

FINITE ELEMENT METHODS FOR INTEGRATED AERODYNAMIC HEATING ANALYSIS

P.22

Report for the Period : 1 October 1990 – 30 September 1991

Prepared for the
Aerothermal Loads Branch
NASA Langley Research Center
Hampton, Virginia 23665, U.S.A.

GRANT NAGW-1809 Year 2
Investigators : K. Morgan and J. Peraire

Submitted by the
Department of Aeronautics
Imperial College of Science, Technology and Medicine
Prince Consort Road
London SW7 2BY
United Kingdom

(NASA-CR-190353) FINITE ELEMENT METHODS FOR
INTEGRATED AERODYNAMIC HEATING ANALYSIS
Report, 1 Oct. 1990 – 30 Sep. 1991
(Imperial Coll. of Science and Technology)
22 p

N92-25959

G3/02 Unclass
0091533

1. INTRODUCTION

This report gives a description of the work which has been undertaken during the second year of a three year research program. The objectives of the program are to produce finite element based procedures for the solution of the large scale practical problems which are of interest to the Aerothermal Loads Branch (ALB) at NASA Langley Research Establishment. The problems of interest range from Euler simulations of full 3D vehicle configurations to local analyses of 3D viscous laminar flow. Adaptive mesh procedures for both steady state and transient problems are to be considered. An important feature of the work is the provision of specialised techniques which can be used at ALB for the development of an integrated fluid/thermal/structural modelling capability.

2. WORK ACCOMPLISHED

Over the past twelve months, research work has concentrated upon improving the capability of our techniques for flow modelling. A new 3D Euler flow solver, employing a low-storage data structure and a high accuracy artificial viscosity, has been written. This solver has been combined with improved versions of our 3D pre- and post-processing procedures for unstructured meshes to produce the FELISA3D system. This provides a more comfortable and convenient working environment for the untrained analyst. A further major development has been an initial demonstration of the feasibility and practicality of combining the new flow solver with a fully unstructured 3D multigrid acceleration procedure. We are also striving towards the production of an appropriate method for the generation and adaptation of grids for viscous flow analysis. To date, this work has concentrated in 2D, but demonstrations have been made which confirm the practicality of employing fully unstructured meshes for the solution of complex viscous high speed flows.

We have begun to consider the application to thermal modelling of some of the techniques developed for flow analysis. This work has concentrated upon an investigation of the viability of using explicit, adaptive unstructured grid methods for the simulation of strongly transient 2D heat conduction problems.

In addition, although we have not undertaken any work on structural modelling, we have developed and delivered to ALB a code for generating and adapting triangular meshes for use in the 3D structural analysis of configurations such as intersecting panels.

A fuller description of the work accomplishments within each of these areas is given below.

2.1 FLOW MODELLING

2.1.1 A New Euler Solver

To date, our most efficient code, in terms of computer memory and speed, for the analysis of 3D Euler flows was based upon a Taylor-Galerkin procedure. For the analysis of steady Euler flows, the solutions produced by this code are dependent on the time step employed and experience shows that the quality of the computed results is very sensitive to the quality of the mesh employed. We have therefore worked on the development of a new solver, based upon the use of an explicit hybrid time-stepping scheme, which maintains the low storage requirements while improving on the computational speed and accuracy as compared with the Taylor-Galerkin code. To achieve the low storage requirements, we have abandoned the traditional finite element mesh data structure and we now employ a data structure which just describes the mesh in terms of its edges, which are the lines joining nodes in the mesh. These edges are numbered and the the numbers of the nodes joined by each edge are stored. For a general tetrahedral mesh, the number of elements (NE), number of vertices (NP), number of edges (NS) and the number of boundary triangles (NBF) are related by the expression

$$2 * NS + \chi = 2 * NE + 2 * NP + NBF \quad (1)$$

where χ denotes a constant which depends upon the Euler characteristic of the surface of the computational domain, with $\chi = 2$ for a simply connected domain. For a typical unstructured tetrahedral mesh, the ratio of elements to vertices is approximately equal to 6 and hence, from equation (1), the ratio of edges to vertices is approximately equal to 7. Thus storage of the mesh connectivity in terms of a finite element based data structure requires 24 storage locations per node, while the edge based data structure requires only 14 storage locations per node.

The use of an edge based data structure of this type can only be contemplated if the basic flow solution algorithm may be reinterpreted and expressed in the same format. To illustrate this process, we consider the solution of the 3D compressible Euler equations in the standard conservation form

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}^j}{\partial x_j} = 0 \quad (2)$$

where the summation convention is employed. The application of a conventional Galerkin finite element approximation procedure, on a mesh of linear tetrahedral elements, results in the requirement that

$$\int_{\Omega} \frac{\partial \mathbf{U}}{\partial t} N_I d\Omega = \int_{\Omega} \mathbf{F}^j(\mathbf{U}) \frac{\partial N_I}{\partial x_j} - \int_{\Gamma} \mathbf{F}^j(\mathbf{U}) n^j N_I d\Gamma \quad (3)$$

for each I , where Ω is the computational domain with boundary surface Γ . N_I is the linear shape function associated with node I of the mesh and n^j denotes the component, in the direction x_j , of the unit outward normal vector to Γ . The approximate solution $\hat{\mathbf{U}}$ is constructed in the finite element form as

$$\hat{\mathbf{U}} = \mathbf{U}_J N_J \quad (4)$$

where \mathbf{U}_J is the approximation to the value of the solution at node J . When the approximate solution of equation (4) is substituted into equation (3), it is found that the left hand side integral can be exactly evaluated to give

$$\int_{\Omega} \frac{\partial \mathbf{U}}{\partial t} N_I d\Omega = \sum_{e \in I} \left\{ \int_{\Omega_e} N_I N_J d\Omega \right\} \frac{d\mathbf{U}_J}{dt} = \mathbf{M}_{IJ} \frac{d\mathbf{U}_J}{dt} \quad (5)$$

where \mathbf{M} denotes the consistent mass matrix and Ω_e the volume of element e . In the formulation which is to be employed, \mathbf{M} is replaced by the lumped (diagonal) mass matrix \mathbf{M}_L . The integrals appearing on the right hand side of equation (3) are evaluated approximately, by assuming that the fluxes may also be represented in terms of their nodal values in the form of equation (4) e.g. for an element e , with nodes I, J, K, L the domain integral is approximated as

$$\int_{\Omega_e} \mathbf{F}^j \frac{\partial N_I}{\partial x_j} d\Omega \approx \frac{\Omega_e}{4} \frac{\partial N_I}{\partial x_j} \Big|_e \left\{ \mathbf{F}_I^j + \mathbf{F}_J^j + \mathbf{F}_K^j + \mathbf{F}_L^j \right\} \quad (6)$$

This expression is readily evaluated, when the finite element based data structure is employed, by looping over the elements in the mesh and sending appropriate element contributions to the relevant nodes. The expression has to be reinterpreted, however, if it is to be used in conjunction with an edge based data structure. Suppose that node I in the mesh is directly connected by edges in the mesh to the m_I nodes J_1, J_2, \dots, J_{m_I} . It will be assumed, for the purposes of illustration, that node I is an interior node of the mesh. Using the results of equations (5) and (6), it is readily shown that equation (3) may be expressed in the form

$$[\mathbf{M}_L]_{II} \frac{d\mathbf{U}_I}{dt} = \sum_{s=1}^{m_I} c_{IJ_s}^j \left\{ \mathbf{F}_I^j + \mathbf{F}_{J_s}^j \right\} \quad (7)$$

where there is no summation over I or J_s and where

$$c_{IJ_s}^j = \sum_{e \in IJ_s} \frac{\Omega_e}{4} \frac{\partial N_I}{\partial x_j} \Big|_e \quad (8)$$

In this equation, the summation extends only over those elements e which contain the edge IJ_s . The quantities $c_{IJ_s}^j$ are termed the weights for node I for the edge IJ_s . With a little thought, it can be shown that the weights possess the properties

$$\sum_{s=1}^{m_I} c_{IJ_s}^j = 0 \quad \text{for all } j \quad (9)$$

$$c_{IJ_s}^j = -c_{J_s I}^j \quad \text{for all } j \text{ and } J_s \quad (10)$$

It is convenient to introduce a vector $\mathcal{C}_{IJ,}$ defined according to

$$\mathcal{C}_{IJ,} = (\mathcal{C}_{IJ,}^1, \mathcal{C}_{IJ,}^2, \mathcal{C}_{IJ,}^3) \quad (11)$$

and to let $\mathcal{L}_{IJ,}$ denote the length of this vector and $S_{IJ,}$ the unit vector, with components $S_{IJ,}^j$, in the direction of $\mathcal{C}_{IJ,}$. Then, equation (7) may be expressed in the form

$$[\mathbf{M}_L]_{II} \frac{d\mathbf{U}_I}{dt} = \sum_{s=1}^{m_I} \mathcal{L}_{IJ,} \{ \mathcal{F}_{I,} + \mathcal{F}_{J,} \} \quad (12)$$

where

$$\mathcal{F}_{I,} = \mathbf{F}_I^j S_{IJ,}^j, \quad \mathcal{F}_{J,} = \mathbf{F}_J^j S_{IJ,}^j, \quad (13)$$

It can now be observed how the discrete form of equation (3) may be obtained by looping over the edges in the mesh and sending edge contributions to the appropriate nodes. The property of conservation in the numerical scheme is guaranteed by the asymmetry of the edge weights as expressed in equation (10).

In the practical implementation of the procedure, the mesh generator produces a tetrahedral mesh defined in the standard finite element form. This mesh is then subjected to a pre-processing stage which computes and stores the entries in the lumped mass matrix and the weights for each edge. This is then the only information which is passed to the flow solver.

The Euler flow solver which has been written to employ this edge based data structure is found to require only around 60 storage locations per node.

2.1.2 Artificial Viscosity Models

The condition of equation (9) means that a time stepping scheme applied to equation (12) will allow for the appearance of chequerboarding modes. The creation of a practically useful algorithm therefore requires the addition of an appropriate artificial viscosity model. This procedure may be alternatively viewed as replacing the actual fluxes in equation (12) by consistent numerical fluxes.

