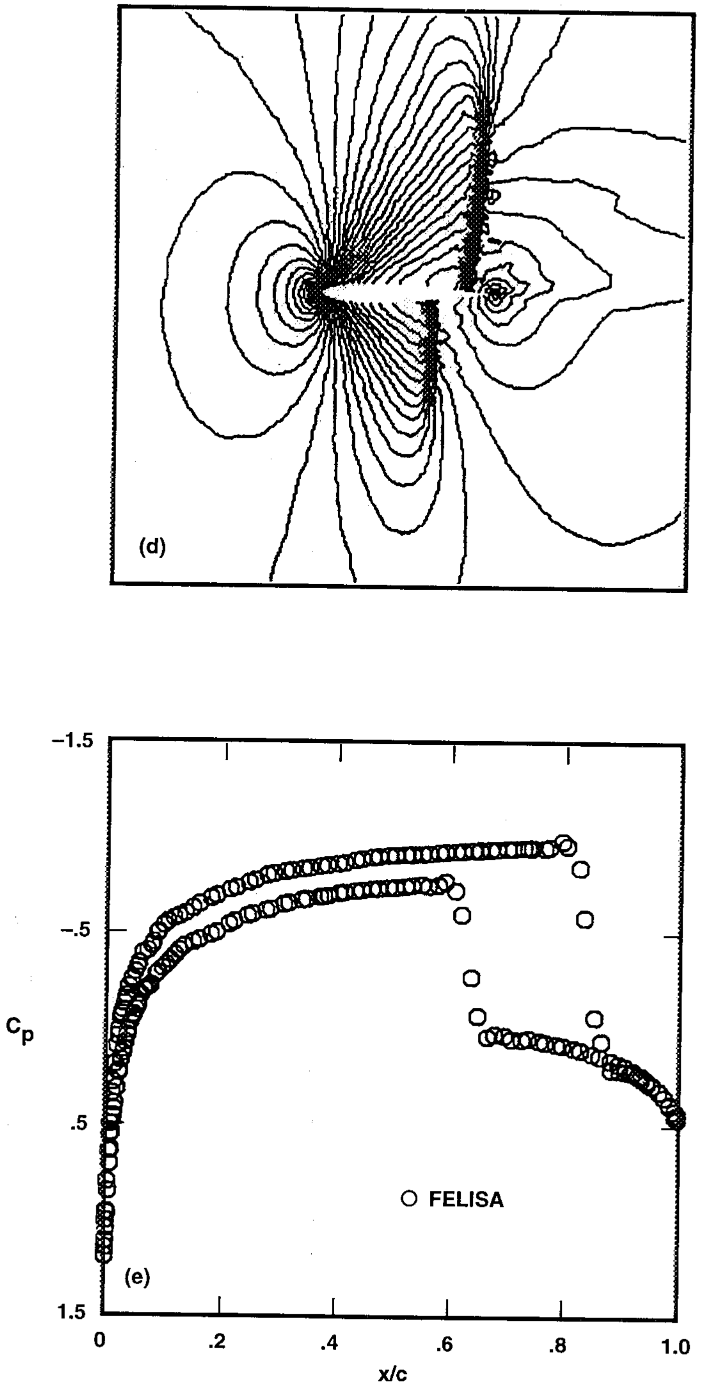

Figure 1. Continued. (d) Partial view of Mach contours at $M_\infty = 0.85$ and $\alpha = 1$ deg. (e) Surface pressure distribution.

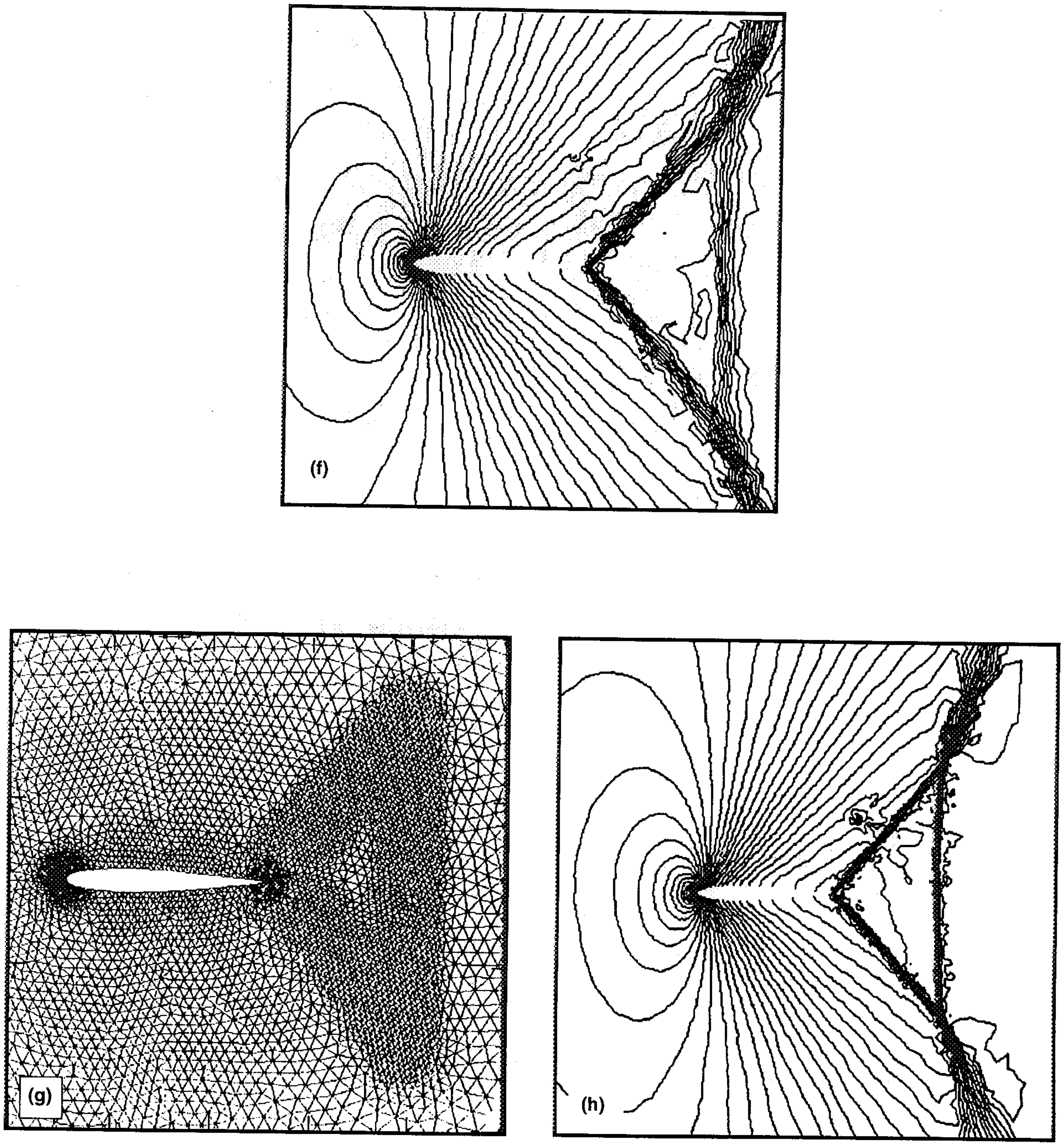

Figure 1. Concluded. (f) Partial view of Mach contours at $M_\infty = 0.95$ and $\alpha$ = 0 deg obtained on the grid of figure 1(a). (g) Expanded view of geometry adaptive grid on symmetry plane. (h) Partial view of Mach contours obtained on the above adapted grid.