We begin by developing a practically useful scheme for smooth flows. An approximation to the gradient of the solution, at each node of the mesh, can be determined directly, by looping over the edges, as

$$[\mathbf{M}_L]_{II} \frac{\partial \mathbf{U}}{\partial x_j} \Big|_I = - \sum_{s=1}^{m_I} \mathcal{C}_{IJ,}^j \{ \mathbf{U}_I + \mathbf{U}_{J,} \} \quad (14)$$

If σ denotes a coordinate along the edge between nodes I and J , of the mesh, a stable algorithm for smooth flows can be constructed in the form

$$[\mathbf{M}_L]_{II} \frac{d\mathbf{U}_I}{dt} = \sum_{s=1}^{m_I} \mathcal{L}_{IJ,} \left\{ \mathcal{F}_{I,} + \mathcal{F}_{J,} - d_4 |\lambda_{IJ,}| \left[\mathbf{U}_{J,} - \mathbf{U}_I - \left(\frac{\partial \mathbf{U}}{\partial \sigma} \Delta \sigma \right)_{IJ,} \right] \right\} \quad (15)$$

Here d_4 is a user-specified coefficient and $|\lambda_{IJ,}| = |\mathbf{u} \cdot S_{IJ,}| + c_{IJ,}$, where c denotes the speed of sound. The extra terms which have been added to the right hand side of this equation are designed to produce an approximation to a fourth difference operator. In fact, in one space dimension, it is readily shown that the right hand side of equation (15) represents a centered approximation to the flux derivative, plus the addition of a standard representation of a fourth derivative operator, when we use the approximation

$$\frac{\partial \mathbf{U}}{\partial \sigma} \Big|_{IJ,} \approx 0.5 \left[\frac{\partial \mathbf{U}}{\partial \sigma} \Big|_I + \frac{\partial \mathbf{U}}{\partial \sigma} \Big|_{J,} \right] \quad (16)$$

This is known to lead to a stable scheme for smooth flows. It should be noted that this form of added diffusion is more accurate than that previously used in conjunction with centered schemes on unstructured meshes. Previously published artificial viscosity models are such that they result in the addition of diffusion

if the solution \mathbf{U} is linear, whereas the new diffusion model returns the value zero for the diffusion in this case, on any tetrahedral mesh.

An additional smoothing procedure is required when flows involving strong discontinuities are to be modelled. Here this is accomplished by the addition of a pressure switched lower order diffusion to the right hand side of equation (15). The resulting scheme can be written as

$$[\mathbf{M}_L]_{II} \frac{d\mathbf{U}_I}{dt} = \sum_{s=1}^{m_I} \mathcal{L}_{IJ}, \left\{ \mathcal{F}_I + \mathcal{F}_J - |\lambda_{IJ}| \left(d_2 P_{Is} [\mathbf{U}_J - \mathbf{U}_I] - d_4 \left[\mathbf{U}_J - \mathbf{U}_I - \left(\frac{\partial \mathbf{U}}{\partial \sigma} \Delta \sigma \right)_{IJ} \right] \right) \right\} \quad (17)$$

where d_2 is a user-specified coefficient and P_{Is} denotes an edge based pressure switch coefficient for node I and edge s . This switch is constructed so as to have the property that $0 \leq P_{Is} \leq 1$ with $P_{Is} \approx 1$ in the vicinity of flow discontinuities and $P_{Is} \approx 0$ in smooth regions of the flow. Investigations are still being performed to determine the best values for the coefficients d_2 and d_4 and also the most appropriate method of computing the edge switches.

With this form for the right hand side, equation (17) is advanced towards steady state by the use of an explicit multi-stage time stepping method. The convergence of the method is accelerated by the application of residual smoothing, which allows for the use of larger time steps.

2.1.3 The FELISA3D System

The new flow solver has been placed at the heart of the FELISA3D system of codes which provide a complete capability for the solution of steady 3D Euler flows using unstructured tetrahedral meshes. The pre- and post-processing codes which complete the system have also been upgraded and improved. The surface triangulator has now been provided with a singular patch capability which should improve its robustness when used in the analysis of complex configurations. An updated version of the volume generator is also included. New post-processing routines, which are X-Windows based, are provided so that the analyst has a basic provision for viewing the surface mesh and the results of the analysis on most modern workstations. The FELISA3D system of codes has been implemented on the NASA computer system and is now operational at ALB.

To illustrate the performance of the FELISA3D system we consider the computation of the flow over a re-entry vehicle at a Mach number of 6 and an angle of attack of 26.6 degrees. The triangulation of the vehicle is shown in figure 1 (a). The volume mesh consisted of approximately 500,000 tetrahedral elements and was generated using about 40 minutes of cpu time on a CRAY computer. Computation of a converged steady state solution required 120 minutes of cpu time on the same computer. The computed distribution of pressure contours on the surface of the vehicle is shown in figure 1(b).

2.1.4 Unstructured Multigrid

It is well known that the use of an explicit multistage scheme, for the computation of steady solutions of the compressible Euler equations, provides an attractive environment for the effective implementation of multigrid acceleration. The basic idea behind the use of multigrid for the solution of a hyperbolic equation system is to use a sequence of successively coarser grids to compute corrections to the fine mesh solution. The coarse grid equations are also advanced with an explicit multistage scheme, making use of the fact that the allowable time step is now larger than that which can be used on the finest mesh. When the coarse mesh steps are completed, the corrections are interpolated back to the fine mesh and are used to update the fine mesh solution. The form of coarse mesh algorithm which is adopted ensures that a converged solution on the fine mesh is unaltered by the multigrid process i.e. the coarse mesh correction will be zero if the fine mesh residual is zero. The transfer of information between two unstructured tetrahedral grids may be accomplished in a variety of different ways. We have implemented, at present, simple methods of transfer, one of which can be regarded as conservative while the other is not. It is our intention to fully investigate the effect of these transfer operators on the multigrid convergence rate and, if necessary, to implement more sophisticated transfer procedures.

To demonstrate the power of the multigrid approach, it has been employed in the analysis of a transonic flow in a channel. This is a 3D simulation of a 2D flow past a 4% thick circular bump. The free stream

conditions correspond to a Mach number of 0.675. The computation was performed using a fine mesh of 28822 tetrahedra and three coarser meshes of 4137, 556, and 119 tetrahedra respectively. The corresponding triangulations of the boundaries of the computational domain are shown in Figure 2(a)–(d). Figure 2(e) shows the pressure solution computed after 150 multigrid V-cycles. The increase in the rate of convergence by the use of the multigrid method is readily seen in Figure 2(f) which compares the convergence history curve for a flow calculation on the finest mesh with the curve obtained by using the multigrid scheme with four meshes.

The addition of multigrid to the basic Euler flow solver increases the memory requirements to around 87 storage locations per node.

An investigation is currently underway into the performance of this multigrid process when used in the simulation of transonic flows involving complex geometries (such as complete aircraft configurations). When this step is completed, the application of the process to problems involving hypersonic and viscous flows will be considered.

2.1.5 The 24° Compression Corner

We have devoted effort to the computation of the flow over a 24° compression corner, at a free stream Mach number of 14.07. The distance from the leading edge to the corner is 1.44 *ft* and the flow Reynolds number is 72,000 *per ft*. This problem has been the subject of much attention recently. The interest has been caused by the fact that coarse grid two dimensional solutions, computed with the structured grid code CFL3D, appeared to agree well with experiment, whereas the results of experiment and computation diverged, as the grid was uniformly refined. The conclusion was that the flow in the experiment was probably three dimensional in nature. Computations performed at ALB with an upwind unstructured grid cell-centered code appeared to contradict these findings, by demonstrating that adaptive refinement of the grid led to convergence to the experimental results. The adaptive grids were created by adaptive remeshing, using triangles, in the essentially inviscid areas of the flow and by manual adaptation, using quadrilaterals, in the viscous dominated regions. This approach led to the creation of cells with much higher aspect ratios in the boundary layer and so the application of this form of adaptivity may not have led to the computation of a grid independent solution. For this reason, Rajiv Thareja at ALB has been attempting to recompute this example with the cell-centered upwind code on totally triangular meshes with acceptable aspect ratios throughout. However, problems have been encountered with this approach, at the solid surface, which are yet to be adequately resolved.

We have employed an explicit Taylor–Galerkin approach and solved the problem on the structured quadrilateral meshes used previously and on triangulations of these meshes. The results of these computations are shown in Figures 3 (a)–(c), which show the distributions of pressure, skin friction and heat transfer coefficients respectively over the solid surface. The computed separation and reattachment points on each mesh are shown in Table 1.

| Mesh | Separation Point | Reattachment Point |
|----------------------------------|------------------|--------------------|
| 101 <i>by</i> 101 triangles | 0.5338–0.5766 | 1.305–1.315 |
| 201 <i>by</i> 201 triangles | 0.461–0.481 | 1.393–1.403 |
| 101 <i>by</i> 101 quadrilaterals | 0.491–0.5338 | 1.388–1.408 |
| 201 <i>by</i> 201 quadrilaterals | 0.461–0.481 | 1.403–1.413 |
| experiment | 0.5475–0.7025 | 1.2728–1.3748 |

TABLE 1

It may be observed that the trend is to move the computational results away from those of experiment as the grid is refined. This is therefore in agreement with the results produced by the CFL3D code. It can be observed that there is an indication in the results that quadrilaterals are better suited for resolving boundary layers, as the triangular results on the 201 *by* 201 mesh lie close to the quadrilateral results on the 101 *by* 101

mesh. However, before such a strong conclusion can be drawn, further work is required to investigate the following effects : (a) the form of triangulation employed, as the best triangulation does not necessarily result from subdividing a quadrilateral mesh. (b) the effect of the artificial viscosity model employed on the quadrilateral mesh, as the construction of the artificial diffusion assumes a triangulation derived from a subdivision of the quadrilateral mesh.

The situation becomes more confused when we consider the results produced for this problem by a cell-vertex upwind code. Wall distributions of pressure, skin friction and heat transfer coefficients computed on a triangulation of a 101 *by* 101 mesh of quadrilaterals are shown in Figures 4(a)–(c). The numerical solutions are seen to lie much closer to the experimental measurements than the solutions produced by CFL3D. In Figure 4(d) we compare the heat transfer coefficient computed on the 101 *by* 101 mesh of quadrilaterals with that computed on the triangular mesh. The predicted distribution on the quadrilateral mesh is seen to lie even closer to the experimental values.

Clearly, we have not yet resolved this problem satisfactorily and further algorithmic development and computation will be required before a definitive conclusion can be reached about the reasons for these discrepancies.

2.1.6 Adaptivity for Viscous Flow Modelling

It has been noted above that our present capability for mesh adaptation for complex viscous flows is not satisfactory. In fact, grid generation itself is a major problem facing the analyst who is interested in solving viscous flows by an unstructured grid method. Although the approach of generating a coarse grid and then refining adaptively by enrichment will probably work in two dimensions, it will generally prove to be too expensive to utilise in three dimensions, as it will lead to the generation of an enormous number of elements.

We have investigated the possibilities of developing an adaptive strategy, for steady two dimensional high speed viscous flows, which is based upon the use of a completely unstructured mesh. The use of an unstructured computational mesh provides a natural framework for incorporating adaptive mesh methods into the solution procedure. The approach which is generally adopted is to advance the solution towards steady state on an initial mesh and then to subject the computed solution to an error indicating procedure. Based upon the results of this procedure, the grid is adapted in some fashion and the process is repeated until the analyst is satisfied with the quality of the computed solution. The available adaptivity techniques can be generally classified into three different categories :

1. *Adaptive Remeshing.* This approach uses an error indicator based upon the second derivative of the computed solution and provides a new distribution of mesh size over the computational domain. A completely new mesh is then generated for the problem by using a triangular mesh generator which attempts to meet this prescribed distribution of element size as closely as possible. The user specifies the minimum mesh spacing to be used at any stage of the process. By allowing for the use of stretched elements, any directionality in the solution can also be reflected in the form of the new mesh. However, to preserve mesh quality, a maximum stretching of around 5 is generally allowed. Adaptive remeshing has been shown to work well for the solution of Euler flows, but it is not well suited to the solution of flows involving both shocks and thin viscous layers. The error indicators employed have difficulty differentiating between the different flow features. This means that if the mesh size necessary to adequately resolve the boundary layer is used as the minimum mesh size, this will be allocated to both shock and boundary layer regions and an excessively large number of elements will be generated in the new mesh.

2. *Mesh Movement.* In this approach the adaptivity is accomplished by redistributing the nodes in the grid, while maintaining the inter-node connectivity of the initial mesh. This redistribution is achieved by considering the element sides as springs of a prescribed stiffness and then moving the nodes until the spring system is in equilibrium. We choose to define this spring stiffness in terms of a non-dimensionalised difference in the absolute value of the velocity at the two nodes. The assembled system is brought into equilibrium by simple iteration and the new nodal coordinates computed. This method experiences severe problems in dealing with flows which involve more than one significant feature.

3. *Mesh Enrichment.* The obvious method of introducing more degrees of freedom into the system is to allow for the local addition of new nodes and the creation of new elements. This method can be implemented by sweeping over all sides in the mesh and again computing the non-dimensionalised difference in the absolute

value of the velocity at the two nodes of the side. A new node is added to any side at which this computed value exceeds a user-specified threshold. When the point addition procedure is completed, the element subdivision is undertaken. It has already been noted that this method, when applied to the solution of realistic viscous flows, generally leads to the use of a large number of degrees of freedom before mesh convergence of the results can be demonstrated.

It is apparent that all of these adaptivity methods have their attractive features, but that none of them is completely satisfactory for the solution of a wide class of flows.

In an attempt to produce a workable procedure, we have investigated the performance of an adaptive strategy which is based upon a combination of the methods described above. The analysis has been undertaken for the problem of Mach 10 flow over a 15° compression corner at a Reynolds number of 143,800. The solutions are obtained by means of the explicit Taylor-Galerkin method. With the solution computed on a general triangular mesh, a new mesh for the problem is produced by adaptive remeshing. The objective here is a repositioning of the degrees of freedom and the removal of elements from regions in which the flow is nearly uniform. Starting from the computed solution on the mesh of Figure 5(a), the new mesh obtained is shown in Figure 5(b) and has the same minimum element size and consists of 5,893 elements and 3,029 points. The solution is advanced for 1,000 steps and one level of enrichment is performed, in order to place more points in the boundary layer region. The mesh produced is shown in Figure 5(c) and consists of 6,877 elements and 3,582 points. After the advancement of the solution for a further 1,000 timesteps a mesh movement algorithm is employed. The algorithm introduced here is a refinement on the basic form described above in that (a) the minimum height of any element is prevented from becoming less than a user-prescribed value; (b) diagonal swapping is attempted for any element for which an interior angle exceeds 160°. If this is not found to improve the local situation, the angle is halved by adding a new mesh point to the opposite side. At the end of the procedure new elements are created to maintain the integrity of the mesh; (c) to maintain the definition of weak features in the flow, a local adaptive remeshing is performed in those regions where the use of mesh movement has led to the appearance of elements which are coarser than those which would have been produced by the application of adaptive remeshing. The resulting grid is shown in Figure 5(d) and consists of 10,382 elements and 5,346 points. Details of this mesh, at a point on the flat plate and in the vicinity of the corner, are shown in Figures 6(a) and 6(b). It can be seen, in Figure 7(a) to 7(c), that the computed wall profiles of pressure coefficient, skin friction coefficient and Stanton number are in excellent agreement with the values produced by an analysis of the same problem using a fine structured triangular mesh.

This exercise has proved to be worthwhile as it has certainly killed of the argument that practical viscous flows cannot be adequately resolved on general unstructured meshes. However, the approach which we have employed is rather cumbersome and unwieldy and we intend to investigate alternative methods of obtaining the same type of results in the next year of the project.

2.2 THERMAL MODELLING

Within this general area, we have investigated the development of an adaptive finite element method for the solution of transient heat conduction problems in two dimensions. The computational domain is represented by an unstructured assembly of linear triangular elements and the mesh adaptation is achieved by local regeneration of the grid, using an error estimation procedure coupled to an automatic triangular mesh generator. The solution is advanced by explicit timestepping, with domain decomposition being used to improve the computational efficiency of the procedure. The software is capable of dealing with arbitrarily shaped two dimensional domains.

The equation governing two dimensional transient heat conduction is considered in the form

$$\frac{\partial H}{\partial t} = \frac{\partial}{\partial x} \left[k \frac{\partial T}{\partial x} \right] + \frac{\partial}{\partial y} \left[k \frac{\partial T}{\partial y} \right] \quad (18)$$

where T is the temperature, k is the thermal conductivity and the enthalpy H is defined by

$$dH = \rho c dT \quad (19)$$

where ρ and c denote the density and the specific heat respectively. The region of interest, Ω , is discretised using 3-noded linear triangular elements and an approximate solution is sought in the form

$$T \approx T = N_J(x, y)T_J(t) \quad H \approx H = N_J(x, y)H_J(t) \quad (20)$$

where the implied summation extends over all the nodes in the mesh. Application of the Galerkin variational procedure leads to the matrix differential equation system

$$\mathbf{M} \frac{d\mathbf{H}}{dt} = \mathbf{b} - \mathbf{K}\mathbf{T} \quad (21)$$

where

$$\mathbf{H}^T = (H_1, H_2, \dots) \quad \mathbf{T}^T = (T_1, T_2, \dots) \quad (22)$$

and

$$\mathbf{K}_{IJ} = \int_{\Omega} k \left\{ \frac{\partial N_I}{\partial x} \frac{\partial N_J}{\partial x} + \frac{\partial N_I}{\partial y} \frac{\partial N_J}{\partial y} \right\} d\Omega \quad (23)$$

In equation (21), \mathbf{b} represents the terms which arise due to the imposition of the problem boundary conditions. A simple forward difference approximation is made, with lumping of the mass matrix, leading to the explicit time-stepping scheme

$$\mathbf{H}^{n+1} = \mathbf{H}^n + \Delta t [\mathbf{M}_L]^{-1} [\mathbf{b}^n - \mathbf{K}^n \mathbf{T}^n] \quad (24)$$

where the superscript n denotes an evaluation at time $t = t_n$ and the timestep $\Delta t = t_{n+1} - t_n$.

Equation (24) is a relatively inefficient solution method for heat conduction problems, because of the stability restrictions placed on the allowable timestep size. However, the efficiency of this scheme can be improved by combining it with an automatic domain decomposition method. Instead of using a single timestep throughout the computational domain, the objective now is to group the elements according to the maximum allowable timestep size and to advance the solution independently within each group. Time accuracy of the procedure is maintained by appropriate interchange of information across the boundaries of the groups. The approach has been employed previously for the explicit solution of hyperbolic equation systems, and its implementation in the current context proceeds algorithmically as follows :

(i) before advancing the solution, compute the allowable stability timestep Δt_e for each element according to

$$\Delta t_e = \frac{h_e^2}{2\kappa_e} \quad (25)$$

where κ_e is the element diffusivity and h_e is a representative element length.

(ii) determine the minimum timestep Δt_{min} allowed on the mesh and sort the elements into m regions according to the ratio $\Delta t_e / \Delta t_{min}$ e.g.

| region | all elements e for which |
|--------|--|
| 1 | $\Delta t_{min} \leq \Delta t_e \leq 2\Delta t_{min}$ |
| 2 | $2\Delta t_{min} \leq \Delta t_e \leq 4\Delta t_{min}$ |
| 3 | $4\Delta t_{min} \leq \Delta t_e \leq 8\Delta t_{min}$ |

(iii) for each region, m , find the boundary nodes of the region.

(iv) overlap the regions by adding two layers of elements to each region m from the regions $p > m$.

(iv) advance the solution one global timestep, e.g. with three regions the solution is advanced through $4\Delta t_{min}$ by following the sequence

| region | no. of timesteps | interchange of information |
|--------|------------------|----------------------------|
| 1 | 2 | |
| 2 | 1 | $region1 - region2$ |
| 1 | 2 | |
| 2 | 1 | $region1 - region2$ |
| 3 | 1 | $region2 - region3$ |

This domain splitting procedure will be most useful when used in conjunction with adaptive refinement techniques, where large variations in the element sizes (and hence, from equation (25), in element timesteps) over the computational domain will be encountered.