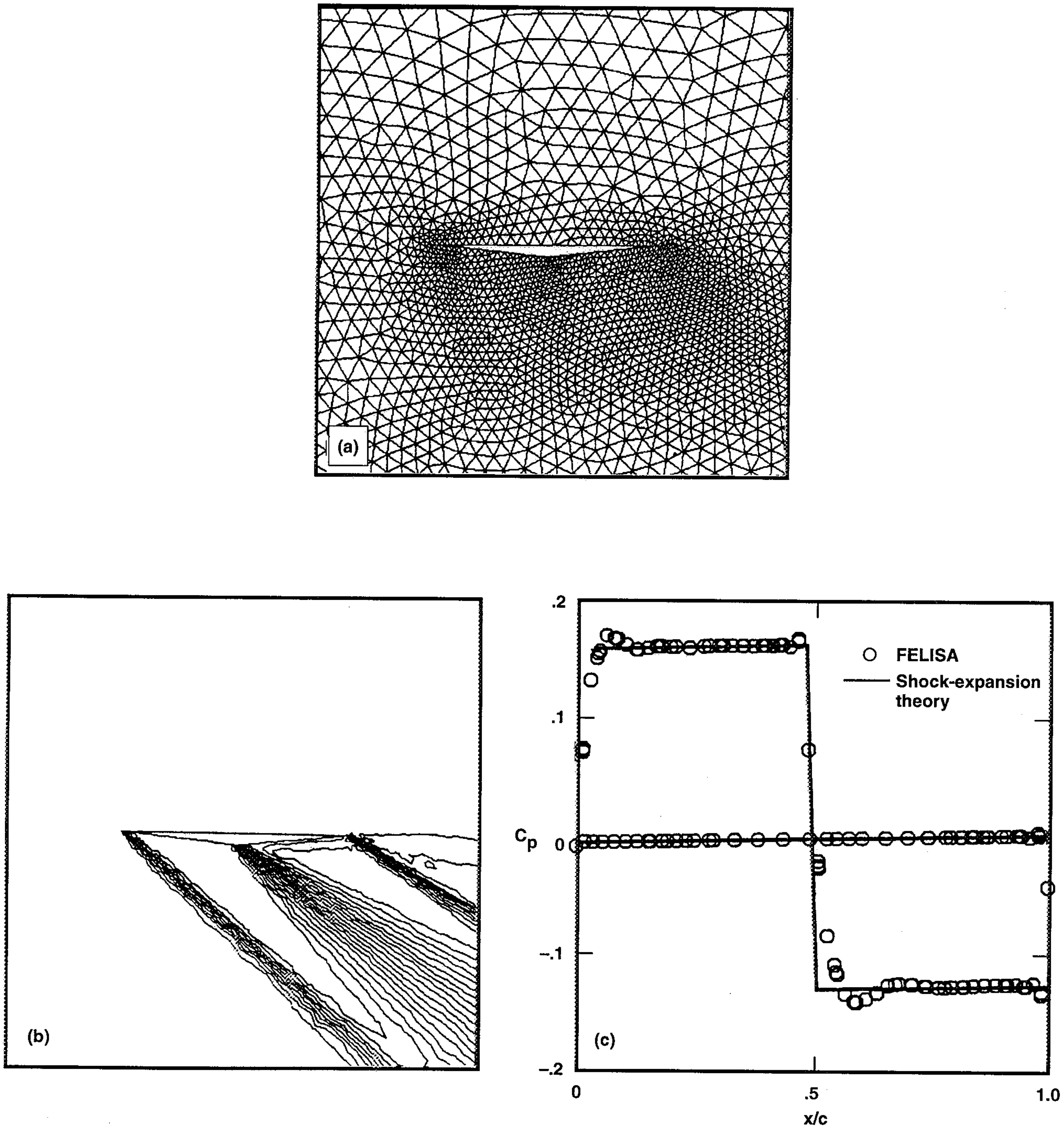

Figure 2. Double-wedge unstructured grids—43,000 elements and 10,000 points, Mach contours, surface pressure coefficients, and solution adaptive grid at $M_\infty = 1.75$ and $\alpha = 0$ deg. (a) Expanded view of grid on symmetry plane. (b) Partial view of Mach contours. (c) Comparison of computed and theoretical surface pressures along upper and lower surfaces.

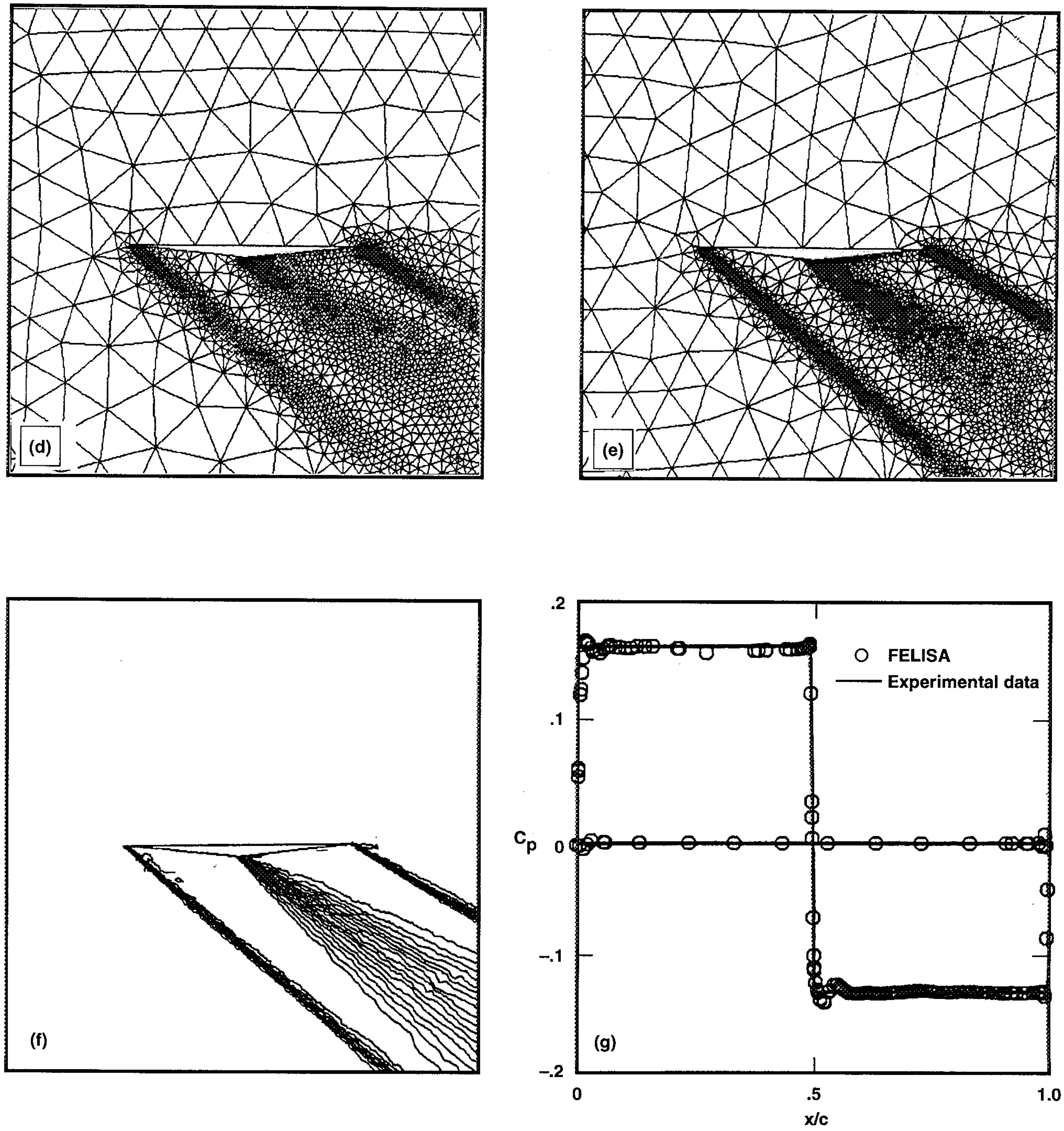

Figure 2. Concluded. (d) Expanded view of the first solution adaptive grid. (e) Expanded view of the second solution adaptive grid—114,000 elements and 26,000 points. (f) Partial view of Mach contours on the second solution adaptive grid. (g) Comparison of computed surface pressures on the second solution adaptive grid with theoretical results.

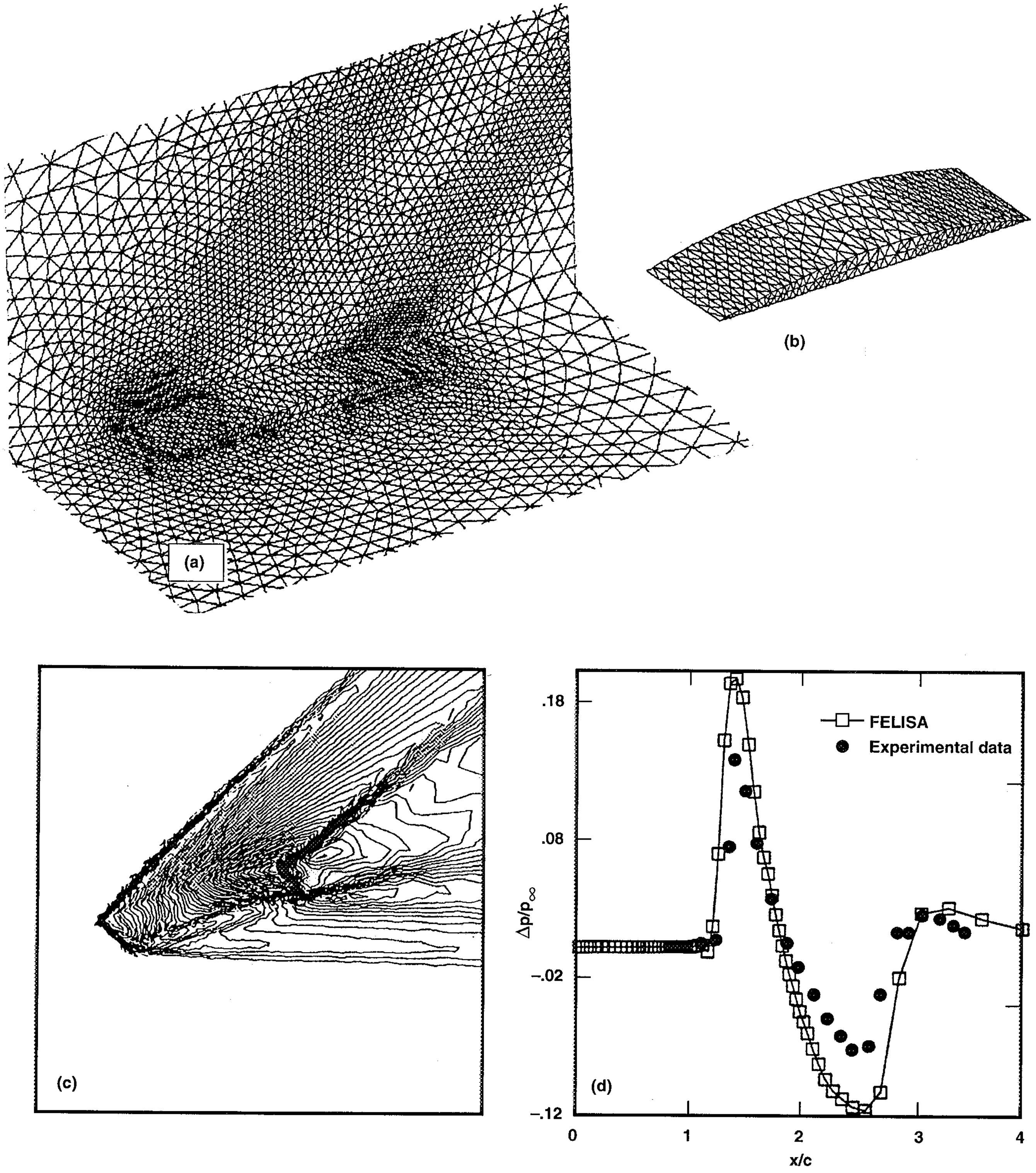

Figure 3. Low-aspect-ratio wing, Mach contours, pressure signatures, and solution adaptive grids, at $M_\infty = 2.01$ and $\alpha$ = 0 deg. (a) Sectional view of initial grid—245,000 elements and 44,000 points, on symmetry planes and wing surface. (b) Expanded view of one quadrant wing. (c) Sectional view of Mach contours. (d) Comparison of computed and experimental off-wing pressure signatures.

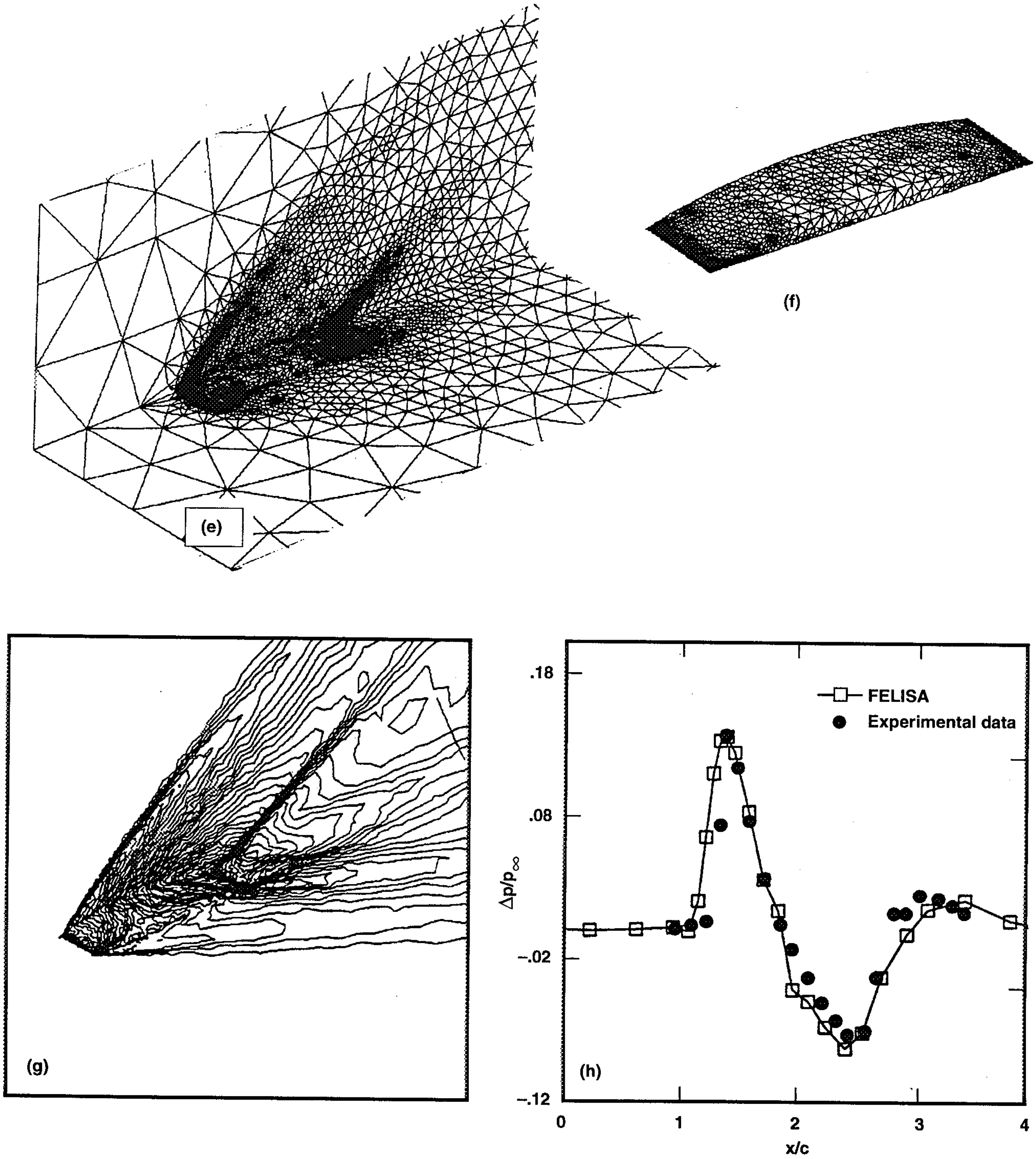

Figure 3. Continued. First solution adaptive computations. (e) Sectional view of solution adaptive grid—95,000 elements and 19,000 points, on symmetry planes and wing surface. (f) Expanded view of one quadrant wing. (g) Sectional view of Mach contours. (h) Comparison of computed and experimental off-wing pressure signatures.