The generation of triangular meshes is accomplished by the advancing front method with the nodal points placed according to a prescribed distribution of the mesh point spacing δ . This specification is accomplished by employing a 'background' grid of linear triangular elements and specifying values of δ at the nodes of this grid. During the grid generation, local values of δ are obtained by linear interpolation over the background grid elements.

With such an automatic triangulation capability, it is possible to contemplate the solution of transient heat conduction problems using adaptive grid methods. An algorithm can be proposed in which the sequence of steps may be described as follows:

- (i) an initial grid of three noded triangular elements is generated to represent the computational domain and to adequately resolve the initial solution
- (ii) the solution is advanced for a prescribed number of time steps
- (iii) an error indicator is used to identify the regions of the mesh which should be refined or coarsened and a new distribution for the mesh parameter δ is computed
- (iv) elements which lie in the regions which have been identified are deleted, thus creating a sequence of holes in the original mesh
- (v) the holes are triangulated, according to the new distribution of the mesh parameters, and a new grid is obtained which is adapted to the solution
- (vi) the nodal values of the solution on the new grid are obtained by interpolation from the previous grid
- (vii) return to step (ii).

This procedure will involve the use of searching operations and, to date, this has been accomplished by the use of simple searching techniques. To speed up the triangulation process, each hole in the mesh is considered separately, so that the searching is performed only over those sides of the front for the current region. As the computation advances, more small regions need to be remeshed, rather than one large region, and the effect of this is to ensure that the remeshing procedure is accomplished faster, even though there may be an increase in the total number of elements generated. The success of this process will depend crucially on the ability to define a suitable error indicator, which must be capable of determining an optimal distribution of the mesh parameters for use by the triangular mesh generator.

A simple error indicator based upon the use of interpolation theory has been applied to give an indication of the accuracy of the computed solution. The basic assumption is that, at any instant, the computed solution is exact at the nodes. The deviation, over each element, between the computed solution and a higher order interpolation through the nodal values, is determined and grid adaptation is accomplished by imposing the requirement of equi-distribution of the indicated error. We employ a multi-dimensional extension of a one dimensional technique based upon the use of the root mean square value of the indicated error over each element. At any instant, the error E is defined as

$$E = T - \hat{T} \quad (26)$$

Assuming that the computed linear solution is exact at the nodes, the error E can be estimated as the difference between a quadratic interpolation and the linear computed solution. In this case, the variation of E within element e can be expressed as

$$E_e = \frac{1}{2} \zeta (\ell_e - \zeta) \left. \frac{d^2 T}{dx^2} \right|_e \quad (27)$$

where ζ is a local element coordinate and ℓ_e denotes the element length. The second derivative of the piecewise linear solution is estimated by variational recovery techniques. The root mean square value of the error over element e can then be shown to be given by

$$E_e^{RMS} = \frac{1}{\sqrt{120}} \ell_e^2 \left| \frac{d^2 T}{dx^2} \right|_e \quad (28)$$

We define the 'optimal' mesh, for a given degree of accuracy, as the mesh in which this root mean square error is equal over each element. The requirement is therefore written as

$$\ell_e^2 \left| \frac{d^2 T}{dx^2} \right|_e = C \quad (29)$$

where C denotes a positive constant. The requirement of equation (29) suggests that the optimal spacing δ on the new adapted mesh should be computed according to

$$\delta_e^2 \left| \frac{d^2 \hat{T}}{dx^2} \right|_e = C \quad (30)$$

This expression may be directly extended to the 2 dimensional case by employing the quadratic form

$$\delta_\beta^2 \left(\sum_{i,j=1}^2 m^{ij} \beta^i \beta^j \right) = C \quad (31)$$

Here β is an arbitrary unit vector, δ_β is the spacing along the direction of β , and m^{ij} are the components of a 2 by 2 symmetric matrix, \mathbf{m} , of second derivatives of \hat{T} , i.e.

$$m^{ij} = \frac{\partial^2 \hat{T}}{\partial x^i \partial x^j} \quad (32)$$

The 'optimal' values for the mesh parameters are calculated at each node of the current mesh. Directions α_i ; $i = 1, 2$ are taken to be the principal directions of the matrix \mathbf{m} . The corresponding mesh spacings are computed from the eigenvalues λ_i of the matrix \mathbf{m} as

$$\delta_i = \sqrt{\frac{C}{\lambda_i}} \quad (33)$$

The spatial distribution of the mesh parameters is completely defined when a value is specified for the constant C . In the practical implementation of the method, two threshold values for the computed spatial distribution of spacing are used: a minimum spacing δ_{min} and a maximum spacing δ_{max} , such that

$$\delta_{min} \leq \delta_i \leq \delta_{max} \quad \text{for } i = 1, 2 \quad (34)$$

The definition of the value δ_{max} is to account for the possibility of a vanishing eigenvalue in (33) which would render that expression meaningless. The value of δ_{max} is chosen as the spacing which will be used in the regions where the temperature distribution is uniform. On the other hand, when maximum values of the second derivatives occur the error indicator will demand that smaller elements are required. The error indicator of equation (33) can be used to identify areas in which the mesh should be refined or coarsened. This is accomplished by marking for removal all nodes P at which the computed optimum spacing δ_P differs from the spacing on the previous mesh by more than a factor of 15%. All elements connected to these nodes are removed from the mesh and the original grid remains, but with one or more regions of elements removed. The advancing front generator fills the holes which have been created in the mesh by constructing new elements according to the distribution of mesh parameters provided by equation (33).

The example included to illustrate the numerical performance which may be achieved by the use of the adaptive remeshing approach in the analysis of transient thermal problems is that of a cylindrical surface which is subjected to heating by a moving heat source. This problem is representative of that encountered in the analysis of scramjet engine structures for hypersonic vehicles, where leading edges are subjected to highly localised, intense aerothermal loads induced by shock-shock interactions.

In the first analysis, a heat source of constant magnitude of $26,000 \text{ BTU/ft}^2 - s$ and of width 0.01 in is applied to the external surface of a half-cylinder of outer radius 0.125 in and inner radius 0.11 in . The heat source is initially positioned at the lower edge of the half-cylinder ($\theta = -90^\circ$) and subsequently moves around the cylinder at a speed of 2 in/sec . This distribution of the heat flux is meant to be representative of the conditions which result when the nose bow shock of the vehicle sweeps across and interacts with the leading edge bow shock. Monitoring points are placed at six locations, with three on the external surface at $\theta = -30^\circ$, 15° and 60° and three at the corresponding locations on the curved internal surface. The curved internal surface of the half-cylinder is subjected to a convection boundary condition, with heat transfer coefficient $h = 7.8 \text{ BTU/ft}^2 - s - ^\circ R$ and surrounding temperature $T_0 = 50^\circ R$. The vertical faces of the half-cylinder are maintained at a constant temperature of $50^\circ R$. A radiation boundary condition is also applied to the outer surface. The half-cylinder is assumed to be made of nickel-based superalloy and representative values for the thermal properties were taken to be $\rho = 0.283 \text{ lbm/in}^3$, $c = 0.1825 \text{ BTU/lbm} - ^\circ R$ and $k = 2 \times 10^{-4} \text{ BTU/in} - s - ^\circ R$.

Initially, fixed mesh simulations were performed, using what were considered to be a representative fine and a representative coarse mesh of unstructured triangular

| Mesh | Remeshing | Domain Splitting | cpu Time (Min) |
|---------|-----------|------------------|----------------|
| Fine | No | No | 1018 |
| Fine | No | Yes ¹ | 860 |
| Adapted | Yes | No | 197 |
| Adapted | Yes | Yes ² | 143 |

¹maximum of 2 domains

²maximum of 4 domains

TABLE 2

elements. The fine mesh consisted of 12,287 elements and 6,527 points, while the coarse mesh had 1,111 elements and 678 points. The adaptive mesh procedure was then employed and examples of the meshes produced at three stages of the computation are displayed in Figure 8. For the adapted meshes, the minimum and maximum mesh sizes employed correspond to those used in the fixed fine and coarse meshes respectively. Figure 9 shows a comparison of the computed temperature variations with time at the monitoring points on the external and internal surfaces of the cylinder. It can be observed that the coarse mesh severely underpredicts the maximum temperature attained, whereas the fine mesh and the adapted mesh predictions are in excellent agreement. The computations were performed on a SUN3 workstation and an indication of the effectiveness of the proposed procedures can be gained from the cpu requirements for these examples which are detailed in Table 2.

In a second example, the amplitude of the applied heat source was varied in time as the source moves along the exterior surface of the half-cylinder. The source strength varies from a maximum of $26,000 \text{ BTU}/\text{ft}^2 - \text{s}$, at $\theta = -24^\circ$, to zero, at $\theta = -90^\circ$ and at $\theta = 90^\circ$. This is more representative of the conditions experienced as the bow shock sweeps across the leading edge. Only the adaptive mesh procedure was employed in this case and examples of the meshes produced at three stages of the computation are displayed in Figure 10. For the adapted meshes, the minimum and maximum mesh sizes employed are the same as those used in the previous analysis. Figure 11 shows the computed temperature variations with time at the monitoring points on the external and internal surfaces of the cylinder and the effect of the time variation in the amplitude of the heat source is clearly apparent.

This work has been successful in demonstrating that the use of adaptive remeshing is a promising approach for the computation of strongly transient heat conduction analysis on unstructured triangular meshes. Further studies will be undertaken into the best method of advancing the solution in time in such problems. The role to be played by implicit timestepping methods will be a subject of particular interest.

2.3 STRUCTURAL MODELLING

Although we have not worked directly in the area of structural modelling, we were requested, by Pramote Dechaumphai of ALB, to provide a suitably modified version of the surface triangulation software. The requirement was for a capability for modelling surfaces which are typically encountered when performing structural analysis of intersecting panels. This software has been delivered to ALB during this period, together with a facility for adaptively regenerating the resulting surface mesh according to requirements of a user-specified error indicator.