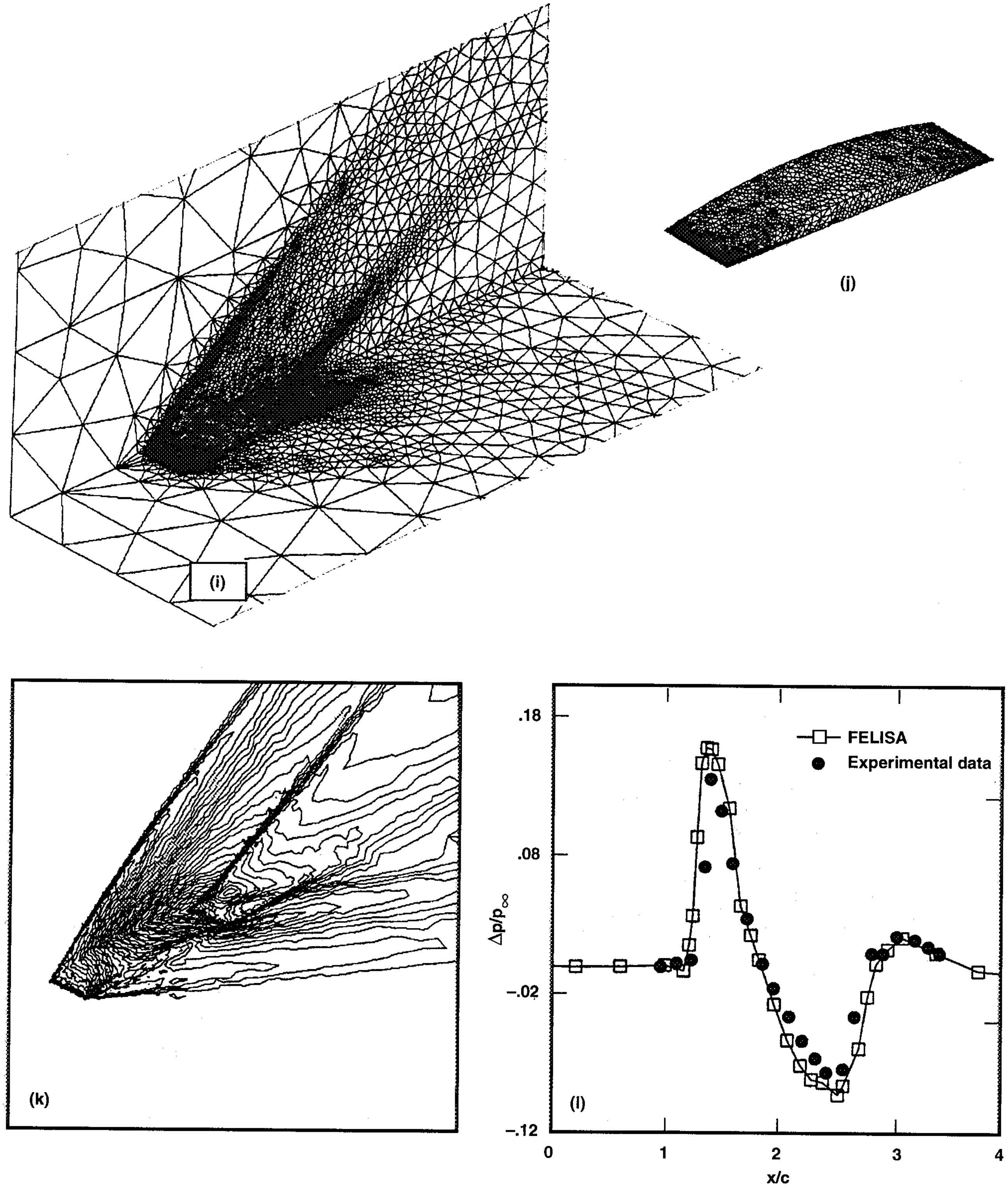

Figure 3. Concluded. Second solution adaptive computations. (i) Sectional view of solution adaptive grid—171,000 elements and 34,000 points, on symmetry planes and wing surface. (j) Expanded view of one quadrant wing. (k) Sectional view of Mach contours. (l) Comparison of computed and experimental off-wing pressure signatures.