3. CONCLUSIONS

Research work has been successfully undertaken in a wide variety of areas during this twelve month period. The most notable achievement of this work have been the release to NASA of the FELISA3D system with its new Euler flow analysis modelling capability. Slow progress has been made in the development of procedures for the analysis of high speed viscous flows. It is hoped that this progress will accelerate over the next twelve month period, within the framework provided by the new developments which have been made in flow solver technology for general unstructured meshes.

PUBLICATIONS

1. A.R. Wieting, P. Dechaumphai, K.S. Bey, E.A. Thornton and K. Morgan, 'Application of integrated fluid-thermal-structural analysis methods', *Thin-Walled Structures* 11, 1-23, 1991
2. O. Hassan, K. Morgan and J. Peraire, 'An implicit finite element method for high speed flows', *Int. J. Num. Meth. Engng.* 32, 183-205, 1991
3. K. Morgan, J. Peraire, J. Peiro and O. Hassan, 'The computation of three dimensional flows using unstructured grids', *Comp. Meth. Appl. Mech. Engng.* 87, 335-352, 1991
4. J. Probert, O. Hassan, J. Peraire and K. Morgan, 'An adaptive finite element method for transient compressible flows', *Int. J. Num. Meth. Engng.* 32, 1145-1159, 1991
5. J. Probert, O. Hassan, K. Morgan and J. Peraire, 'An adaptive finite element method for transient compressible flows involving moving boundaries', *Int. J. Num. Meth. Engng.* 32, 751-765, 1991
6. O. Hassan, J. Peiro, J. Peraire and K. Morgan, 'The application of an adaptive unstructured grid method to the solution of hypersonic flows past double ellipse and double ellipsoid configurations', in *Hypersonic Flows for Re-Entry Problems I*, Springer Verlag, 1991
7. O. Hassan, K. Morgan and J. Peraire, 'A contribution to Problem III', in *Proc. Workshop on Hypersonic Flows for Re-Entry Problems*, Antibes, 1991
8. O. Hassan, K. Morgan and J. Peraire, 'A contribution to Problem IV', in *Proc. Workshop on Hypersonic Flows for Re-Entry Problems*, Antibes, 1991
9. O. Hassan, K. Morgan, J. Peraire, E.J. Probert and R.R. Thareja, 'Adaptive unstructured mesh methods for steady viscous flow', *AIAA 10th CFD Conference*, Hawaii, Paper AIAA-91-1538, 1991
10. E.J. Probert, O. Hassan, K. Morgan and J. Peraire, 'Adaptive remeshing applied to the thermal analysis of a convectively cooled cylindrical leading edge', in *Proc. 7th Int. Conf. on Numerical Methods in Thermal problems*, Stanford, 811-823, 1991

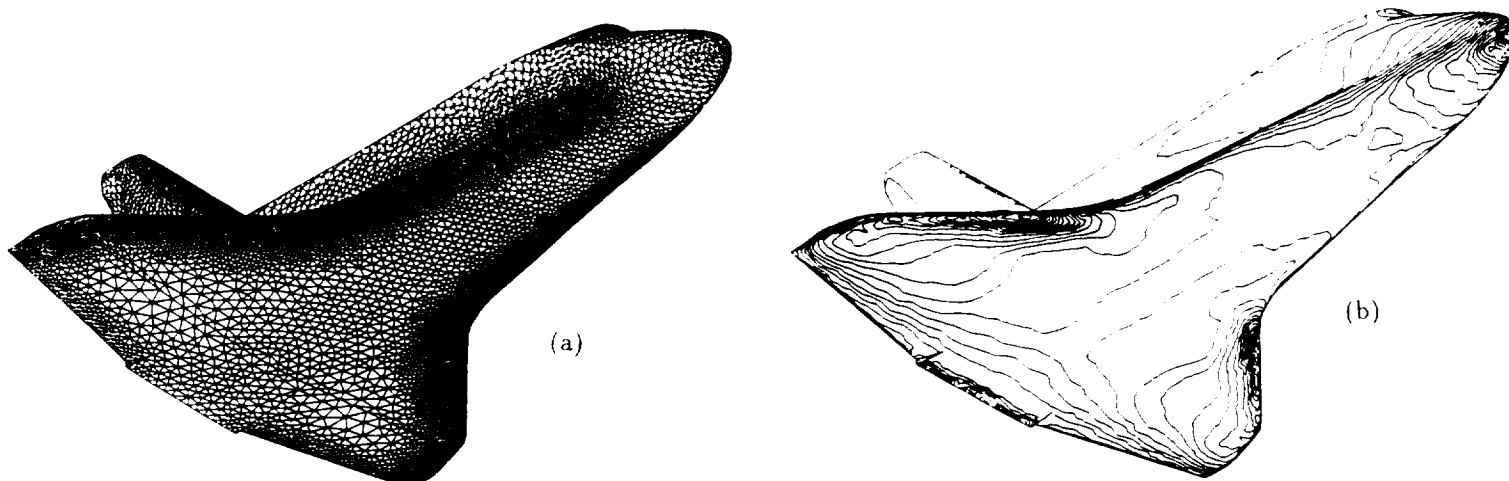


Figure 1

Computation of inviscid flow past a re-entry vehicle at a free stream Mach number of 6 and at an angle of attack of 26.6° showing (a) the surface discretisation and (b) the surface pressure distribution.

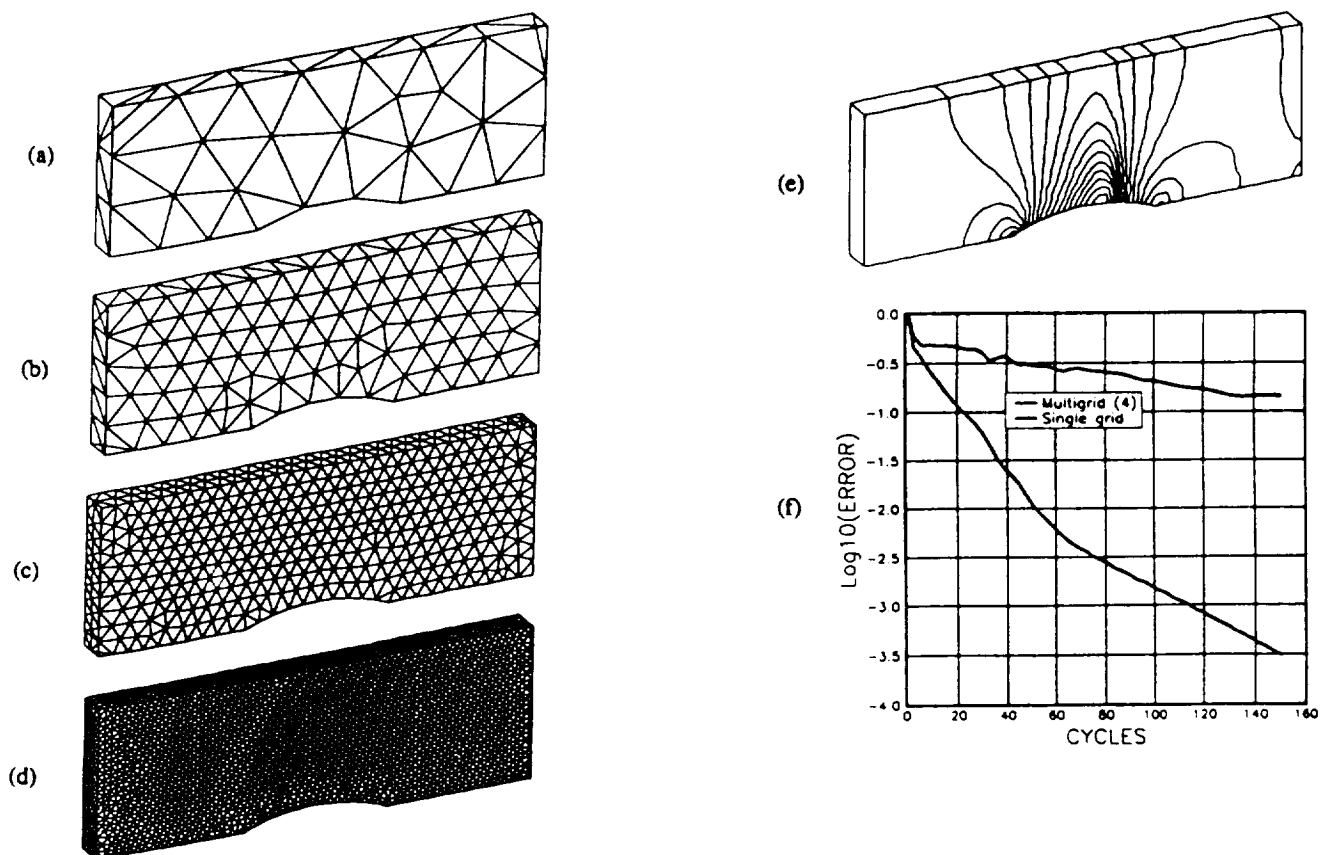


Figure 2

Transonic flow in a channel at a free stream mach number of 0.675 showing (a) first mesh: 119 elements, 63 points; (b) second mesh: 556 elements, 216 points; (c) third mesh: 4137 elements, 1142 points; (d) fourth mesh: 2733 elements, 723 points; (e) computed distribution of pressure; (f) comparison of the convergence

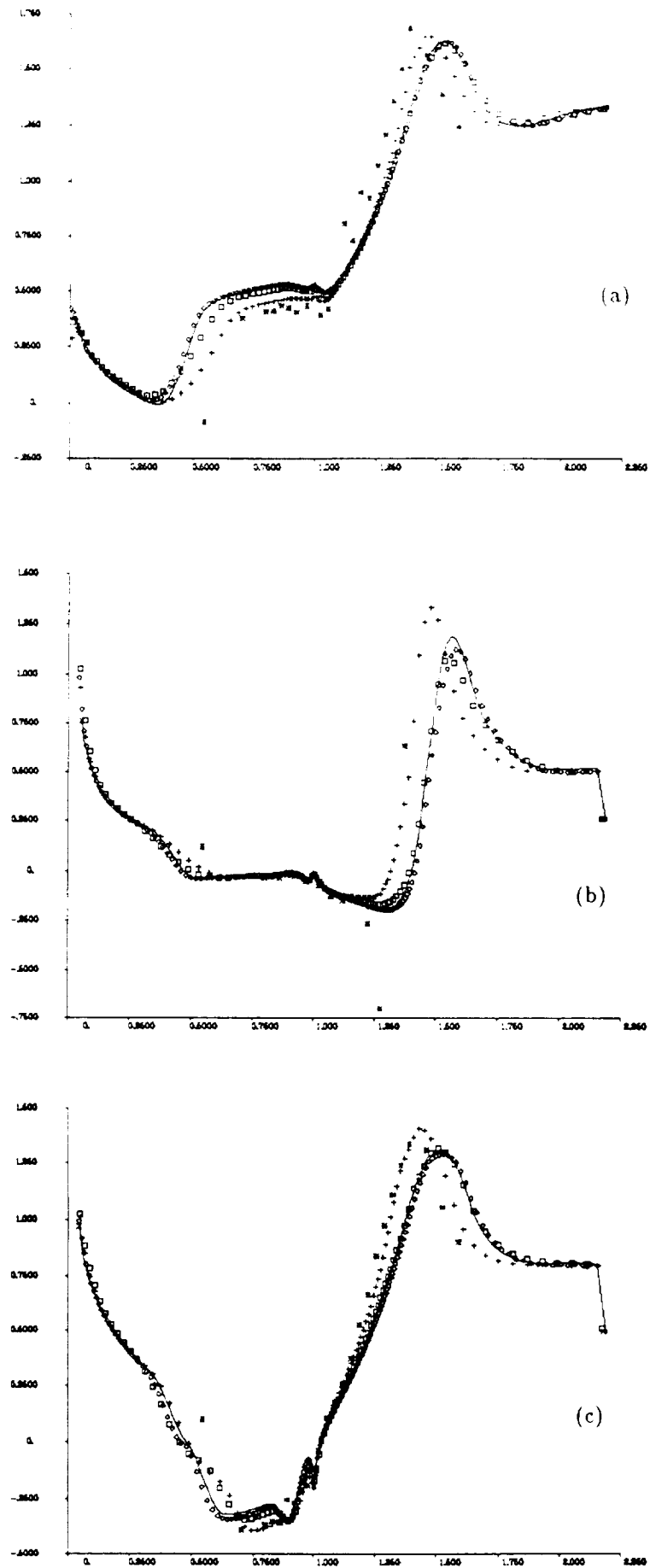
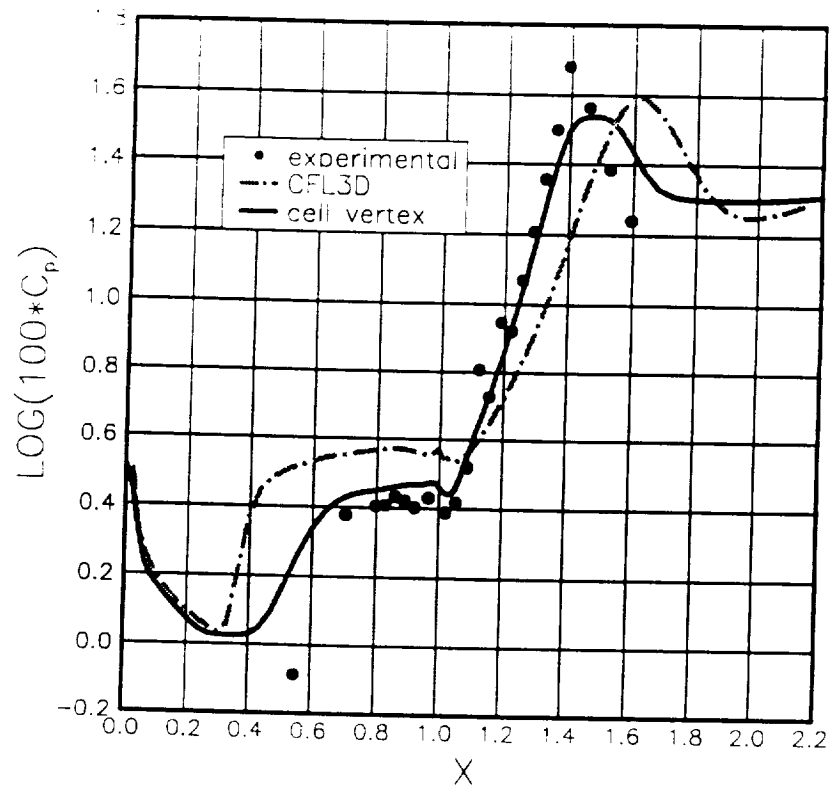
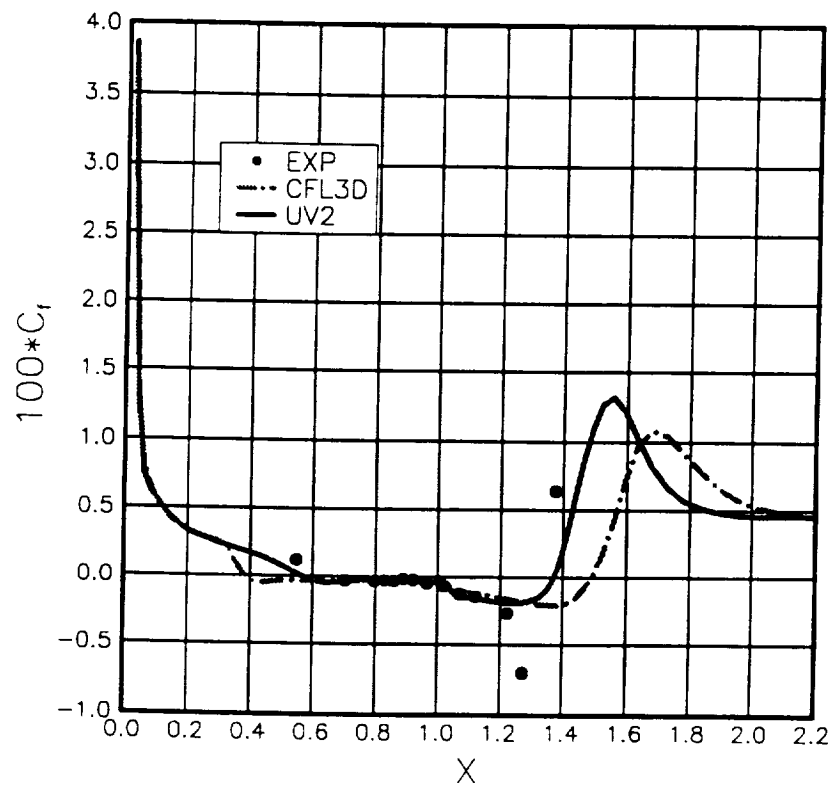


Figure 3

Mach 14.07 flow over a 24° compression corner showing the wall distributions of (a) pressure coefficient (b) skin friction coefficient (c) heat transfer coefficient. Results are from the 2D calculations.

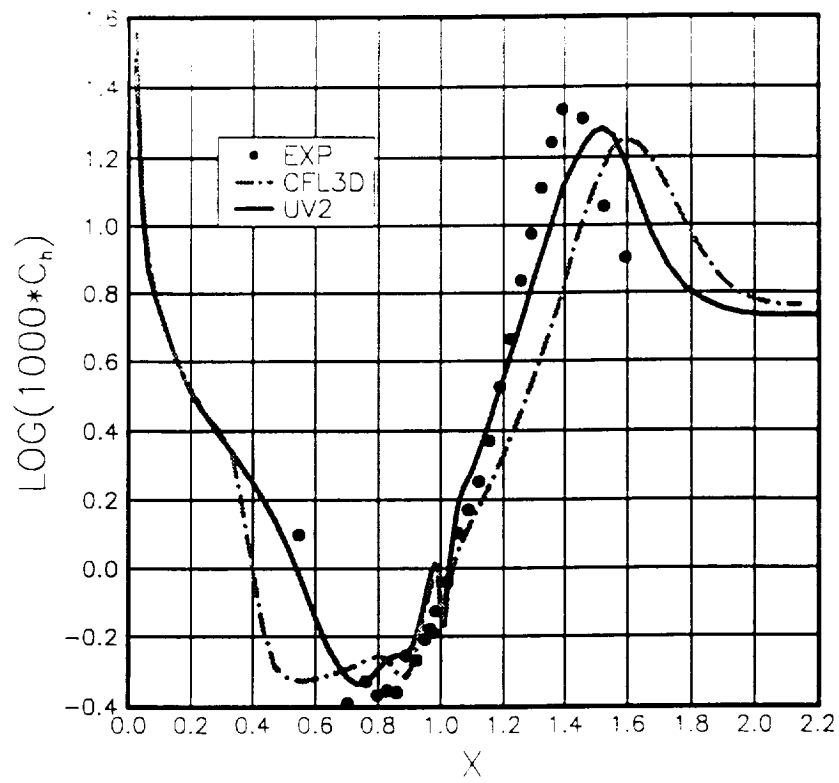


(a)

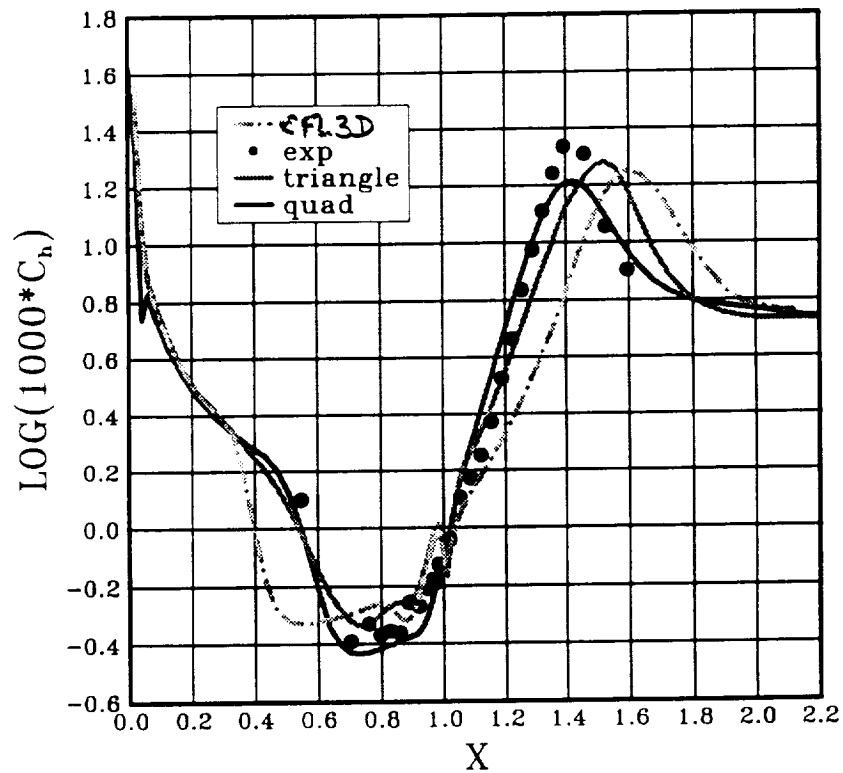


(b)