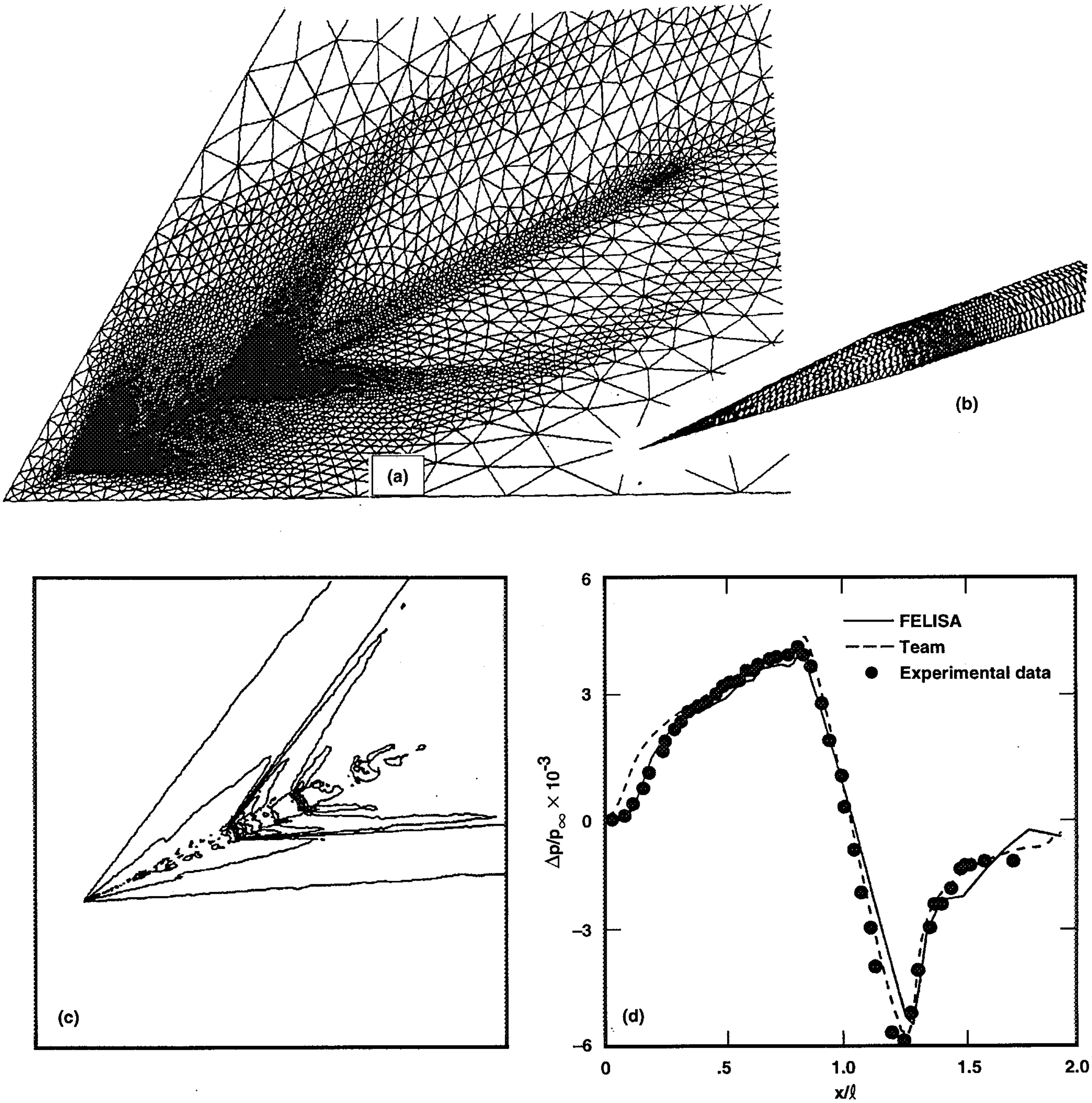

Figure 4. Cone cylinder, isometric view of grids, Mach contours, pressure signatures interpolated at $h/l = 0.4$ from computed solution data and extrapolated to $h/l = 10.0$, with $M_\infty = 1.68$ and $\alpha = 0$ deg. (a) Sectional view of geometry adaptive grid—319,000 elements and 61,000 points, on symmetry planes and model surface. (b) Expanded view of one quadrant model. (c) Sectional view of Mach contours. (d) Comparison of off-body pressure signatures computed by FELISA and TEAM with experiment.

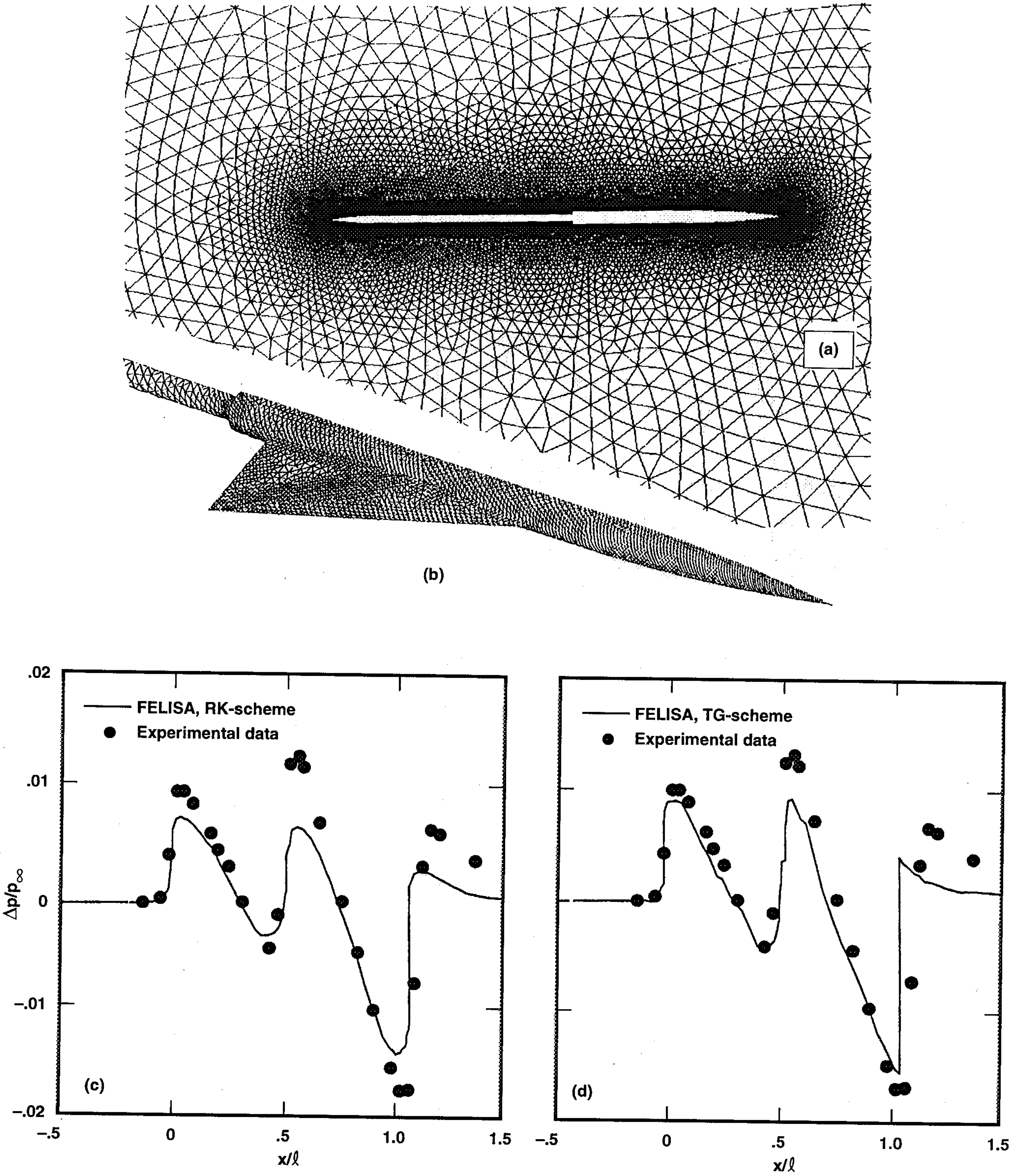

Figure 5. Delta wing-body sting, isometric view of grids, comparison of computed and experimental off-body pressure signatures for two solvers in FELISA, with $M_\infty = 1.68$ and $\alpha = 0$ deg. Pressure signatures interpolated at $h/l = 0.3$ from computed solution data and extrapolated to $h/l = 3.6$. (a) Nonadapted initial grid—423,000 elements and 79,000 points, on the symmetry plane. (b) Triangulations on the model surface. (c) Pressure signatures obtained by Runge–Kutta scheme. (d) Pressure signatures obtained by Taylor–Galerkin scheme.