Figure 4



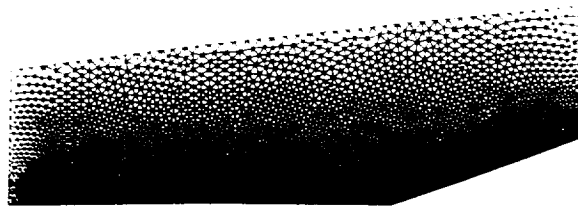
(c)



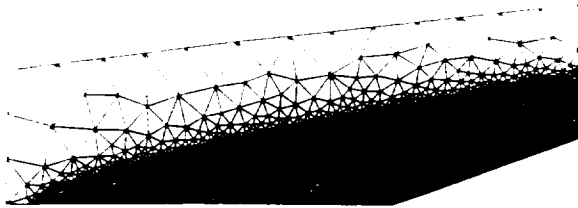
(d)

Figure 4 (continued)

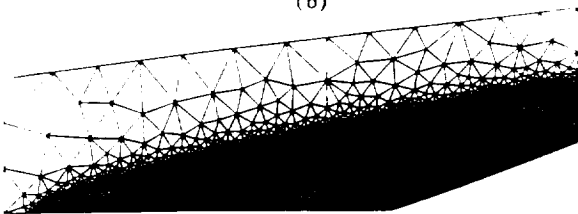
Mach 14.07 flow over a 24° compression corner showing the wall distributions of (a) pressure coefficient (b) skin friction coefficient (c) heat transfer coefficient computed on a 101 by 101 triangular mesh using a cell-vertex upwind scheme (d) heat transfer coefficient computed on a 101 by 101 quadrilateral mesh using a cell-vertex upwind scheme.



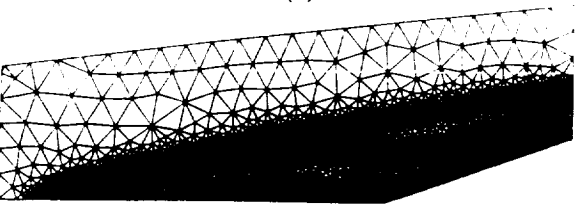
(a)



(b)



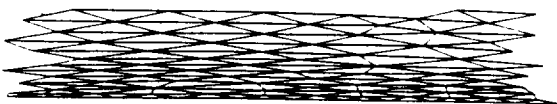
(c)



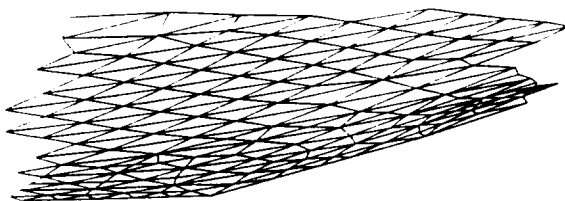
(d)

Figure 5

Flow over a 20° compression corner. $M = 10$; $Re = 143,800$. (a) the initial unstructured mesh and the mesh after the application of (b) adaptive remeshing (c) plus one level of mesh enrichment (d) plus mesh movement.



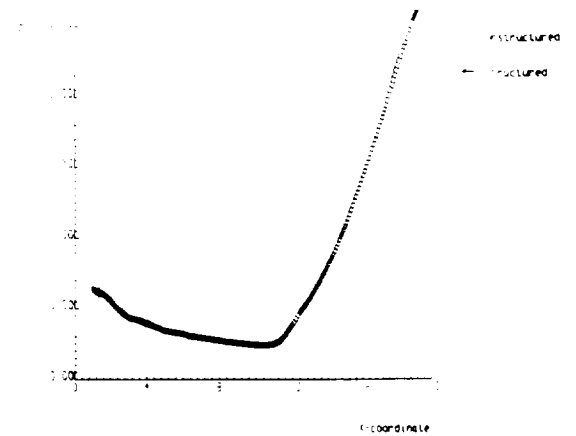
(a)



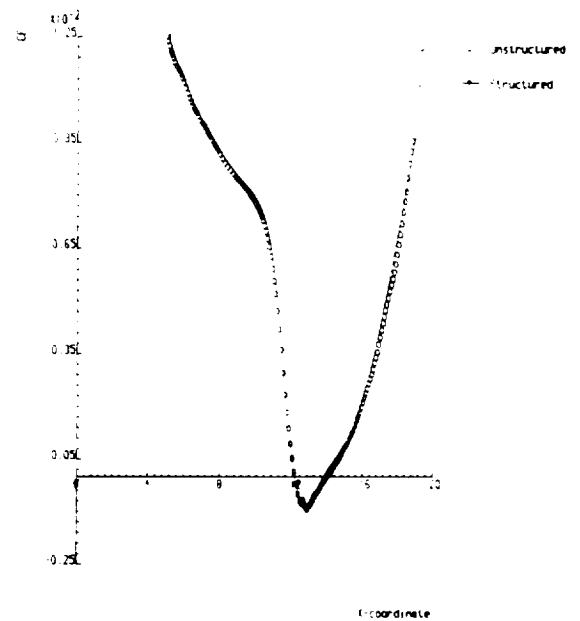
(b)

Figure 6

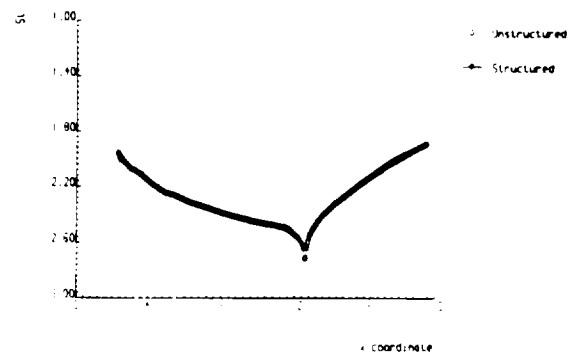
Flow over a 20° compression corner. $M = 10$; $Re = 143,800$. Details of the mesh of figure 9d in the vicinity of (a) a point on the plate and (b) the compression corner.



(a)



(b)



(c)

Figure 7

Flow over a 20° compression corner. $M = 10$; $Re = 143,800$. Wall distributions of (a) pressure coefficient, (b) skin friction coefficient and (c) Stanton number computed on the mesh of figure 9d compared with the structured mesh results of figure 3.

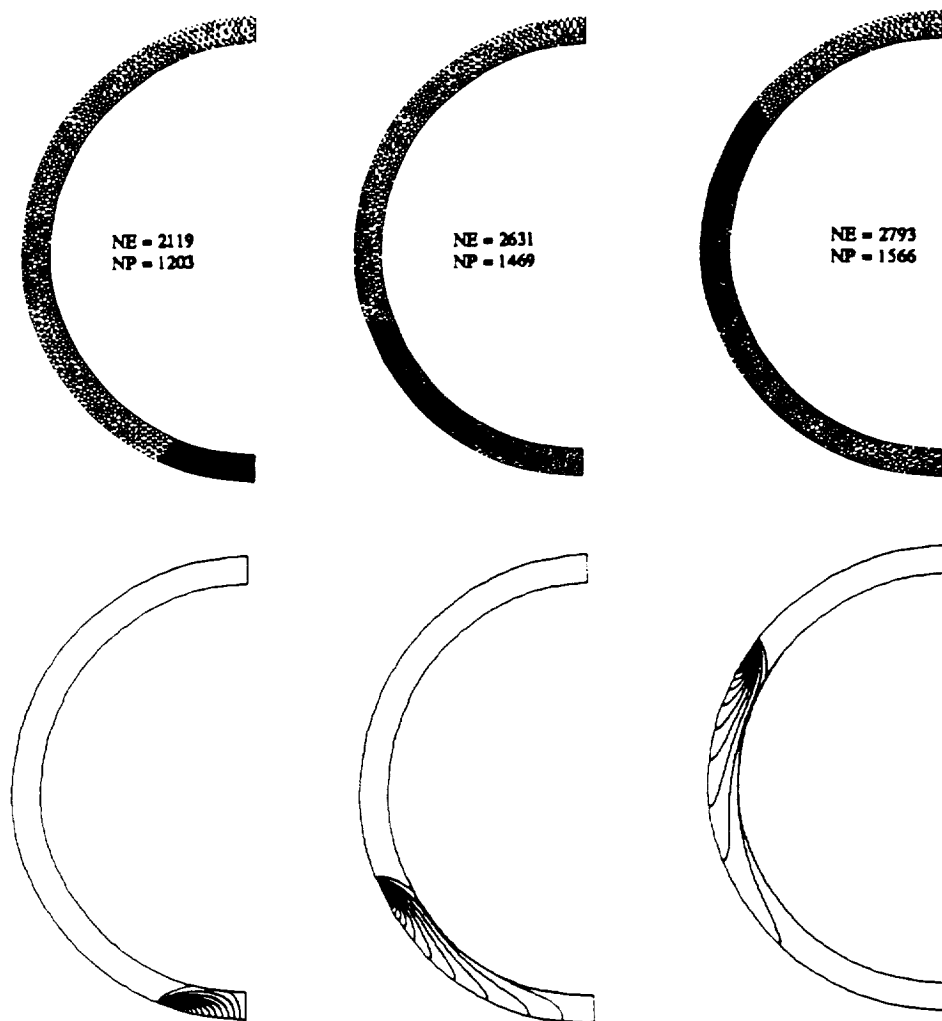
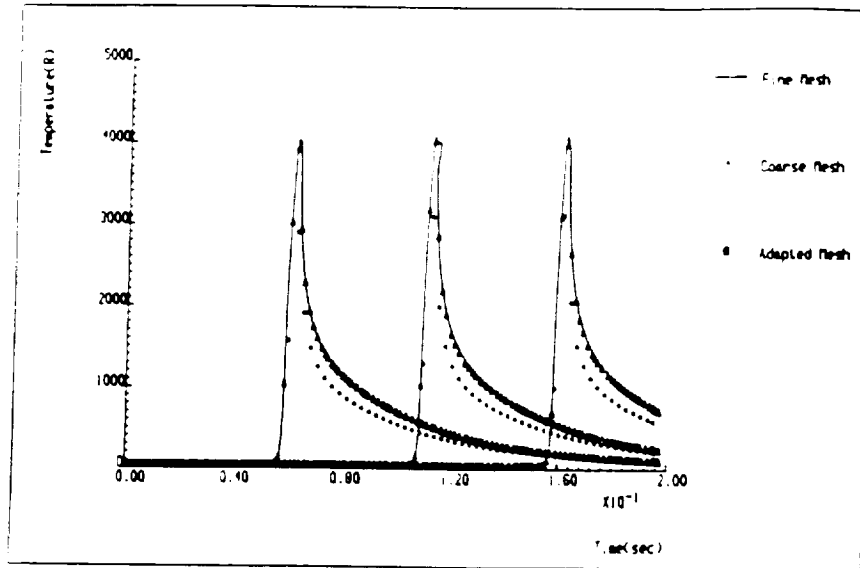
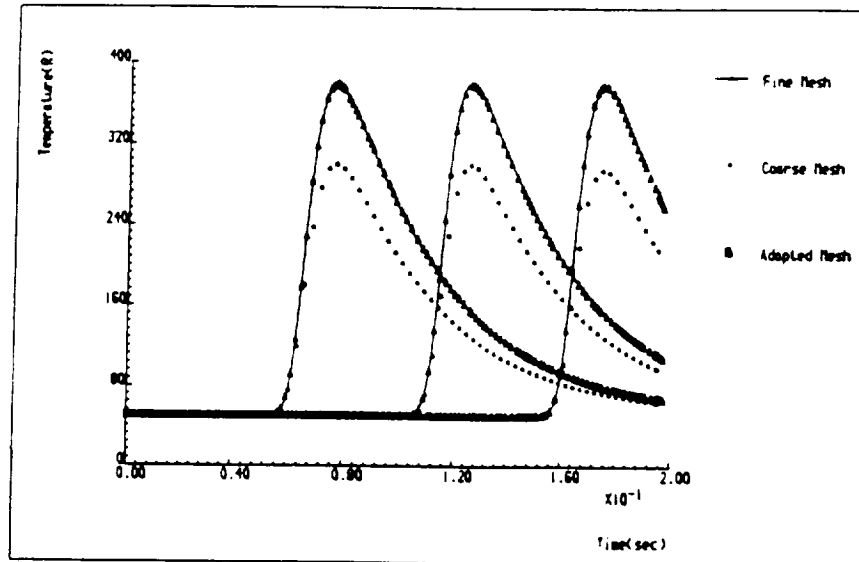


Figure 8

Computation of the temperature distribution produced by a moving heat source of constant magnitude showing the adapted mesh and corresponding temperature contours at three different times.



(a)



(b)

Figure 9

Comparison of the temperature distribution produced by a moving heat source of constant magnitude at the three monitoring points on (a) the external surface (b) the internal surface of the cylinder. Results obtained using a coarse mesh, fine mesh and adapted mesh.

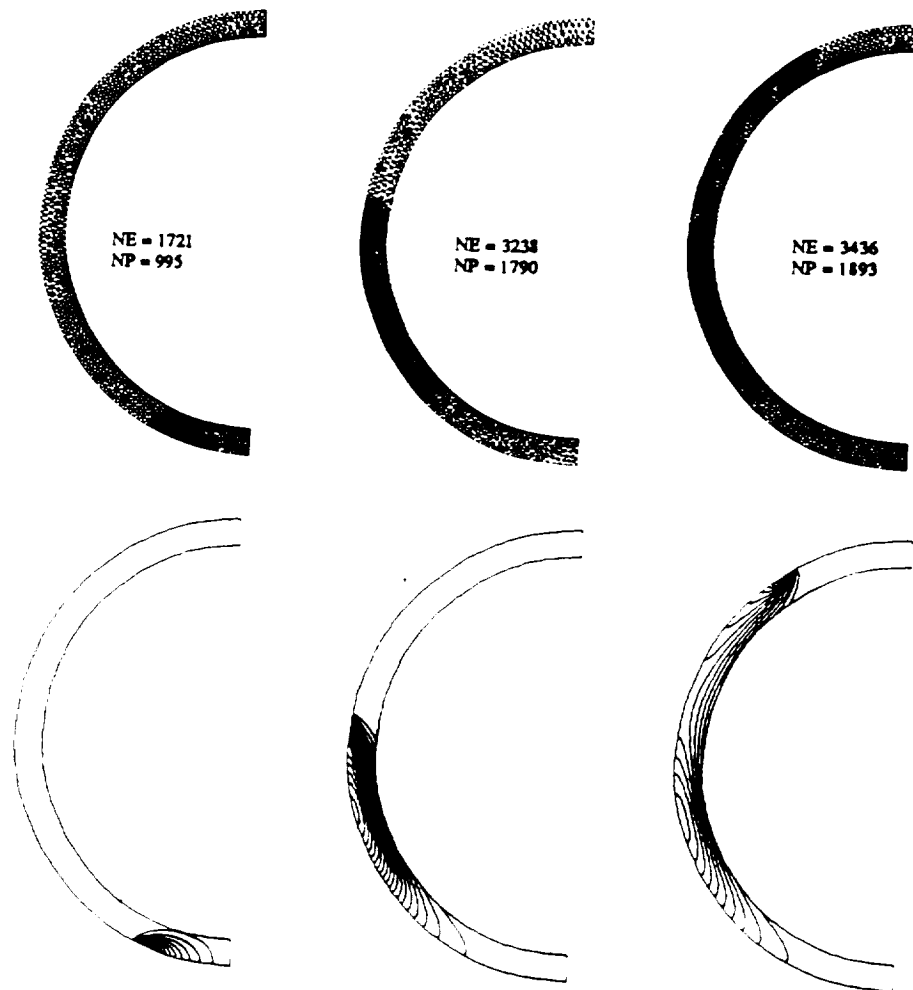
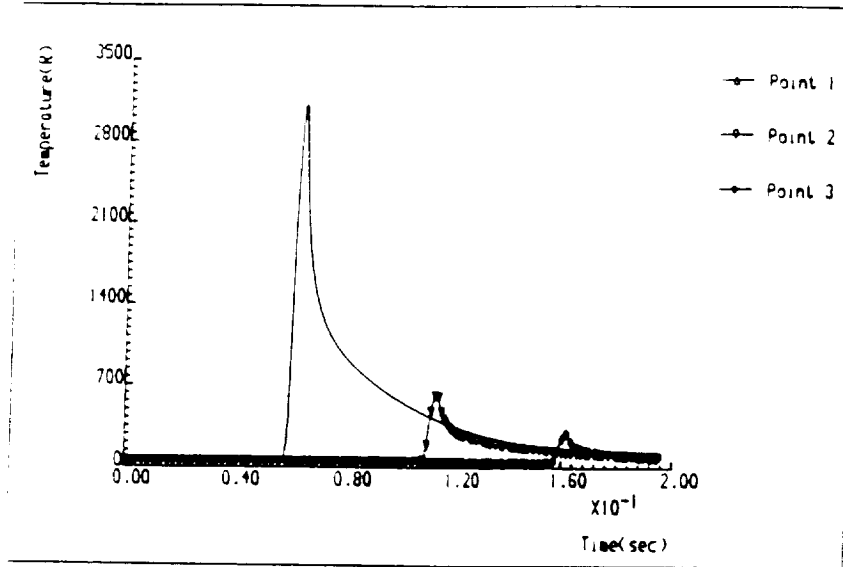
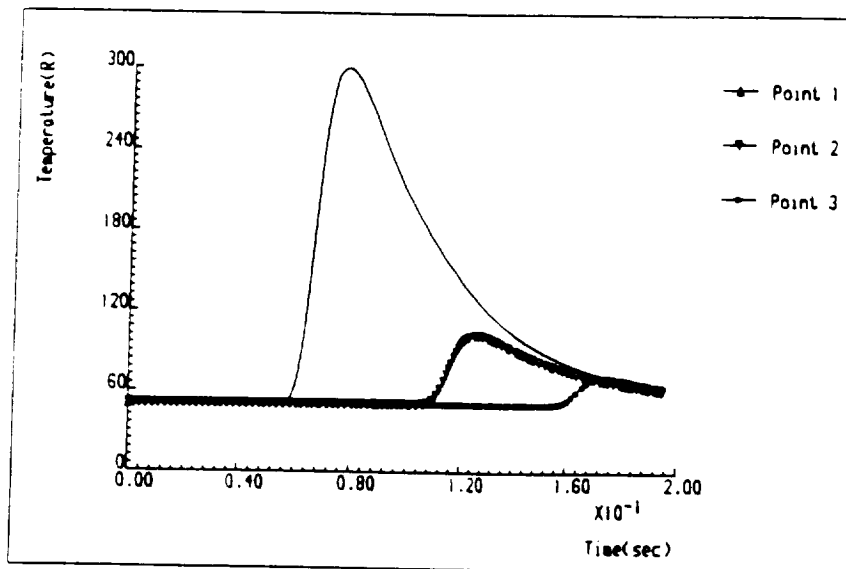


Figure 10

Computation of the temperature distribution produced by a moving heat source of varying magnitude showing the adapted mesh and corresponding temperature contours at three different times.



(a)



(b)

Figure //

Comparison of the temperature distribution produced by a moving heat source of varying magnitude at the three monitoring points on (a) the external surface (b) the internal surface of the cylinder. Results obtained using an adapted mesh.