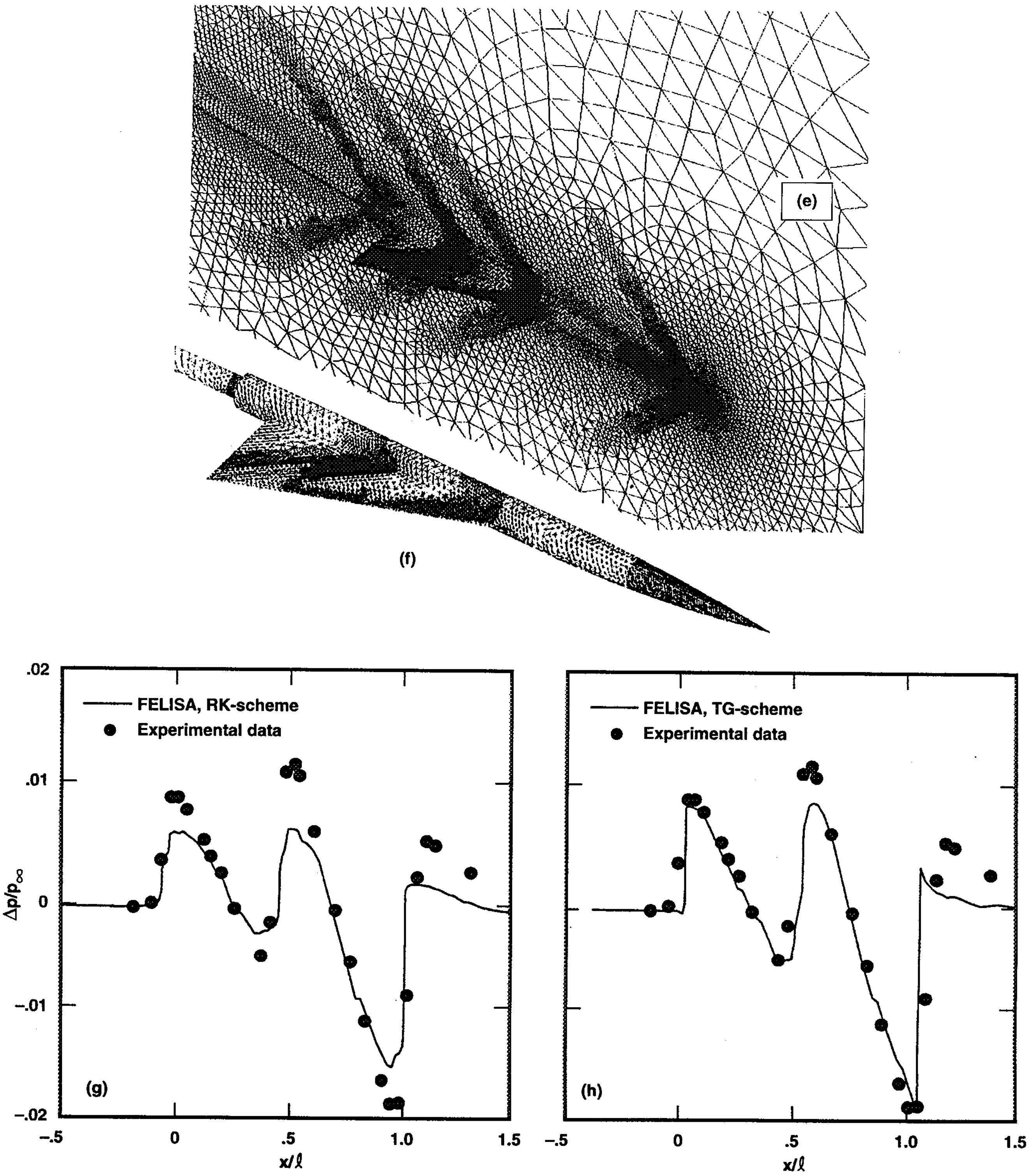

Figure 5. Continued. Comparison of adaptive grid solutions obtained by two solvers in FELISA. (e) Grid on the symmetry plane—930,000 elements and 169,000 points. (f) Triangulations on the model surface. (g) Pressure signatures obtained by Runge–Kutta scheme. (h) Pressure signatures obtained by Taylor–Galerkin scheme.

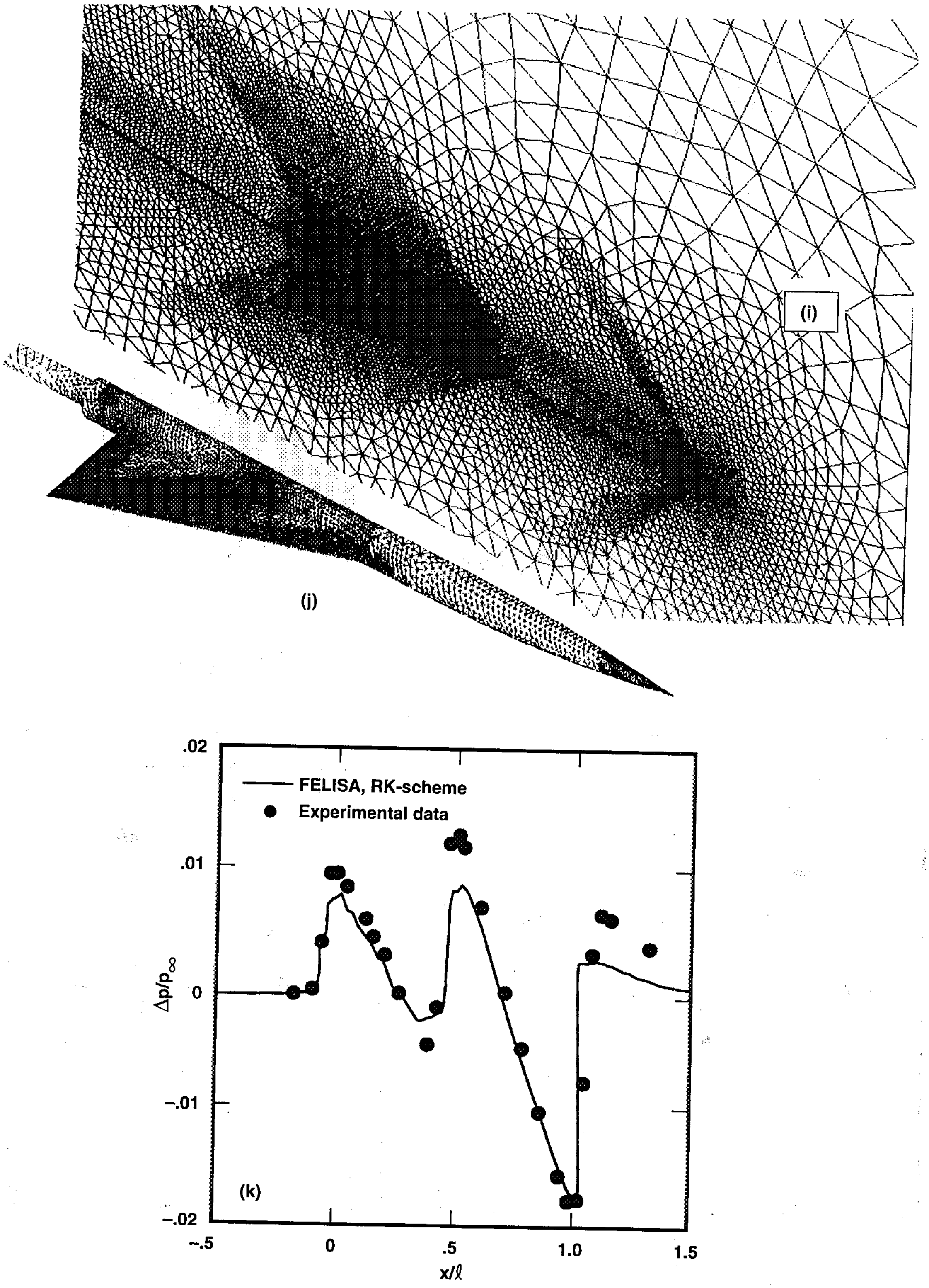

Figure 5. Concluded. Comparison of adaptive grid solutions obtained by Runge–Kutta scheme. (i) Grid on the symmetry plane—1,150,000 elements and 193,000 points. (j) Triangulations on model surface. (k) Pressure signatures obtained by Runge–Kutta scheme.

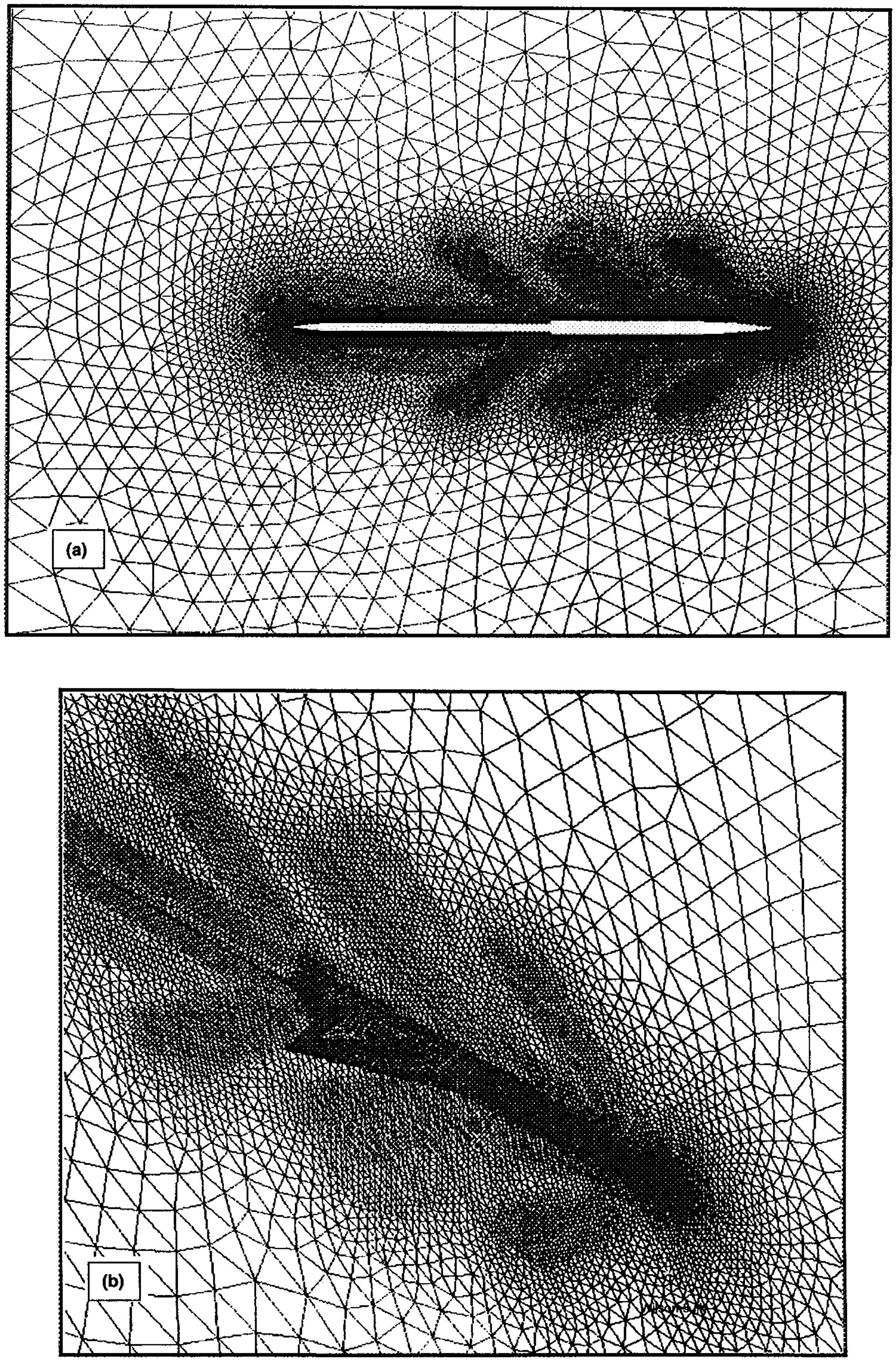

Figure 6. Delta wing-body sting, isometric view of adapted grids to be used for zero and nonzero angles of attack. (a) Grid on the symmetry plane. (b) Isometric view of the grid on the model and symmetry plane, backward-facing-step geometry.

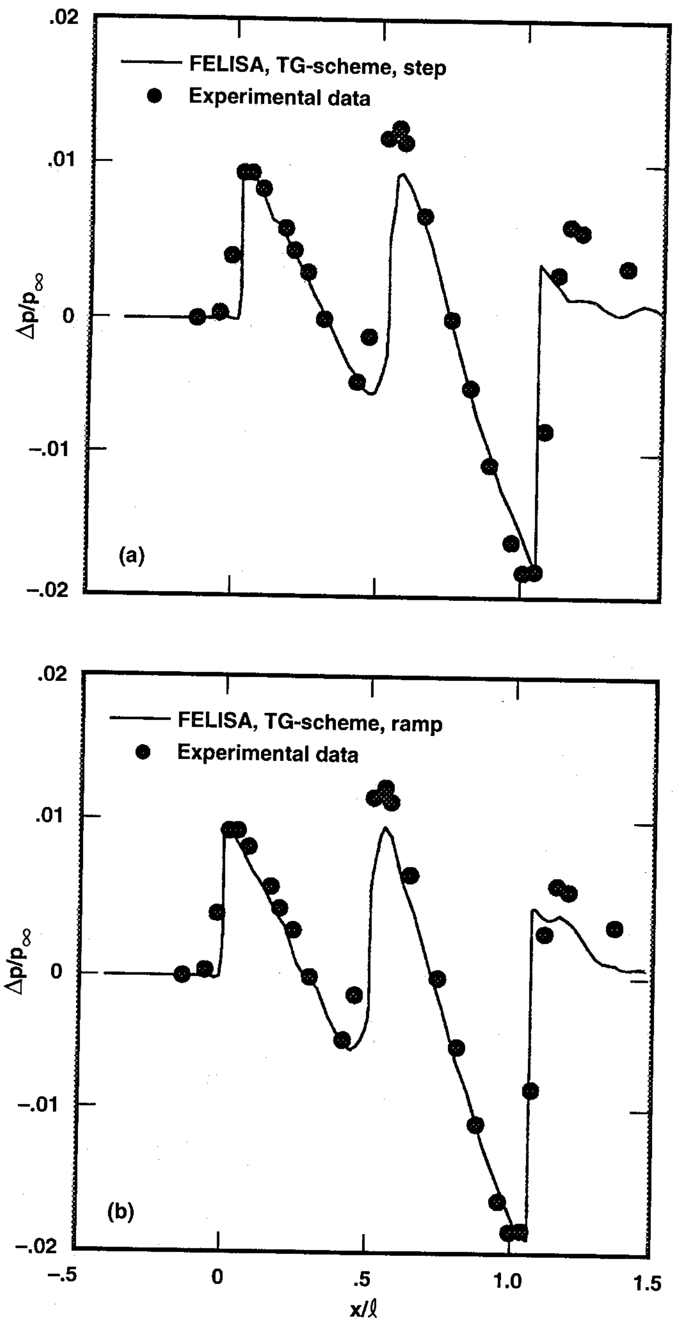

Figure 7. Delta wing-body sting, with $M_\infty = 1.68$ and $\alpha = 0$ deg, pressure signatures interpolated at $h/l = 0.3$ from computed solutions and extrapolated to $h/l = 3.6$, comparison of computed solutions with experiment. (a) Pressure signatures obtained on the backward-facing-step connection—780,000 elements and 142,000 points. (b) Pressure signatures obtained on the ramp connection—796,000 elements and 146,000 points.

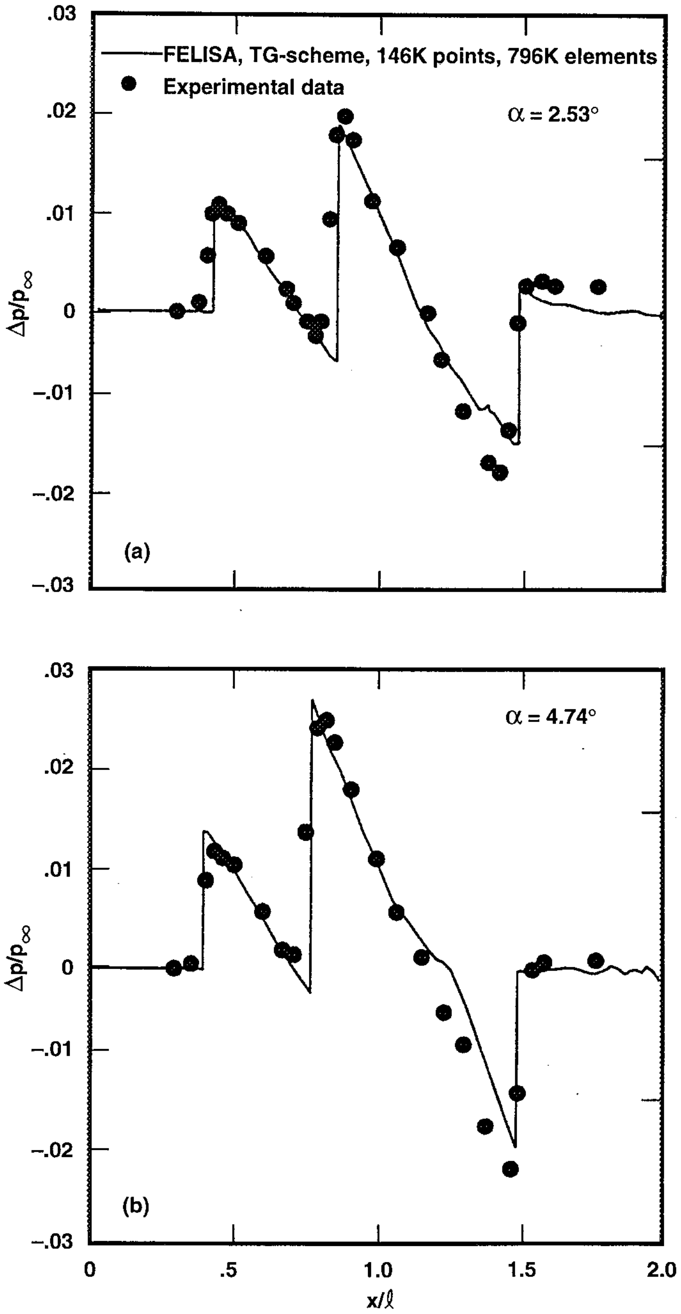

Figure 8. Delta wing-body sting, with $M_\infty = 1.68$ and nonzero $\alpha$, pressure signatures interpolated at $h/l = 0.3$ from computed solution data and extrapolated to $h/l = 3.6$, comparison of computed solutions with experiment, Taylor–Galerkin scheme—796,000 elements and 146,000 points. (a) Pressure signatures corresponding to $\alpha = 2.53$ deg. (b) Pressure signatures corresponding to $\alpha = 4.74$ deg.

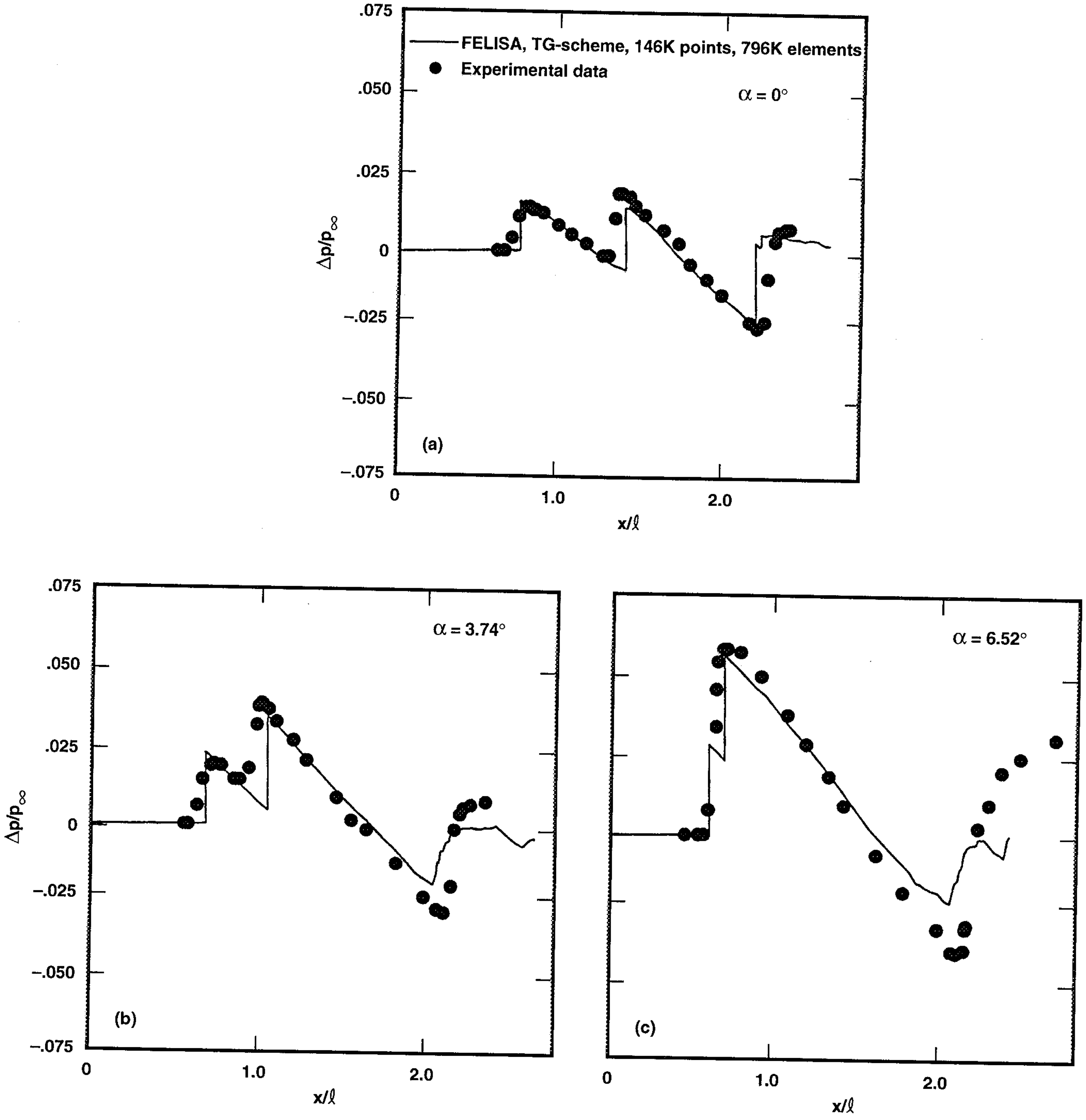

Figure 9. Delta wing-body sting, with $M_\infty = 2.7$ and $\alpha \geq 0$ deg, pressure signatures interpolated at $h/l = 0.3$ from computed solution data and extrapolated to $h/l = 3.1$, comparison of computed solutions with experiment, Taylor–Galerkin scheme—796,000 elements and 146,000 points. (a) Pressure signatures corresponding to $\alpha = 0$ deg. (b) Pressure signatures corresponding to $\alpha = 3.47$ deg. (c) Pressure signatures corresponding to $\alpha = 6.52$ deg.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*

| | | |
|---|---|---|
| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE<br>December 1994 | 3. REPORT TYPE AND DATES COVERED<br>Technical Paper |

4. TITLE AND SUBTITLE

An Assessment of the Adaptive Unstructured Tetrahedral Grid, Euler Flow Solver Code FELISA

5. FUNDING NUMBERS

505-59-20

6. AUTHOR(S)

M. Jahed Djomehri* and Larry L. Erickson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Ames Research Center
Moffett Field, CA 94035-1000

8. PERFORMING ORGANIZATION REPORT NUMBER

A-94147

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TP-3526

11. SUPPLEMENTARY NOTES

Point of Contact: Larry L. Erickson, Ames Research Center, MS 227-2, Moffett Field, CA 94035-1000
(415) 604-6217
*Calspan Corporation, Ames Research Center, Moffett Field, California

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified — Unlimited
Subject Category 02

Available from the NASA Center for Aerospace Information,
800 Elkridge Landing Road, Linthicum Heights, MD 21090; (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A three-dimensional solution-adaptive Euler flow solver for unstructured tetrahedral meshes is assessed, and the accuracy and efficiency of the method for predicting sonic boom pressure signatures about simple generic models are demonstrated. Comparison of computational and wind tunnel data and enhancement of numerical solutions by means of grid adaptivity are discussed. The mesh generation is based on the advancing front technique. The FELISA code consists of two solvers, the Taylor-Galerkin and the Runge-Kutta-Galerkin schemes, both of which are spacially discretized by the usual Galerkin weighted residual finite-element methods but with different explicit time-marching schemes to steady state. The solution-adaptive grid procedure is based on either remeshing or mesh refinement techniques. An alternative geometry adaptive procedure is also incorporated.

14. SUBJECT TERMS

Unstructured grid, Euler flow solver, Supersonic flow

15. NUMBER OF PAGES

36

16. PRICE CODE

A03

| | | | |
|---|---|---|---|
| 17. SECURITY CLASSIFICATION OF REPORT<br>Unclassified | 18. SECURITY CLASSIFICATION OF THIS PAGE<br>Unclassified | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |