Spring 1989

An Adaptive Remeshing Finite Element Method for High-Speed Compressible Flows Using Quadrilateral and Triangular Elements

Gururaja R. Vemaganti
Old Dominion University

AN ADAPTIVE REMESHING FINITE ELEMENT METHOD FOR HIGH SPEED COMPRESSIBLE FLOWS USING QUADRILATERAL AND TRIANGULAR ELEMENTS

by

Gururaja R. Vemaganti

B. E. May 1979, Osmania University
M.Tech. July 1981, Indian Institute of Technology
M.Engg. July 1984, Carlton University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

MECHANICAL ENGINEERING

OLD DOMINION UNIVERSITY
MAY, 1989

Approved by:

Dr. Earl A. Thornton (Director)

Dr. Allan R. Wieting

Dr. Sushil K. Chaturvedi

Dr. Ernst von Lavante

Dr. Jean Hou

ABSTRACT

AN ADAPTIVE REMESHING FINITE ELEMENT METHOD FOR HIGH SPEED COMPRESSIBLE FLOWS USING QUADRILATERAL AND TRIANGULAR ELEMENTS

Gururaja R. Vemaganti
Old Dominion University
Director: Dr. Earl A. Thornton

In this study a new adaptive remeshing method for high speed compressible flow analysis is presented. The method uses quadrilateral elements where possible, and triangles are introduced as needed. The primary goal of this study is to develop a remeshing method which uses both the concepts of unstructured and structured meshes for the finite element analysis to predict accurate aerodynamic heating in problems related to high speed viscous flows. The remeshing method uses a solution based on an old mesh to create a new mesh based on an advancing front technique. In the present implementation, a stuctured mesh of quadrilaterals is created in the boundary layer and an unstructured mesh consisting of quadrilaterals and triangles is created in the inviscid region. Studies of convergence rates show that for problems with highly localized solution variations, the remeshing approach gives smaller solution errors with fewer unknowns than refinement of uniform, structured meshes. Results for high speed compressible flows show that the remeshing method improves the solution quality significantly without necessarily increasing the number of unknowns in large proportions.

DEDICATION

This dissertation is dedicated to my mother Smt. V. Saraswathi and in memory of my late father Sri V. Ranganatha Rao whose love, affection and self-sacrifice are behind my every dream , success and pursuit of life.

ACKNOWLEDGEMENTS

America is about trying and an opportunity to try. I am thankful to all of the American people who gave me this opportunity to try something new. Especially, I am thankful to my advisor Dr. Earl A. Thornton who gave me an insight in a new direction and provided guidance, help and support throughout this research. His moral support in moments of despair and diffidence are gratefully acknowledged. I am grateful to the technical monitor of the ODU research grant NSG-1321, Dr. Allan R. Wieting, for both supporting this research and providing technical guidance and advice. I and other members in the finite element community appreciate Dr. Wieting's efforts and concern over making the finite element method a viable computational tool in the area of high speed flows.

Due regards and thanks to Drs. Ken Morgan and Jaime Peraire from University of Swansea, Wales, U.K. for their helpful discussions and recommendations in developing the new method. Thanks are also due for Dr. Surendra N. Tiwari, Dr. Earnst von Lavante, Dr. Jean Hou and Dr. Sushil K. Chaturvedi for readily agreeing to serve on both my guidance and dissertation committees. I am grateful for their helpful suggestions and comments.

I am thankful to Dr. Pramote Dechaumphai and Dr. Ramki Ramakrishnan for their helpful suggestions and comments and the former for providing the flow analysis program which is used in this research. I am thankful to my friends Dr. Ram Prabhu, Mr. Chris Glass, and Mrs. Cindy Albertson for their kind help

with the word-processor used in preparing this report. I am also thankful to all my friends and colleagues at the Aerothermal Loads Branch, NASA LaRC for providing a stimulating research environment.

Last but not the least, I am thankful to my wife Devi and our son Rishi for their patience, love and cooperation which they have shown me more than ever, during this research. Nevertheless without help from GOD, I could not have carried-out this research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF SYMBOLS

A	area of the domain, Jacobian matrix
B	constant, matrix associated with the viscous fluxes
BT	boundary term
C	constant
C_f	local skin friction coefficient
C_p	Coefficient of pressure
C_H	Coefficient of heat transfer
c_p	specific heat at constant pressure
c_v	specific heat at constant volume
E	interpolation error, left element
F	inviscid flux
G	viscous flux
h	dimension of an element, spacing
H	enthalpy
IC	inviscid contribution
k	thermal conductivity
L	length
M	number of elements, mass matrix, Mach number
N	element shape functions, number of sides
n	direction cosines

p	pressure
Pr	Prandtl number
Q	volumetric heat generation rate
q	heat transfer rate
R	position vector, gas constant, right element
Re	Reynolds number
S	stretching parameter
St	Stanton number
T	temperature
t	time
U	conservation variable
u	u-component of velocity
VC	viscous contribution
v	v-component of velocity
X	principal axis
x,y,z	coordinates of the reference frame

GREEK SYMBOLS

α	vector of stretching
δ	distance
ε	total energy
ϵ	eigenvalue limiter
γ	ratio of specific heats
Γ	edge of an element

λ	eigenvalue of the second derivative tensor
Λ	eigenvalue
μ	viscosity
ν	kinematic viscosity
θ	angle
ρ	density
σ	normal stress
τ	shear stress
Ω	area of an element
ϕ	function
Ψ	function

SUBSCRIPTS

a	average
c	centroid
i	node, element
E	exact, element
e	element, boundary layer edge
FE	finite element
K	node
L	lumped value
M	mid point
n	normal direction
o	reference value

SP	specified value
s	side
t	tangential direction
w	wall
x,y	coordinate axes
∞	free-stream
λ	eigenvalue
1,2	principal directions

SUPERSCRIPTS

*	current value, average value
m	time step index

Chapter 1

INTRODUCTION

Design, performance, and optimization studies for new aerospace vehicles require techniques which can calculate aerodynamic heating rates accurately and efficiently. Aerodynamic heating has a significant affect on the thermo-structural response of high speed flight vehicles. Typical flight vehicles that have significant aerothermal-structural interactions include hypersonic cruise vehicles, the space shuttle, and the National Aero-Space Plane.

Approximate analytical methods to predict aerodynamic heating are reviewed in [1]*. The approximate methods are applicable mainly for a perfect gas or equilibrium air and simple shapes of the body such as plates and cones. These approximate methods in general are not extendable to separated flows and flows with significant chemical nonequilibrium effects. Some of these analytical methods also have limitations on the geometry(e.g. 2-D geometry only), Mach number, Reynolds number of the flow etc.

* numbers in [ ] indicate references

1.1 Background

Numerical techniques such as finite difference, finite volume, and finite element methods play a significant role in fluid, thermal, and structural analyses. Numerical flow simulation has created a new discipline named "Computational Fluid Dynamics" (CFD). CFD has grown out of its infancy as an alternative to wind-tunnel testing for many flows of interest. Finite difference methods are the predominant numerical methods for Computational Fluid Dynamics. A major problem in finite difference CFD is the generation of grids for realistic three dimensional vehicles such as a finned projectile or a complete aircraft. The development of well-constructed CFD grids has been the subject of intensive research for several years. However, the generation of a single grid that discretizes the entire flow region for a complex configuration is an extremely difficult and sometimes an impossible task. These difficulties have led to recent research into alternative approaches for handling complex geometries including zonal schemes, the finite volume approach, and the finite element method.

However, finite difference and finite volume methods in many cases suffer from lack of mathematical rigor. There are no true error-estimates for the methods in many situations because of the fact that they are based on local Taylor series expansions, which are invalid in presence of singularities [2]. On the other hand, finite element methods are backed by mathematical theory and can be applied to arbitrary domains on unstructured grids. In addition finite elements often offer advantages in handling complex boundary conditions. For the solution of parabolic and elliptic partial differential equations, finite element error estimates have been established, and the finite element method possesses the best approximation property for this class of problems [3,4]. In addition, the finite element flow-field solution methodology can be easily coupled to existing finite element structural or heat transfer analysis codes within the same integrated database. The finite element method also avoids transformation techniques to map a real domain into a computational domain for a complex geometry and thus provides ease in data-processing in a multi-disciplinary environment. Use of unstructured grids gives the Finite Element Method(FEM) the capability to adapt to the deformation of a body during an analysis as well as to high transient gradients in the flow-field [5].

Research is underway in the Aerothermal Loads Branch at the NASA Langley Research Center to improve the capabilities and efficiency of finite element high-speed flow analysis methods and to develop more efficient integration of finite element fluid, thermal and structural analyses. The focus of the research is the prediction of aerothermal loads for complex three dimensional bodies. The research combines analyses with experimental studies. The aerothermal loads research includes aerodynamic heating on control surfaces, pronounced localized heating effects on wavy surfaces, localized heating in shuttle tile gaps and leading edge heating on scramjet fuel injection struts. Detailed aerothermal loads for a hypersonic vehicle are shown in Fig. 1, as described in [6].

For high speed viscous compressible flows, typical CFD algorithms including the finite element method require efficient mesh generation schemes since these flows involve very high gradient regions like thin boundary layers and sharp discontinuities like strong shocks. Fig. 2 [7] shows a regular refinement of a mesh requires more than 11,000 nodes for the sharp leading

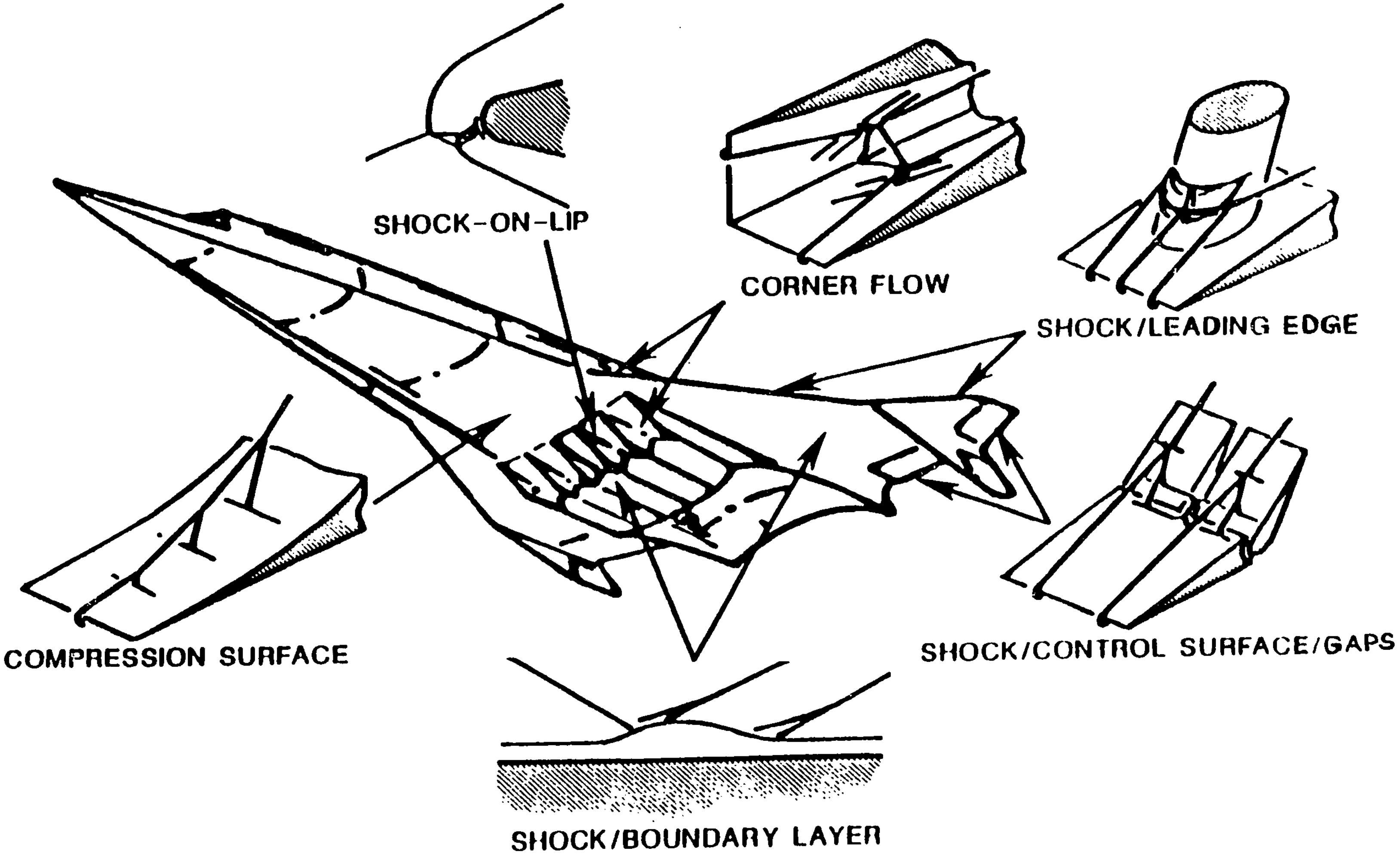


Fig. 1. Typical aerothermal loads inherent to hypersonic vehicles.

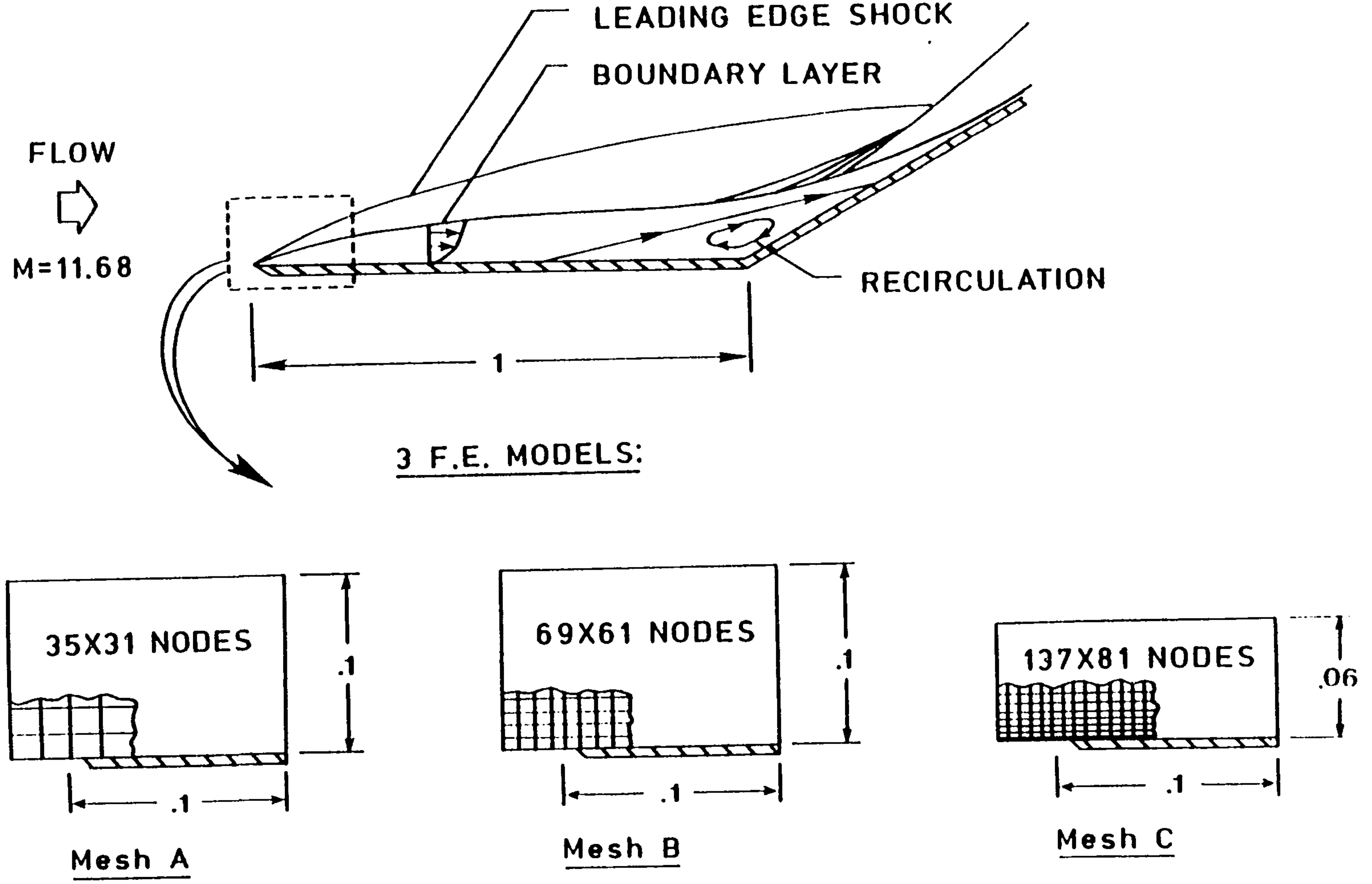


Fig. 2. Finite element meshes for the analysis of a sharp leading edge in Mach 11 flow.

edge analysis of a Mach 11 flow over a compression corner. A mesh having more than 15,000 nodes is required in the compression corner region to capture all the flow features. A regular refinement thus becomes enormously expensive for high-speed flows. Adaptive refinement based on a previous solution is an alternative that is efficient and computationally economical in this context. Adaptive refinement reduces the number of grid points relative to a uniformly refined mesh because it refines the mesh where the refinement is required.

1.2 Research Goals and Objectives

A majority of research in the area of adaptive mesh refinement is confined to structured meshes(literature survey in this field is given in Chapter 2). But structured meshes have certain limitations like they are suitable for fairly regular geometries and transition between a refined region and an unrefined region may not be smooth. On the other hand unstructured meshes can handle the transition between a refined region and a relatively crude region easily and are applicable to any irregular geometry. However, less information is available on unstructured grids in the literature. Error estimates and convergence rate studies are not performed on the unstructured grids as extensively as they are performed on the structured grids. The full potential of unstructured meshes in terms of geometric flexibility, control over mesh density(number of nodes per unit area) is yet to be fully explored.

Nevertheless, structured meshes excel in boundary layers where flow gradients are predominant in the the transverse direction of the flow and elements that are stretched in the streamwise direction of the flow are required.

Structured meshes can easily accommodate the high aspect ratio(of the order of 1000) elements that are needed to optimize the number of elements in the boundary layer region. These considerations suggest that a combination of structured and unstructured meshes may be suitable for high-speed flow applications. Thus, there is a strong motivation for developing mesh generation schemes that make use of both the concepts of structured and unstructured meshes.

The goal of this dissertation is to investigate adaptive finite element mesh generation methods for high speed viscous compressible flows. Ultimately, the adaptive mesh generation method when combined with an effective solution algorithm will provide an efficient, reliable approach for predicting detailed aerothermal loads for problems with complex geometries and boundary conditions.

Specific objectives of the research are to:

(1) Develop an algorithm for two-dimensional adaptive mesh generation using the concepts of structured and unstructured meshes such that the number of unknowns is a minimum.

(2) Validate the code on some simple elliptic problems with exact solutions where error-norms can be computed based on the exact solution.

(3) Study applications of the code for hyperbolic problems with known analytical solutions to examine the convergence rates.

(4) Investigate different error-indicators for high-speed viscous flows in order to predict accurate aerodynamic heating and optimize the number of grid points.

(5) Apply the mesh generation code in association with existing analysis programs to solve practical hypersonic viscous flow problems.

(6) Investigate the feasibility of extending the adaptive method to three-dimensions.

1.3 Scope of the Dissertation

After the background, purpose, and goals of the research are introduced in this chapter, recent trends in finite element mesh generation are presented in the second chapter. Recent developments in the automatic mesh generation followed by the conventional adaptive refinement techniques are presented in Chapter Two. At its end , the concept of remeshing is introduced and potential benefits of the method are cited.

In chapter 3, the remeshing method is described. Adaptivity of the method to a previous solution is described first, followed by a discussion on different stages of remeshing. This chapter concludes with the concept of "structured remeshing" which is desirable for discretizing boundary layers. In the fourth chapter, convergence studies of the new method on some elliptic problems are presented. Comparative results with a uniform mesh refinement for these problems are tabulated.

The governing equations followed by the description of the numerical algorithm for the compressible flows are presented in Chapter 5. The numerical algorithm is based on an upwinding method using a cell-centered scheme. Application of the remeshing method to problems in high speed viscous flows which include typical flow complexities is described in Chapter 6. Results showing improvement in the solution quality through successive remeshing are presented in the sixth chapter. Conclusions are drawn based on this research and recommendations for future work in this area are presented in Chapter 7.

Chapter 2

TRENDS IN FINITE ELEMENT MESH GENERATION

The generation of high quality finite element models is vital for effective finite element analysis. The generation of optimal meshes to achieve the solution accuracy is always a challenge for the analysts. Though automatic mesh generators are commercially available (e.g. PATRAN), they are not efficient for high speed flow applications since these generators do not take the flow features into consideration. To resolve strictly one-dimensional features like shocks and boundary layers in high speed flows, mesh refinement at the location of the flow feature is desirable to obtain an optimal mesh. Adaptive strategies based on *a posteriori* error estimates have been found to be very effective in identifying the location of the flow features with strong gradients and the direction in which they occur. In this chapter, automatic mesh generation techniques are discussed first. Then adaptive strategies and conventional mesh refinement techniques are discussed focusing on current research in this area. Finally the link between automatic mesh generation and adaptive mesh control is brought out in light of the evaluation of adaptive remeshing method which is the core of this research.

2.1 AUTOMATIC MESH GENERATION

An automatic mesh generator is an algorithmic procedure capable of producing an optimal finite element mesh in a domain of arbitrary complexity given no input past the computerized geometric representation of the domain to be meshed. Earlier automatic mesh generators used the application of mapping techniques for generating structured meshes. Zienkiewicz et al. [8] used the concept of the isoparamatric element to generate well-controlled meshes within individual cells called as "patches" in a domain. Each "patch" is treated like an isoparamatric element for further subdivision. PATRAN, a commercially available mesh generation program uses the same "patch" concept to automatically generate two-dimensional and three-dimensional meshes. However, reducing a general three-dimensional domain into a set of mappable regions (patches) which gives the desired mesh is complicated.

Recently efforts have been made to generate unstructured meshes automatically, avoiding the definition of patches and with only the boundary of the domain being defined. These unstructured meshes in general consist of simplex element topologies, i.e. triangles in two-dimensions and tetrahedra in three-dimensions. Cavendish et al. [9,10] and Watson [11] used the approach of node generation followed by element creation to generate unstructured meshes. In this approach, nodes are generated first in the entire domain based on a specified mesh density and elements are created based on the concept of "Delaunay triangulation" [12]. In this concept a new element is generated out of the available nodes such that no other node in the domain exists within this element and the sides of the new element do not intersect the sides of any other existing element. Although the concept of Delaunay triangulation is extendable to three-dimensions, there is no guarantee that the resulting elements will have a satisfactory shape in terms of the volume to surface area. Since the method always tries to generate elements close to equilateral triangles in the computational domain, it is not quite suitable to resolve solution features which show a strong directionality because stretched elements are desired in the neighborhood of these flow features. Lee et al. [13] proposed another scheme which uses the concepts of constructive solid geometry. The geometric definition of the object and a value for the required mesh density are the only inputs in this scheme.

The advancing frontal technique proposed by Lo [14] follows a different approach of simultaneous node generation and element creation. In this approach the front defines the chain of line segments that surround the domain to be discretized. The front changes as the mesh construction proceeds and vanishes when the mesh generation is complete. Joe and Simpson [15] reduce a two-dimensional domain into simply connected regions by the same procedure and then triangulate each of these sub-domains. Shephard and Yerri and others [16,17] use another approach of decomposing the domain of interest into a set of simple cells and to then mesh the individual cells in such a manner that the resulting mesh is optimal. The quadtree and the octree mesh generators in two and three dimensions developed by Shephard et al. [18] follow this approach.

In addition to the ability to generate a valid mesh for any geometry, automatic mesh generators must permit mesh gradations necessary to produce efficient and accurate finite element models. Ideally, the mesh control criteria available allow for the convenient specification of both *a priori* and *a posteriori* mesh control information. *a priori* mesh control devices are used to specify the distribution of elements in the initial finite element model, while *a posteriori* mesh control devices are used during an "adaptive analysis" process to improve the mesh as dictated by the results on the current mesh. The ease with which particular forms of mesh control can be exercised depends both on the mesh generation and its implementation. Since the basic input to an automatic mesh generator is a geometric representation, any *a priori* mesh control device must be tied to the geometric representation. A general method to define mesh control information is to tie this information to the model through the topological entities such as vertices, edges, faces, and regions in the boundary representation of the object.

One major limitation of any *a priori* control device is that it is independent of the finite element solution. In problems related to high speed flows, steep gradients in flow variables may occur even on fairly regular geometries and *a priori* knowledge regarding the location of these gradients may not be available in many situations. Therefore indicators based on an earlier solution are required to identify regions where refinement is necessary. So an *a posteriori* control or an adaptive mesh refinement is a better suited approach, since it takes the solution into account. Oden [19] has proven that such a control is computationally more economical than *a priori* control of the meshes.

2.2 ADAPTIVE MESH REFINEMENT

In an adaptive finite element analysis, the solution on a given mesh in combination with knowledge of the mesh, are used to estimate the accuracy of the solution and modify the mesh to improve the solution accuracy.

The major components of such an analysis include:

(1) Finite element solution algorithms,

(2) *a posteriori* error estimation techniques

(3) error indicators to determine where and how to improve the finite element discretization, and

(4) mesh generation schemes to produce the finite element discretization as suggested by the error indicators.

Since this research is mainly related to the last two components of an adaptive finite element analysis approach, the following discussion is mainly focused on these aspects. Since application of the adaptive procedure is for high-speed flows, only references in this area are cited. Though efforts were made to survey the entire research in this field, the author feels the survey is neither exhaustive nor covers every work.

Adaptive mesh refinement methods mainly fall in to four categories:

(1) "h-refinement" where a mesh is refined and/or derefined when local errors fall outside preassigned upper and lower bounds,

(2) "p-refinement" where the order of the polynomial is increased when the error is larger than a preassigned value,

(3) "r-refinement" where the positions of the nodes are relocated with in a given mesh topology so as to equi-distribute error and

(4) combination of any of these methods.

Each method has its advantages and disadvantages with respect to the quality of the solution, computational efficiency, ability to capture features of the flow accurately and the overall modelling which are discussed below.

2.2.1 h-refinement method

This is one of the most commonly used methods where more elements of the same type are introduced by subdividing elements of the original mesh.

This is done in an adaptive manner where the selection of elements to be divided is based on a local error estimation. Lohner et al. [20] have employed this method using linear triangular elements for high-speed inviscid flows. Linear triangles are especially suitable since they do not create new nodes on the element edges during refinement. Though the method is computationally efficient and improves the solution, it increases the number of unknowns from refinement to refinement. Also the method lacks the directional refinement required for resolving one-dimensional features. Oden et al. [21,22] employed a similar concept using quadrilateral elements, but used a derefinement strategy in association with mesh enrichment by subdivision. Though the method has some control over the increase in the number of unknowns, quadrilateral elements give rise to midside nodes during refinement. One way to handle the midside nodes is to employ constraint equations which average nodal values at the end nodes to obtain the variables at the midside nodes. Ramakrishnan and Thornton [23] used an alternate approach to avoid midside nodes by transitioning from a crude quadrilateral mesh to a fine one using triangular elements.

Shapiro and Murman [24] used a similar refinement/derefinement method as Oden et al. but introduced a directional adaptation to it, by either dividing an element into four smaller elements or two elements along either horizontal or vertical direction. The three "h-refinement" schemes are shown in Fig. 3 for supersonic flow over a compression corner.

The "h-refinement" method mainly lacks directional refinement near the one-dimensional flow features and increases the number of unknowns from refinement to refinement which are major considerations for a

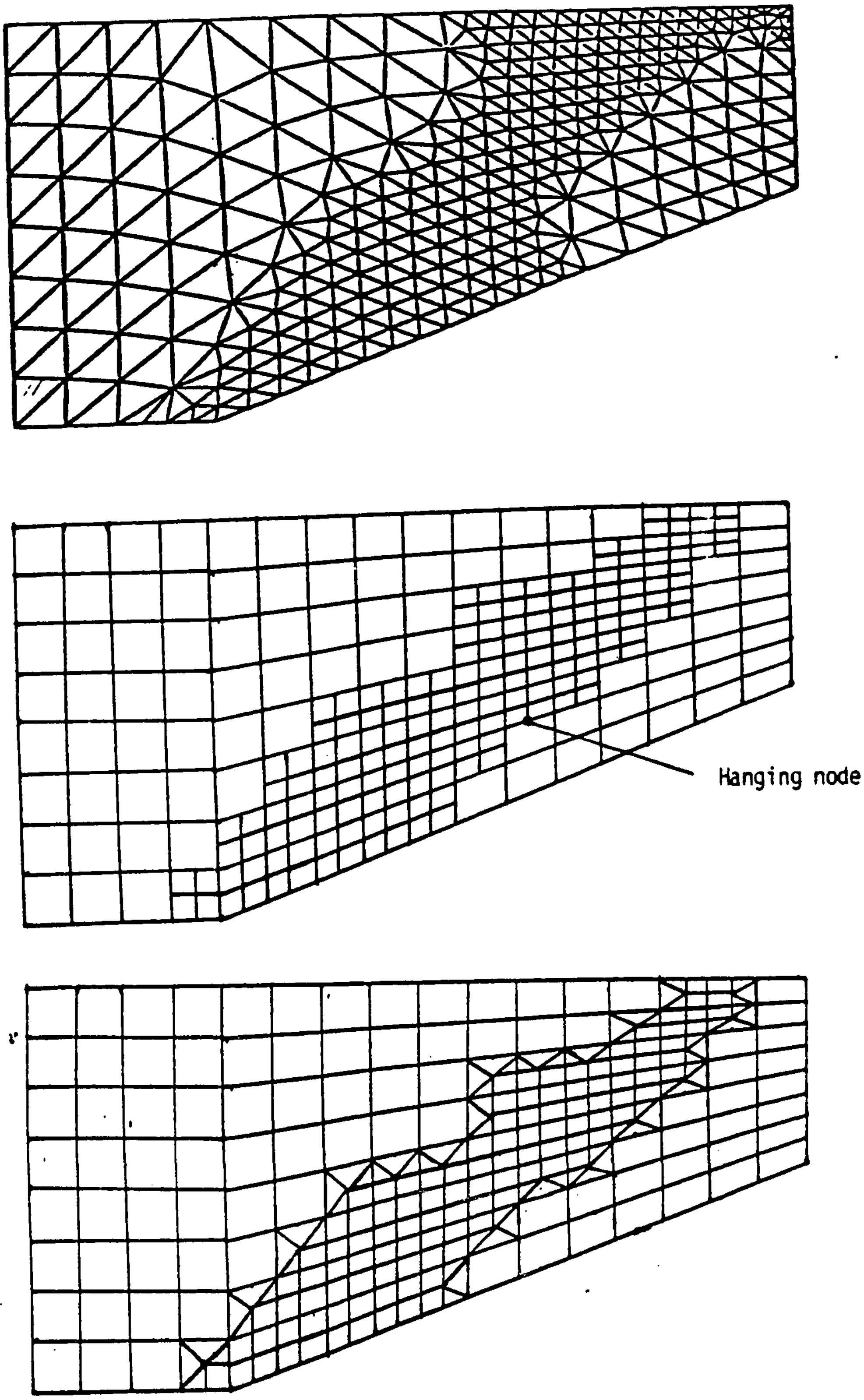


Fig. 3. "h - refinement" methods for a supersonic flow over a compression corner. [ref.23]

complex flow field. The original location of the node points remain at the same place in this method, though this limitation can be avoided by combining the method with "r-refinement" method which is discussed later.

2.2.2 p-refinement method

The "p-method" in which the local degree of the polynomial shape function is increased where the error is large, is closely related to spectral element methods. The "p" method has been applied to incompressible, viscous steady flow by Patera [25] and Devloo [27]. Shapiro and Murman [26] used a biquadratic element in a Galerkin formulation using biquadratic interpolation functions. They faced certain problems related to smoothing near discontinuities. It was also difficult for them to compute the stability limit and they found that these higher-order elements are unstable for certain algorithms. Devloo [27] et al. proposed a combination of "h" and "p" methods of refinement for high-speed flows, where the mesh is refined to a specified level in both "h" and "p" parameters in the boundary layer and a "h" refinement performed in the inviscid region where discontinuities like shocks exist. The algorithm associated with the "p" method must work for general orders of the polynomial and some current flow algorithms do not have this capability. Implementation of the "p" method results in major changes in the analysis codes and this method is still under development. A combination of "h" and "p" methods suggested by Oden et al. applied to supersonic viscous flow over a flat plate is shown in Fig. 4.

2.2.3 r-refinement method

The "r-method", also called the relocation method or mesh movement method, is based on some node moving criteria and is being successfully

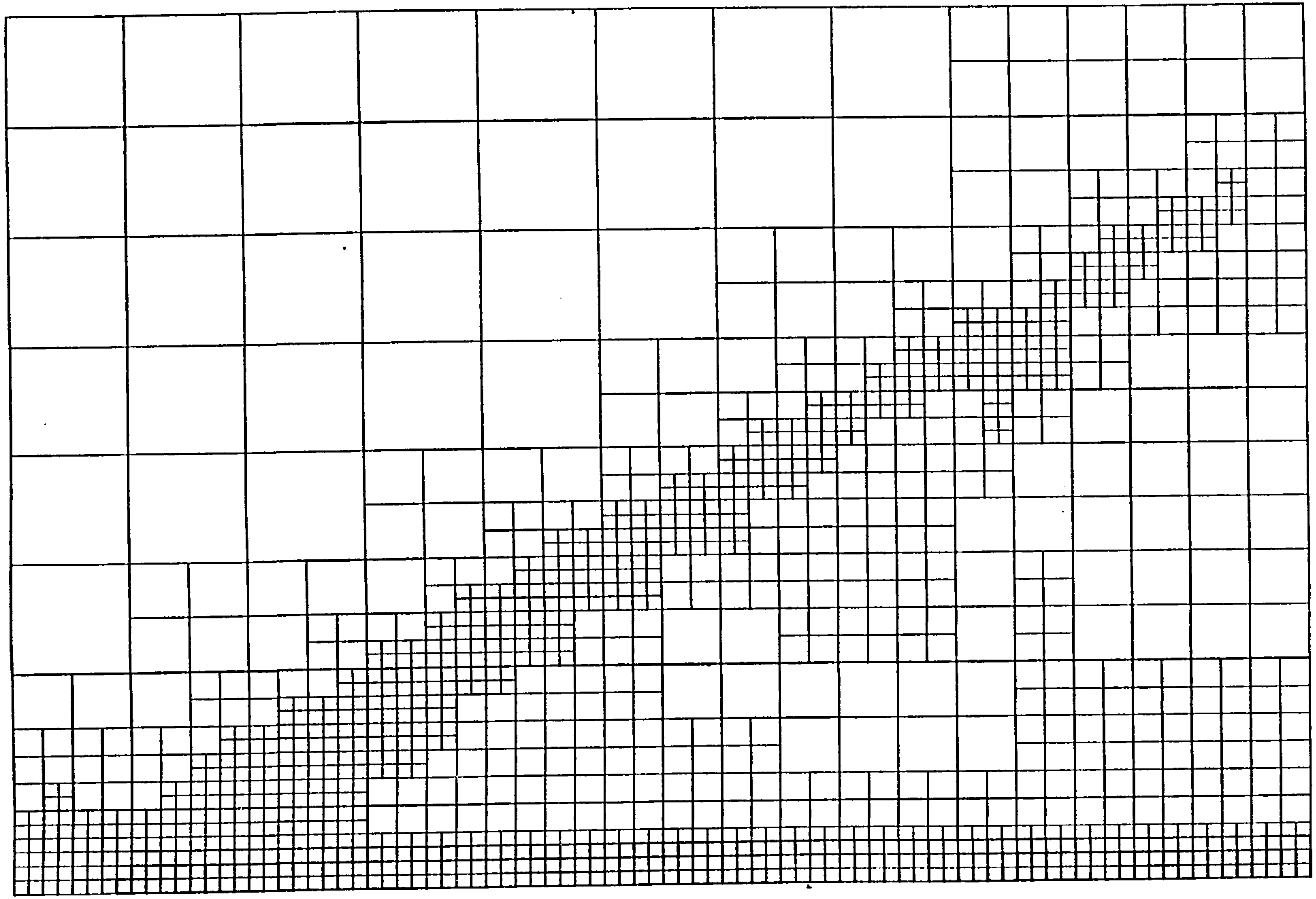
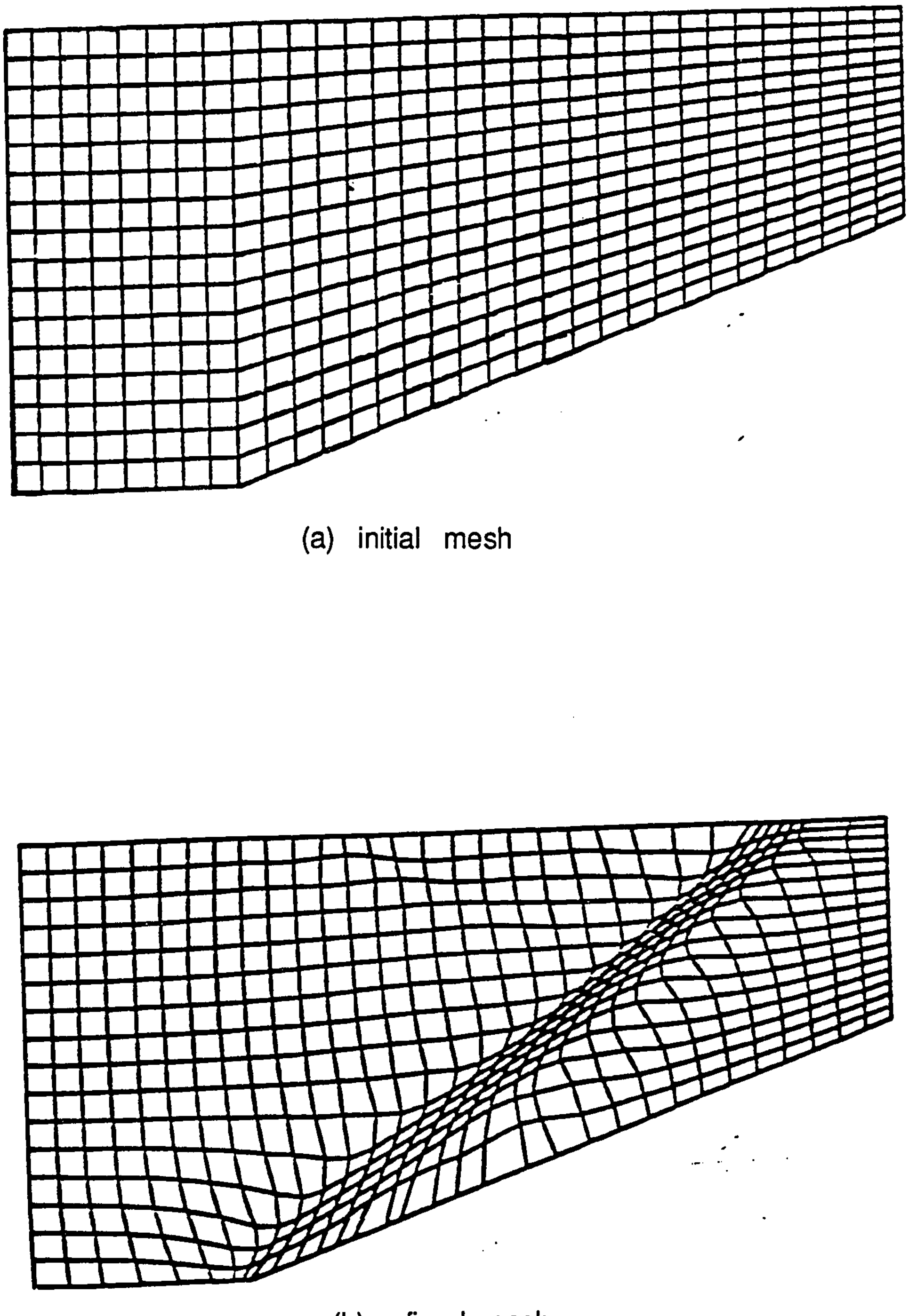


Fig. 4. "p - refinement" method applied to a supersonic viscous flow over a flat plate. [ref.27]

used for the solutions to nonlinear parabolic and hyperbolic problems. A recent review on this method is found in [28]. This method is also common in finite difference and finite volume schemes. Eiseman and Erlebacher [29] gave an extensive review of this method in finite difference and finite volume applications where structured grids are adapted to a previous solution by moving the nodes with respect to a weight function. Erlebacher [30] implemented the method for unstructured triangular meshes for solving Euler equations. Abolhassani [31] implemented the method to adapt finite difference grids dynamically for high speed flow applications.

Lohner, Morgan, and Zienkiewicz [32] applied the method in finite elements for solving the compressible Navier-Stokes equations. In this work, the sides of an element are treated as a system of springs which exert a force proportional to the length of each spring . The stiffness coefficients are defined through some weight functions derived from the local error. The mesh is then moved in such a way that the resultant of the spring-forces at each node vanishes and the equivalent spring system is thus in equilibrium. The method sometimes gave rise to badly distorted elements, and they tried to eliminate this distortion by flipping the diagonals of a quadrilateral which would result by combining any two bad shaped triangles. Oden [22] implemented a similar method for quadrilateral elements using the principle of equi-distribution of the error. The method applied to a supersonic flow over a compression corner is shown in Fig. 5. Oden [22] found in some instances, a deterioration in the solution quality after four adaptations due to the badly graded mesh produced from the oscillations of the adaptive scheme. Though the "r" method takes the least computational time compared to any other refinement method, and is suitable for transient analysis, it has certain



(a) initial mesh

(b) refined mesh

Fig. 5. "r - refinement" method applied to a supersonic flow over a compression corner. [ref.22] limitations. There is a limit on the solution accuracy possible because the number of elements does not change during refinement. The method also requires special care to maintain the shape of the elements and numerical stability of the meshes as the nodes move. These limitations become more critical as the dimensionality of the problem increases.

2.2.4 Combination of "h","p" and "r" methods

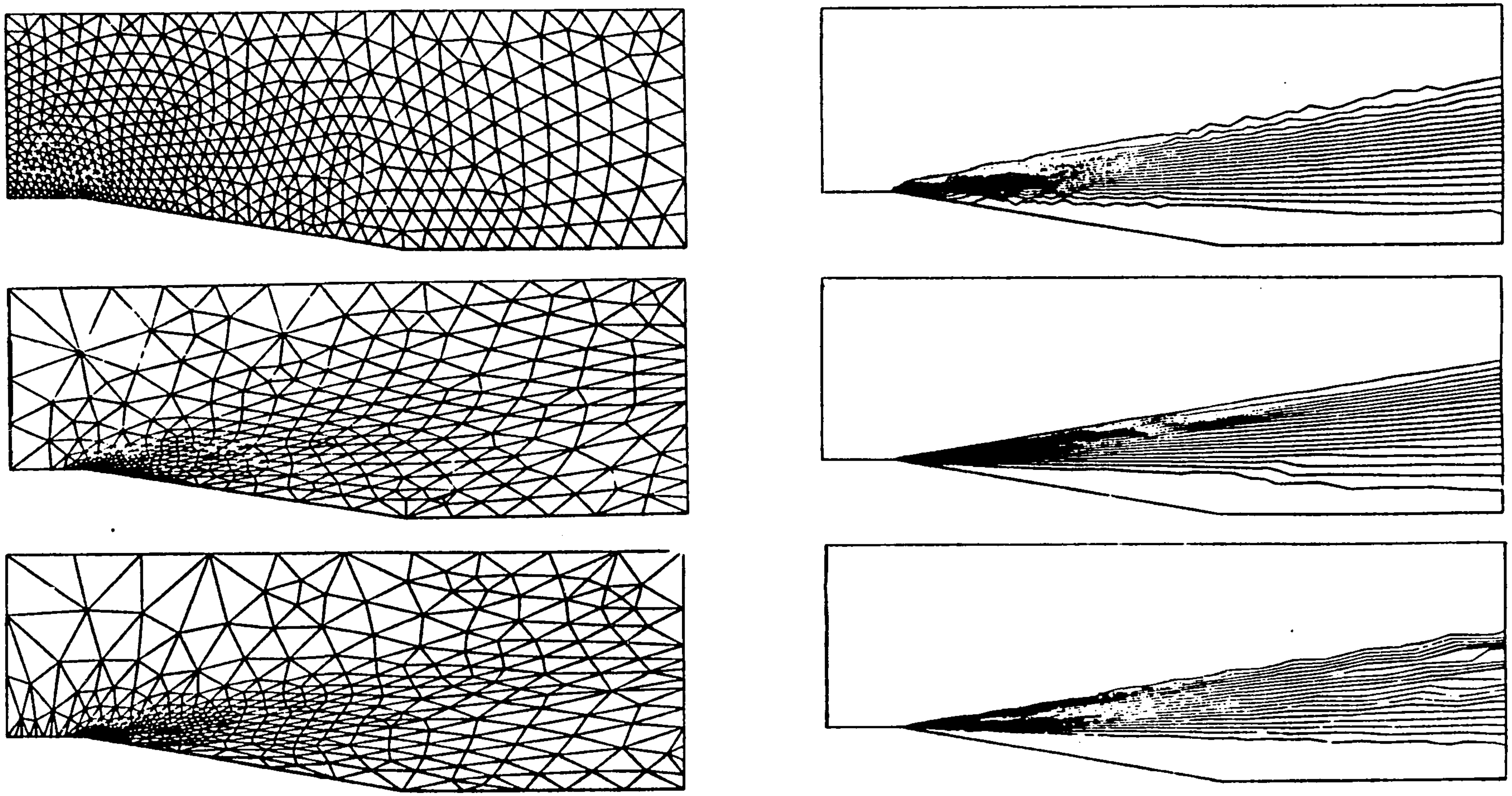
As mentioned earlier, Devloo [27] combined the "h" and "p" methods for compressible viscous flows and research is still underway. Shapiro and Murman's [26] implementation of biquadratic element in association with the regular bilinear elements can be classified under the same category. Ramakrishnan and Thornton [33] combined the "r" and "h" methods for high-speed inviscid flows and improved their solution quality obtained by the "h" method alone.

In the case of elliptic problems with singularities present, it has been shown [34] that proper combination of "h" and "p" methods is an extremely efficient combination and optimal "h-p" refinement can give exponential rates of convergence. Such research related to parabolic and hyperbolic problems is yet to be initiated.

2.3 ADAPTIVE REMESHING

All the mesh refinement methods discussed above try to improve the original mesh. Usually these methods start from a uniform mesh which is crude and try to refine it. For complex geometries, creating an initial mesh is not trivial and refining these meshes may not give optimal meshes within three or four adaptations. Combining the mesh refinement procedure directly with the functionality of an automatic mesh generator results in an improvement in the mesh's approximation to the domain being analyzed and an improvement of the mesh within the domain to increase the quality of the finite element solution. A feedback approach to the development of improved finite element meshes is to use the results on the current mesh to guide the generation of an entirely new mesh. This approach is referred to as "adaptive remeshing". In this approach *a posteriori* parameters are converted into *a priori* parameters and used for the development of a new mesh.

One technique that has been developed [35] plotted contours of a specific solution parameter e.g. stress contours in structures problems that gave an indication of how the mesh should be distributed and then allowed the user to generate a new mesh interactively that followed these contours. This approach is not practical to solve problems related to high speed flows because of the complexity of the solution features involved. Peraire and Morgan [36] developed a new adaptive remeshing approach using linear triangles mainly designed for high speed flow problems. The method regenerates entirely a new mesh adaptively and accommodates stretching of the elements along shocks and boundary layers so as to align the elements along these one-dimensional features. The method has been successfully employed in solving several inviscid problems. The method applied to analyze an expansion fan is shown in Fig. 6. It was also found that the number of degrees of freedom on the final mesh is not significantly larger than the number used in the original mesh. Peraire et al. recently [37] extended this approach to three-dimensions generating meshes consisting of tetrahedron elements.



(a) Three successive meshes

(b) Density distributions

Fig. 6. Adaptive remeshing applied to the supersonic flow around an expansion corner. [ref.36]

Out of all the methods discussed above, the most promising approaches are the adaptive refinement/derefinement method and the adaptive remeshing method. The adaptive remeshing method seems to be very effective in solving inviscid problems. In three dimensions, this method created meshes with high element to node ratios (of the order of 5 to 6) , that is 5 to 6 elements were associated with each node in the mesh. Instead, if a mixture of hexahedron and tetrahedron elements is used to generate three-dimensional meshes, such a method would reduce the element to node ratio and reduce the total number of elements relative to an all tetrahedra mesh. Extension of the remeshing method to boundary layers using structured layers of elements is to be investigated. For this extension, the boundary layer edge is to be identified in an adaptive manner.

A new remeshing method that uses the concepts of structured and unstructured meshes for the high speed viscous flows is presented in the next chapter. In this method the boundary layer edge is determined from an earlier solution and layers of quadrilaterals are created in the boundary layer and the inviscid region is discretized in an unstructured manner. The new method uses a mixture of quadrilateral and triangular elements in the inviscid region and quadrilateral elements in the viscous region.

Chapter 3

ADAPTIVE REMESHING USING QUADRILATERAL AND TRIANGULAR ELEMENTS

The adaptive remeshing method developed by Morgan and Peraire [36] that creates unstructured meshes consisting of linear triangles proved to be very effective in inviscid flows. But using a combination of quadrilateral and triangular elements may reduce the number of elements compared to using triangles alone. Thus a combination of quadrilateral and triangular elements may result in less computer storage requirements and processing time. These savings may become much more significant in three dimensions. In boundary layers the gradients of the flow variables dominate in the transverse direction of the flow compared to the streamwise direction. Hence an optimal mesh in the boundary layer will consist of elements with a larger dimension in the streamwise direction and a smaller dimension in the transverse direction (high aspect ratio elements). It is very difficult to generate unstructured meshes with such elements. Even if such meshes are created [38] they contain highly distorted triangles with large aspect ratios and large variation in the internal angles. A structured mesh with highly stretched quadrilateral elements is desirable in boundary layers. So there is a need to combine the concepts of both structured and unstructured meshes in the context of adaptive remeshing applied to high speed viscous flows. Such a combination may be suitable for a "h-p" refinement algorithm too.

This chapter describes a new remeshing method which uses predominantly quadrilateral elements. The method uses quadrilateral elements where possible, and triangles are introduced as needed. The remeshing method at each stage uses the solution based on a previous mesh to generate a new mesh. Unstructured meshes are created except for high speed viscous flows where a structured mesh of quadrilateral elements is created in the boundary layer and an unstructured mesh consisting of quadrilateral and triangular elements is created in the inviscid region .

The concept of adaptive remeshing is discussed first followed by a discussion of remeshing with a mixture of quadrilateral ("quads") and triangular elements. Later, element generation based on an advancing front technique is described. Finally the concept of structured remeshing using quadrilateral elements and the integration with unstructured remeshing is presented. A simple procedure to smooth the meshes based on "Laplace filter" suggested by Baehmann et al. [39] concludes the chapter.

3.1 Adaptive Remeshing

3.1.1 Concept of remeshing

The main idea of remeshing lies in generating a completely new mesh based on solution information from a previous mesh. This information takes the form of mesh generation parameters computed on the previous mesh at all the nodal points. The method becomes adaptive when these parameters are computed from a numerical solution on the previous mesh. The mesh generation process produces smaller elements where refinement is required and provides smooth transition from a high resolution region to a low resolution region through triangles. This method also introduces stretching of the elements along strictly one dimensional features like shocks which is highly desirable for high speed flows. Proper clustering of the elements near regions of high gradients is achieved in the method without refining existing elements. Since the remeshing method is a combination of automatic mesh generation and adaptive refinement, the nodal location as well as their number change from mesh to mesh. Numerical examples (see Chap 4) show that with remeshing the quality of the solution may sometimes be improved significantly without increasing the total number of unknowns. In some instances, solution quality increases even with a reduction of the number of unknowns since the node points are ideally distributed.

The crude initial mesh in this method is referred to as the background mesh. Initially a mesh with few elements can be used as a background mesh. Since a solution may not be attainable on this mesh, mesh generation parameters may be specified at the nodal points of the mesh. Intuitive solution knowledge may be useful in this specification, but a regular and uniform distribution of the parameters is sufficient in many cases. Once this information is available, a new mesh is created within the domain that is to be discretized. The domain boundaries are predefined by the user. The mesh generation parameters needed for the new mesh at various locations within the domain are linearly interpolated from the background mesh. A search algorithm based on the nodal coordinates of an element is used to determine the element on the background mesh to interpolate these parameters for a new point and is described in Appendix A. The initial background mesh need not coincide with the domain of discretization. Then a finite element solution is obtained on the new mesh, and the mesh generation parameters are computed based on the numerical solution. Continuing, the current mesh becomes a background mesh for the next mesh to be generated. The process of generating sequential meshes continues until the desired convergence for the solution is achieved.

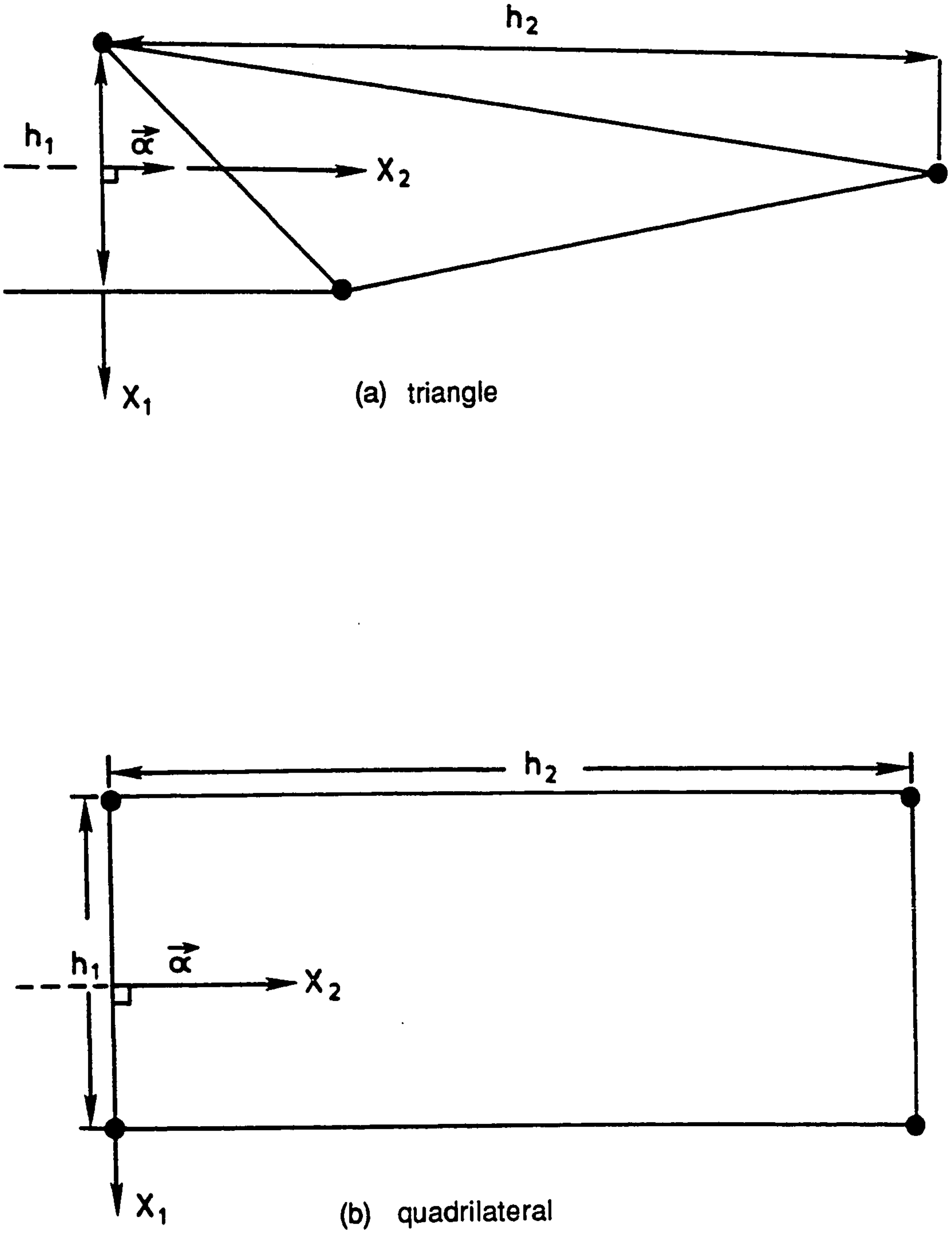
3.1.2 Mesh generation parameters

The mesh generation parameters used for the construction of a new mesh are shown in Fig. 7. They are: (1) two components of a vector α along which an element is to be stretched, (2) a spacing h_1 normal to this vector, and (3) a spacing h_2 tangential to this vector. Thus a new element has a dimension h_2 in the direction of the vector and a dimension h_1 in a direction normal to the vector .

To compute these mesh generation parameters adaptively, a dependent variable from an earlier solution is considered as an indicator and the magnitude and the direction of the error related to this indicator are used. The error estimator can be constructed in different ways depending on the error norm considered. One way to [40,41] determine the error estimation is by computing the interpolation error. This method has the advantage that it does not require the solution of local auxiliary problems for error indicators for each element, but among other things it has the disadvantages of requiring the computation of higher-order derivatives over each element and being global. A numerical scheme is devised to compute higher-order derivatives and is described in Appendix B. Though this scheme lacks mathematical rigor, it has proven adequate for practical purposes .

In one-dimensional problems the interpolation error E [42] for an element is defined as

$$E(x) = T_E(x) - T_{FE}(x) \qquad (3.1)$$



(a) triangle

(b) quadrilateral

Fig. 7. Mesh generation parameters.

where T_E is the exact solution, T_{FE} is the finite element solution and h is the length of the element. A one-dimensional element whose end nodes are "i" and "i+1" is shown in Fig. 8. Taylor series expansion of the error about an interior point M can be written as follows,

$$E(x) = E(x_M) + \frac{dE(x_M)}{dx}(x - x_M) + \frac{1}{2}\frac{d^2E(x_M)}{dx^2}(x - x_M)^2 + \text{........} + \text{......} \qquad (3.2)$$

Now, select x_M at the point where the error is maximum so that

$$\frac{dE(x_M)}{dx} = 0 \qquad (3.3)$$

Hence equation (3.2) can be written as follows, after neglecting the third and higher order terms in Taylor expansion.

$$E(x) = E(x_M) + \frac{1}{2}\frac{d^2E(x_M)}{dx^2}(x - x_M)^2 \qquad (3.4)$$

Assuming the finite element solution matches with the exact solution exactly at the end nodes, that is $E(x_i) = E(x_{i+1}) = 0$. At x_i we have

$$0 = E(x_M) + \frac{1}{2}\frac{d^2E(x_M)}{dx^2}(x_i - x_M)^2 \qquad (3.5)$$

or

$$E(x_M) = -\frac{1}{2}\frac{d^2E(x_M)}{dx^2}(x_i - x_M)^2 \qquad (3.6)$$

or

$$|E(x_M)| \leq \frac{1}{2}\left|\frac{d^2E(x_M)}{dx^2}\right|(x_i - x_M)^2 \qquad (3.7)$$

We can choose node "i" or "i+1" in computing x_i such that

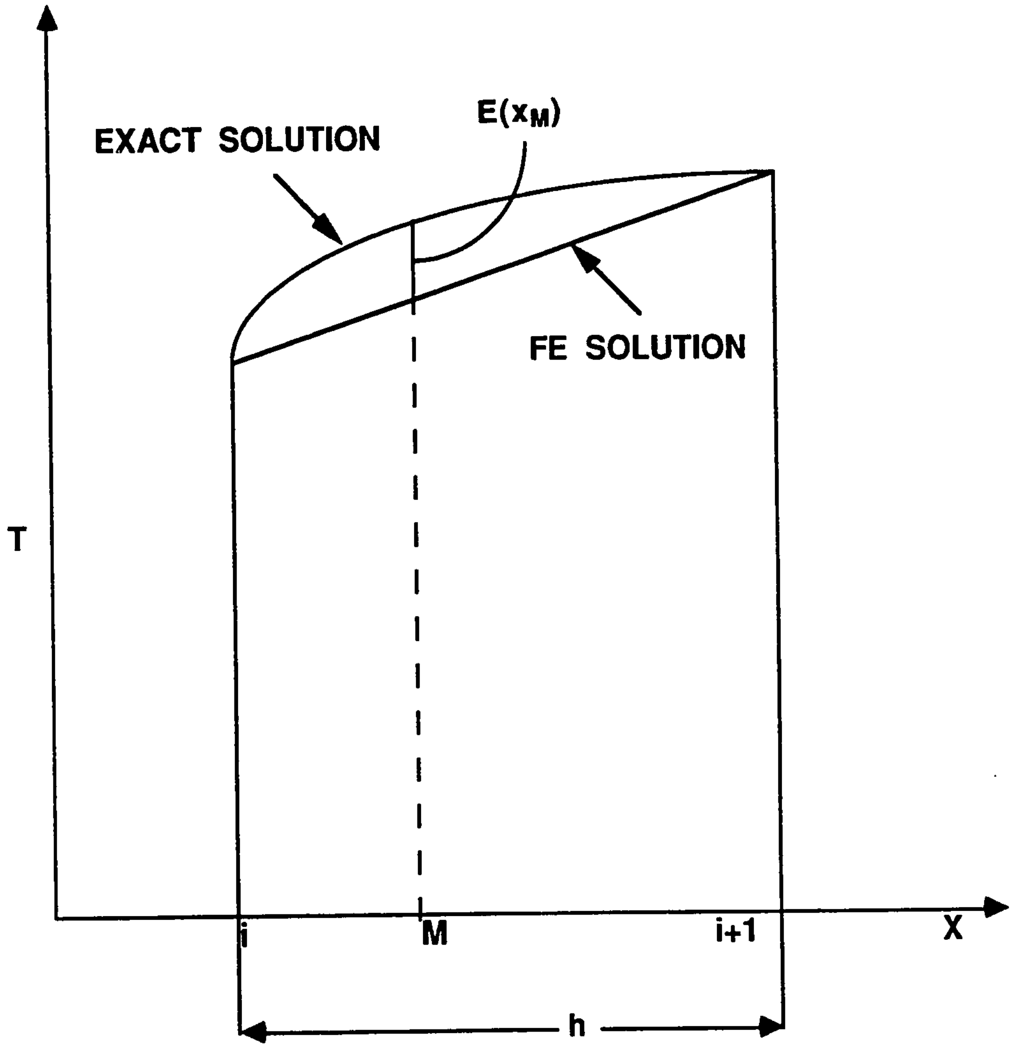


Fig. 8 Interpolation error for a one-dimensional element

$$|x_i - x_M| \leq \frac{h}{2} \qquad (3.8)$$

Implementing equation (3.8) in equation (3.7) we have the following expression which bounds the maximum interpolation error in the element.

$$\begin{aligned} |E(x_M)| &\leq \frac{1}{8}h^2\left|\frac{d^2E(x_M)}{dx^2}\right| \\ &\leq \frac{1}{8}h^2\left|\frac{d^2T_E(x_M)}{dx^2} - \frac{d^2T_{FE}(x_M)}{dx^2}\right| \\ &\leq \frac{1}{8}h^2\left|\frac{d^2T_E(x_M)}{dx^2}\right| \qquad (3.9)\end{aligned}$$

In the above expression, the second derivative of the finite element solution is made zero because of interpolation function is assumed linear. In elliptic problems it is proved [43] that equi-distribution of the interpolation error would yield optimal meshes. This approach is extended to hyperbolic problems successfully by several researchers [22,23,32,36,44]. This principle motivates the following equation which provides a requirement for obtaining near optimal meshes in 1-d applied to hyperbolic problems.

$$h_e^2\left|\frac{d^2\,T_E}{d\,x^2}\right|_e = \text{CONSTANT} \qquad (3.10)$$

where h_e is the length of the element "e". In adaptive refinement schemes, the adaptive strategy is based on equal distribution of solution error for elements. In remeshing, the adaptive strategy is based on an interpolation of the solution error at nodes.

In an extension of the one-dimensional interpolation error concept to two-dimensions, E(x,y) can be written in a Taylor expansion about an interior point $(x_{M,}y_M)$ as follows, after neglecting the higher order terms.

$$E(x,y) = E(x_M, y_M) + (x - x_M)\frac{\partial E(x_M, y_M)}{\partial x} + (y - y_M)\frac{\partial E(x_M, y_M)}{\partial y}$$

$$+ \frac{1}{2}\left[(x - x_M)^2 \frac{\partial^2 E(x_M, y_M)}{\partial x^2} + (y - y_M)^2 \frac{\partial^2 E(x_M, y_M)}{\partial y^2}\right]$$

$$+ (x - x_M)(y - y_M)\frac{\partial^2 E(x_M, y_M)}{\partial x\, \partial y} \tag{3.11}$$

By selecting the point $(x_{M,}y_M)$ such that the interpolation error is maximum, equation (3.11) can be written as

$$E(x,y) = E(x_M, y_M) + \frac{1}{2}\left[(x - x_M)^2 \frac{\partial^2 E(x_M, y_M)}{\partial x^2} + (y - y_M)^2 \frac{\partial^2 E(x_M, y_M)}{\partial y^2}\right]$$

$$+ (x - x_M)(y - y_M)\frac{\partial^2 E(x_M, y_M)}{\partial x\, \partial y} \tag{3.12}$$

At a typical node "i" of an element , $E(x_{i,}y_i) = 0$. Hence at this node, equation (3.12) can be written as follows,

$$0 = E(x_M,y_M) + \frac{1}{2}\left[(x_i - x_M)^2 \frac{\partial^2 E(x_M,y_M)}{\partial x^2} + (y_i - y_M)^2 \frac{\partial^2 E(x_M,y_M)}{\partial y^2}\right.$$

$$\left. + (x_i - x_M)(y_i - y_M) \frac{\partial^2 E(x_M,y_M)}{\partial x \, \partial y}\right] \qquad (3.13)$$

or in other words,

$$E(x_M,y_M) = -\frac{1}{2}\left[(x_i - x_M)^2 \frac{\partial^2 E(x_M,y_M)}{\partial x^2} + (y_i - y_M)^2 \frac{\partial^2 E(x_M,y_M)}{\partial y^2}\right.$$

$$\left. + (x_i - x_M)(y_i - y_M) \frac{\partial^2 E(x_M,y_M)}{\partial x \, \partial y}\right] \qquad (3.14)$$

For a square element of side "h", by making suitable assumptions the maximum interpolation error is bounded by the following expression.

$$|E(x_M,y_M)| \leq \frac{h^2}{8}\left[\frac{\partial^2 T_E}{\partial x^2} + \frac{\partial^2 T_E}{\partial y^2} + 2\frac{\partial^2 T_E}{\partial x \, \partial y}\right] \qquad (3.15)$$

When the expression given by equation (3.15) is applied with the principle of equi-distribution of error, it resulted in the following expression in two-dimensions.

$$h^2\left[\frac{\partial^2 T_E}{\partial x^2} + \frac{\partial^2 T_E}{\partial y^2} + 2\frac{\partial^2 T_E}{\partial x \, \partial y}\right] = \text{CONSTANT} \qquad (3.16)$$

But the above condition would give information about only one dimension of an element "h" i.e. in other words the condition would yield meshes with very regular elements like squares and equilateral triangles. It is highly desirable in hyperbolic problems where discontinuities like shocks exist, to stretch the elements along these one-dimensional features. So Peraire and Morgan [36] have modified this approach to accommodate a stretching factor in element creation. In this approach the local principal directions X_1 and X_2 are determined from the nodal second derivative tensor of the dependent variable ϕ and the corresponding eigenvalues λ_1 and λ_2 are computed along these principal directions as follows

$$\lambda_1 = \left|\frac{\partial^2 \phi}{\partial X_1^2}\right| \quad ; \quad \lambda_2 = \left|\frac{\partial^2 \phi}{\partial X_2^2}\right| \qquad (3.17)$$

Now the equi-distribution of error principle in 1-d is applied to each of these principal directions separately resulting in

$$h_1^2 \lambda_1 = h_2^2 \lambda_2 = \text{CONSTANT} \qquad (3.18)$$

where h_1 and h_2 are the spacings required in X_1 and X_2 respectively. The constant in the right hand side of the equation (3.18) is computed as h^2_{min} $|\lambda_{max}|$ where h_{min} is the minimum spacing specified by the user and λ_{max} is the maximum eigenvalue in the whole domain. Now the stretching parameter S is computed as

$$S = \left(\frac{\lambda_1}{\lambda_2}\right)^{\frac{1}{2}} \qquad (3.19)$$

Maximum allowable values h_{max} and S_{max} are specified by the user to put an upper bound to the aspect ratio of the element. In addition, in practice the shape of the element is constrained by limits on the internal angles so that a bad shaped element is not created. Usually these limits are between 45° to 135° for a quad element and 30° to 150° for a triangle [39].

3.1.3 A remeshing example

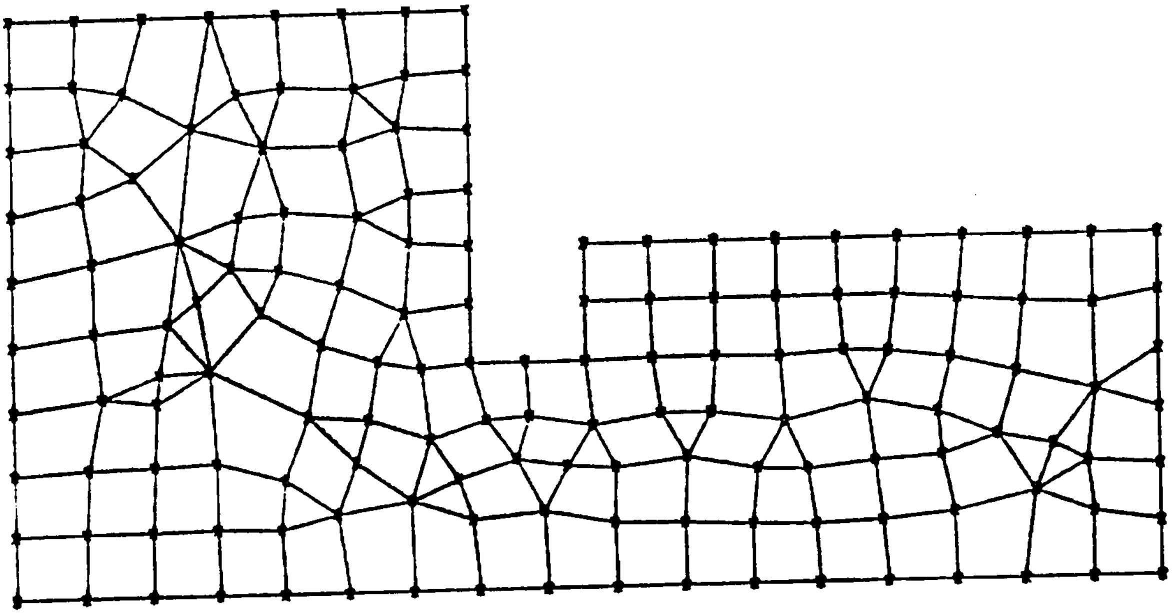
To illustrate the basic steps in remeshing, an example is presented in Fig. 9. The figure illustrates a new mesh being created from a solution obtained on a previous mesh. In Fig. 9a, the initial mesh is shown. Using this mesh, a finite element solution has been obtained. Figures 9b-9c show the evolution of the new mesh. Figure 9b shows the boundary points that are created first; Figs. 9c and 9d show the mesh at two stages of development, and Fig. 9e shows the final remesh. Notice the remeshing proceeds inward from the boundary, and that quads as well as triangles are generated as the remeshing proceeds.

3.2 REMESHING WITH QUADS AND TRIANGLES

3.2.1 Boundary discretization

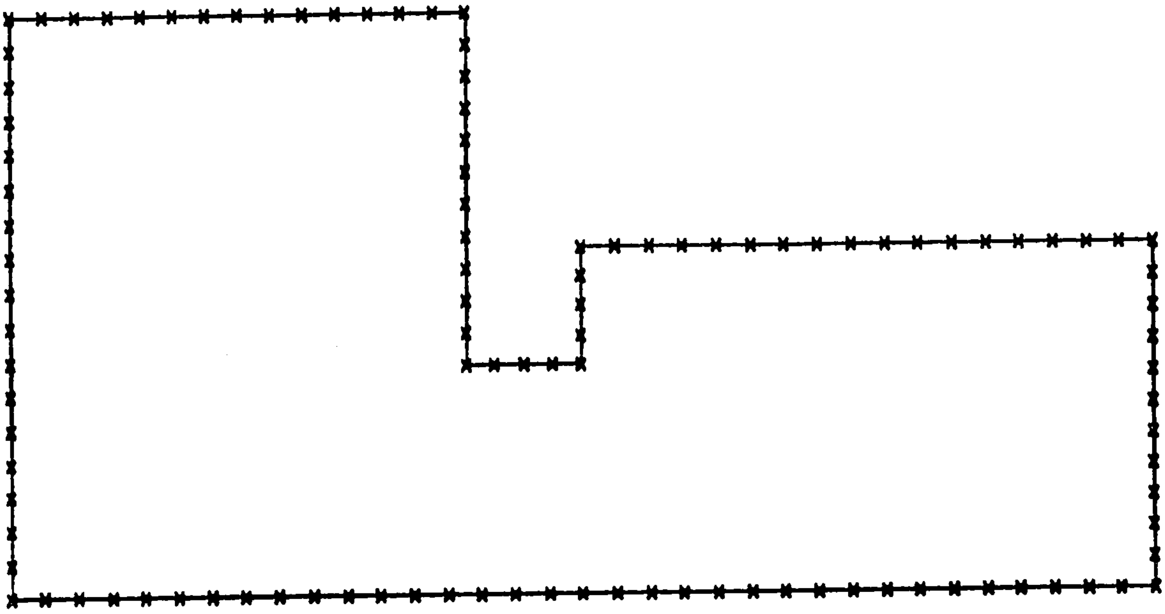
The mesh generation process starts with boundary discretization. Boundary segments joining fixed boundary nodes are ordered in an counter-clockwise manner for an external boundary and are ordered in a clockwise manner for an internal boundary defining a hole within the domain. This way the domain to be discretized always exists on the left hand side of the boundary .

Additional boundary nodes are included to satisfy the spacing requirements compatible with the background mesh. Each boundary



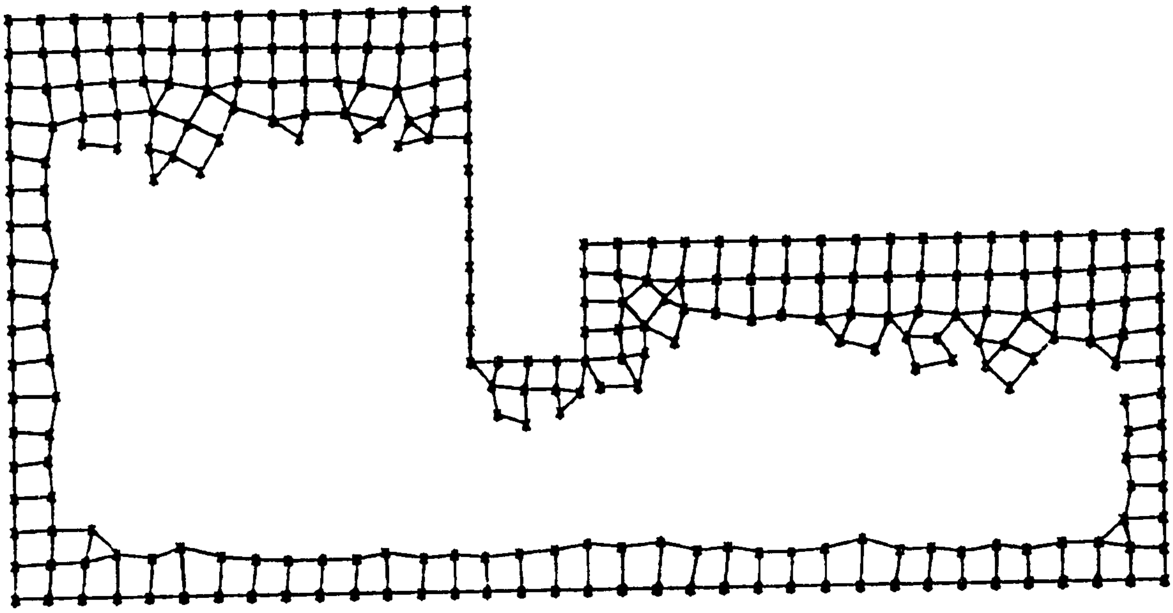
(a) Initial mesh

Fig. 9 An adaptive remeshing example



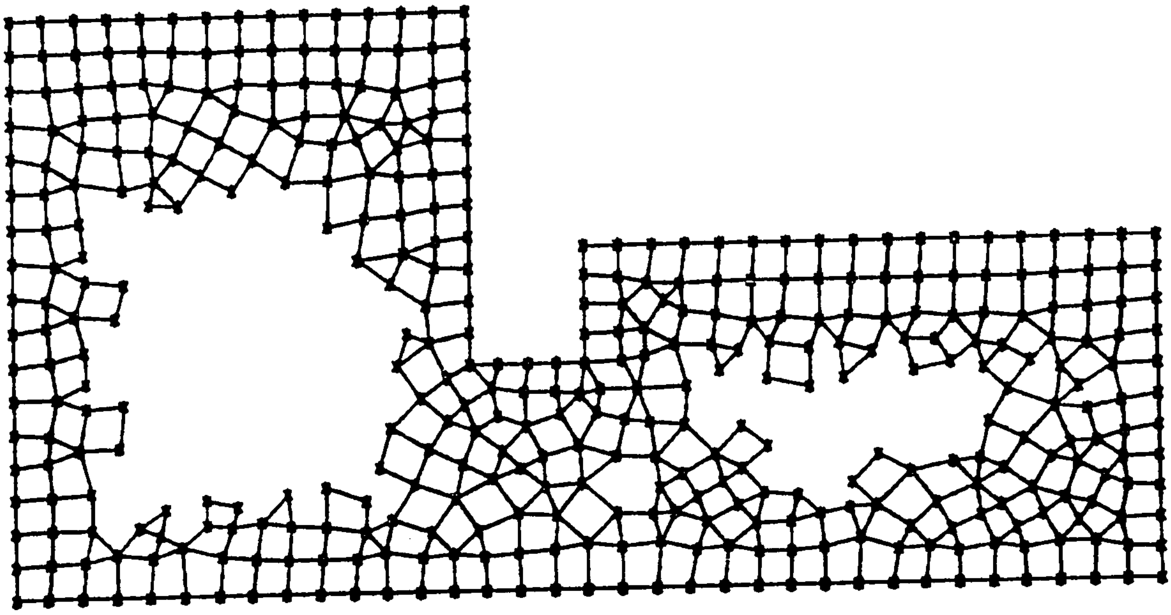
(b) Boundary discretization

Fig. 9 An adaptive remeshing example (continued)



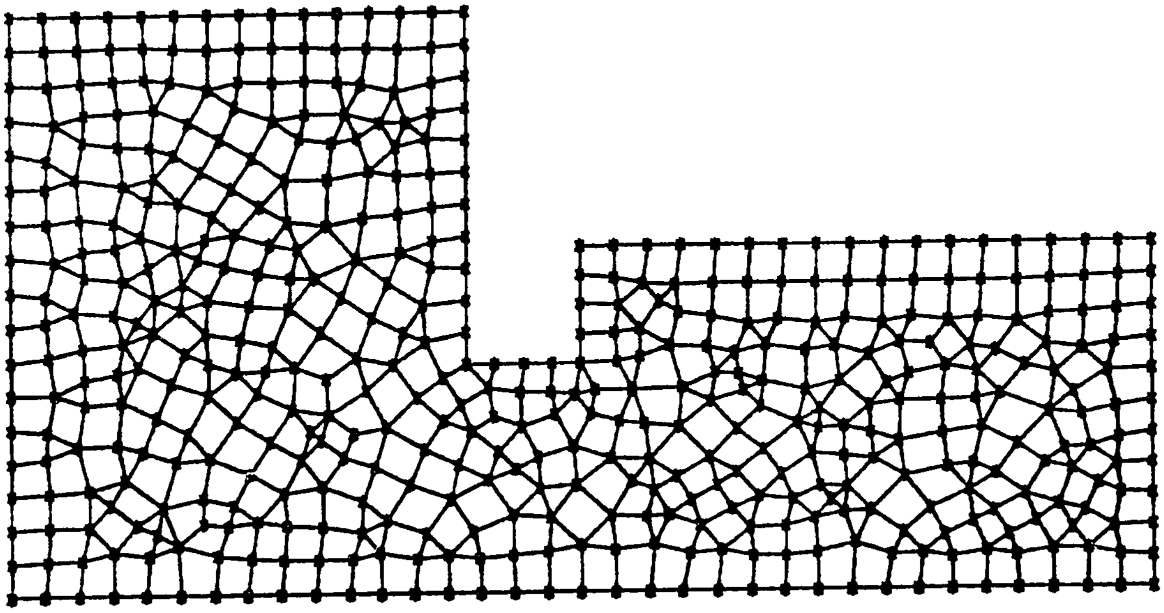
(c) A stage of remeshing

Fig. 9 An adaptive remeshing example (continued)



(d) Later stage of remeshing

Fig. 9 An adaptive remeshing example (continued)



(e) Final mesh

Fig. 9 An adaptive remeshing example (concluded)

segment is discretized in order until the entire boundary is covered. For each boundary segment, the total length L is computed. The mesh generation parameters are interpolated from the background mesh at some sampling points along the segment(typically this number is chosen to be 100 in this work). From known mesh parameters, the value of the spacing in the direction of the segment is computed at all the sampling points.

After the spacing h_1 along the boundary line has been evaluated at the sampling points, the number of sides N which need to be created along the boundary line is calculated as follows.

$$N = \sum_{i=1}^{M-1} \frac{1}{2}\left[\frac{1}{(h_1)_i} + \frac{1}{(h_1)_{i+1}} \right] L_i \qquad (3.20)$$

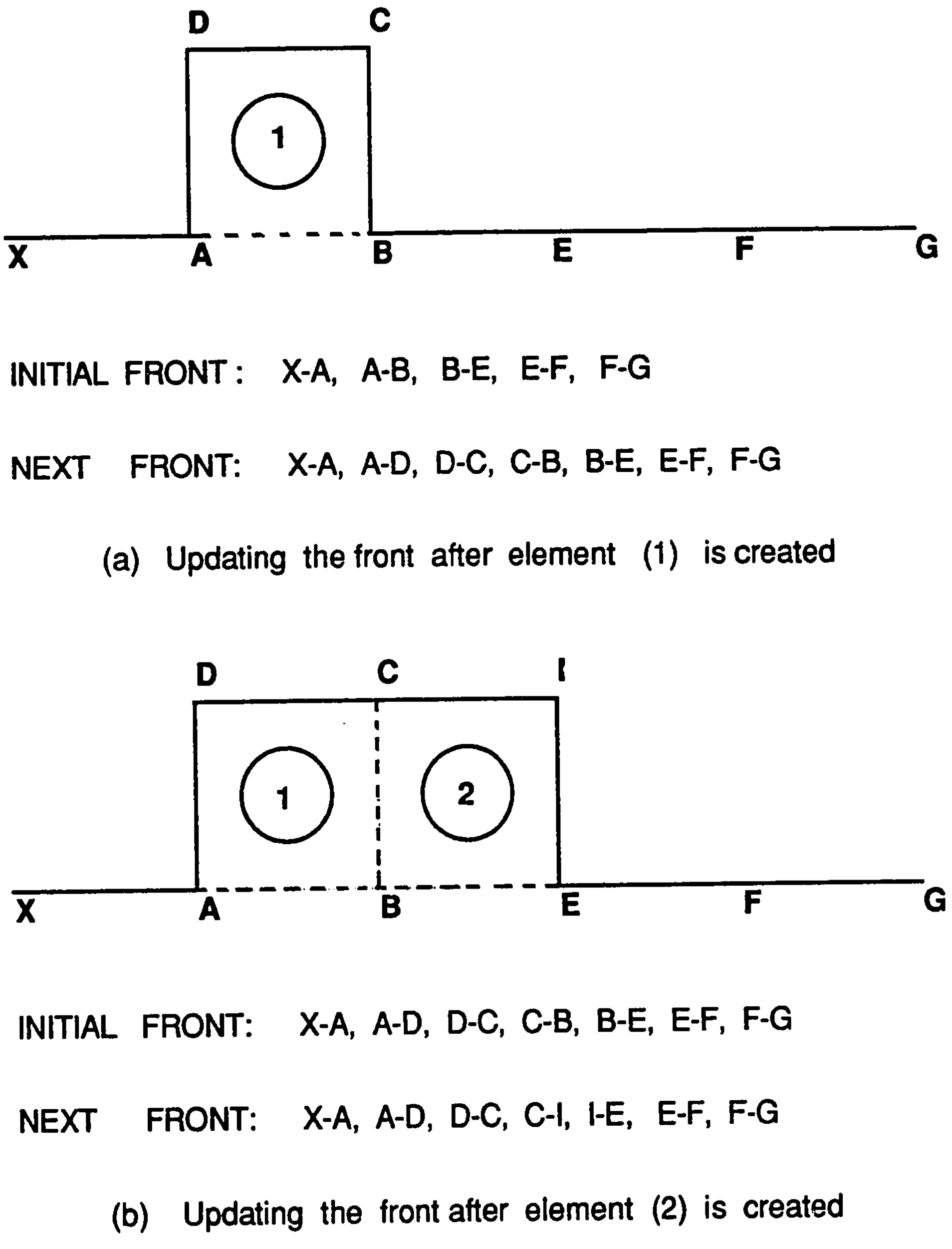
where M is the number of sample points, L_i is the distance between adjacent sample points. Note that N is the nearest integer of the right hand side value in equation (3.20). Then the intermediate spacings between these points are computed based on the local h_1 values. The spacings are scaled appropriately to account for adjusting the value of N. Once the intermediate spacings are established, the coordinates of the new points are computed.

3.2.2 The advancing front technique

The mesh generation process is based on an advancing front technique similar to the method proposed by Lo [14]. The front consists of adjacent nodes joined by line segments. The initial front consists of the boundary segments that link adjacent boundary nodes. As the mesh construction goes on, the front changes its shape. When an element is created, new sides are included in the front which can be used for further creation, and the sides that cannot be used further are deleted from the front. This updating process is shown in Fig 10a. When a new element 1 is created new nodes C and D are created and hence the frontal segments AD, DC and CB become active in the front. At the same time the segment AB that cannot be used any further is deleted from the front. During this updating process a check is made whether any other existing frontal segment counteracts a new segment in which case both the segments are deleted from the front. This is illustrated in Fig. 10b, where a new element 2 is created adjacent to element 1 in Fig. 10a with B, E, I and C as its nodes, the new side BC counteracts with the already existing segment CB and hence both are eliminated from the front. Thus the front can be defined as a chain of line segments that surround the domain that remains to be discretized. The front changes its shape constantly during the construction process and vanishes when the mesh is complete. Fig. 9 shows different stages of front evolution as the mesh generation progresses.

3.3 ELEMENT GENERATION

Element generation in this method takes place along with node creation. Element generation always proceeds from the smallest frontal segment giving priority to regions which require refinement. Once this frontal segment is identified, nodes are created corresponding to this segment so that a new element is generated having this frontal segment as one of its sides. There are two options that are attempted in a sequence until an element is created. The first option creates quadrilateral elements and the second option creates a triangular element. Once the mesh is complete, a post-process



INITIAL FRONT : X-A, A-B, B-E, E-F, F-G

NEXT FRONT: X-A, A-D, D-C, C-B, B-E, E-F, F-G

(a) Updating the front after element (1) is created

INITIAL FRONT: X-A, A-D, D-C, C-B, B-E, E-F, F-G

NEXT FRONT: X-A, A-D, D-C, C-I, I-E, E-F, F-G

(b) Updating the front after element (2) is created

Fig. 10 Updating a front during mesh generation

combines two triangles at a time, wherever possible, to create as many well-shaped quadrilaterals as possible. These options are discussed below.

3.3.1 Option 1

This option attempts to make a quadrilateral on the smallest frontal segment by locating two points within the domain. Shape limitations constrain the internal angles of the element and aspect ratio limitations bring the proper stretching to the element. Maximum values of the stretching factor used in this work are 3 to 4 since higher values resulted in unacceptable elements or incomplete meshes. This is a limitation of the method. The different steps involved in this option to create an element are discussed below .

(1) The smallest frontal segment is chosen as a base and the end nodes of this segment are denoted as A and B. The local mesh generation parameters are computed at the midpoint M of this segment based on the values of the mesh generation parameters at the end nodes. A local coordinate axes transformation is made in such a way that the local x-axis is aligned with α_M , vector of stretching at the midpoint and is scaled by a factor S_M (stretching parameter) at the midpoint of the segment.

(2) In this transformed coordinate system two points D and C are located on the normals to the segment AB at A and B respectively . The distance δ between the new points and the old points is computed as follows :

$$\delta = 0.5 * AB \quad (\text{if } \delta_M < 0.5AB) \ ;$$

$$\delta = 2.0 * AB \quad (\text{if } \delta_M > 2.0AB) \ ;$$

$$\delta = \delta_M \quad (\text{if } 0.5\, AB < \delta_M < 2.0AB) \qquad (3.21)$$

This is to assure that the new element to be created is close to a square in the transformed system. This procedure is illustrated in Fig. 11a.

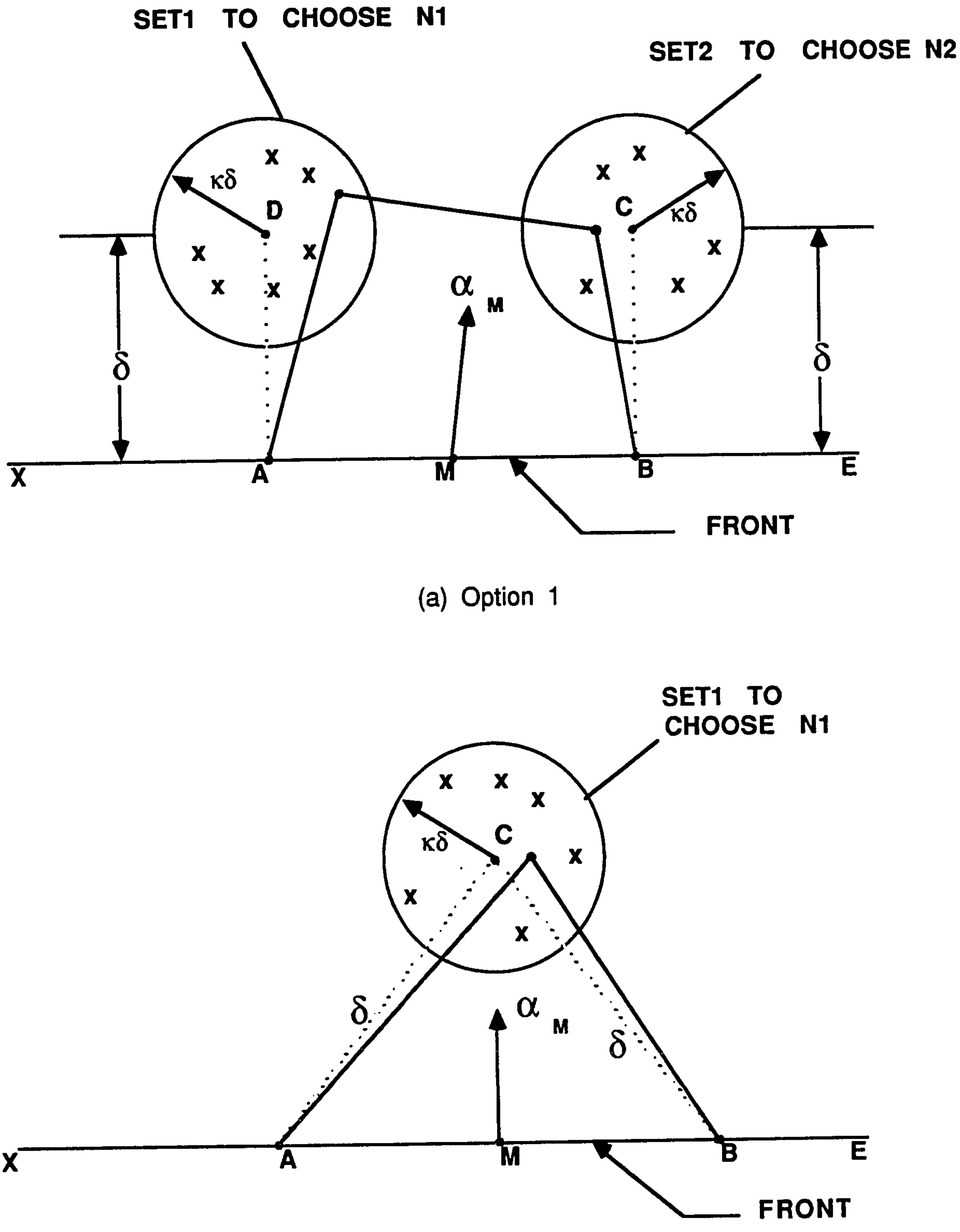
(3) Two sets of nodes that already exist (illustrated as "x" in Fig. 11) and are active are identified corresponding to the newly located points C and D respectively. These sets of nodes are identified in such a way that they are within a circle of radius kδ from each of the points C and D where k is a constant (in the current work k is chosen to be 0.5). In each set the nodes are ordered according to the distance between their location and the corresponding newly located point (either C or D) in an increasing manner. For convenience the set corresponding to D is called set1 and the one corresponding to C is called set2 in future discussion. C and D are added to sets 2 and 1 respectively at the end of each set.

(4) A selection of points, N1 from set1 and N2 from set2 is made in such a way that the first point in set1 is chosen as N1 and the points in set2 are chosen in the order, for N2 so that an acceptable element is created. The following checks are made to ensure that the quadrilateral joining A, B, N2 and N1 is acceptable;

(a) The diagonals of the quad, A-N2 and B-N1 intersect with each other (This is required to avoid elements with internal angles greater than 180°),

(b) no other frontal segment intersects either of the diagonals A-N2 or B-N1 and

(c) no active node on the front exists either in the triangle AN2N1 or BN1N2. If an acceptable element is not created, the next point in set1 is chosen as N1 and the whole process is repeated. The selection of these points among a set of points is required because before new nodes are created, already existing nodes in the immediate vicinity are considered



(a) Option 1

(b) Option 2

Fig. 11 Options considered in element creation

instead, so that the number of nodes is optimized. This process also avoids odd-shaped elements that have to be accommodated between the closely spaced new and old nodes.

(5) Once an element is created, it is included in the list of elements and its connectivity information is stored. The front is updated as explained in the earlier section on the front concept. If N1 and N2 happen to be C and D(new nodes) the mesh generation parameters are interpolated from the background mesh at these nodes and stored along with their coordinates .

A new element is attempted on the updated front starting from the smallest segment following the same steps explained above. In case a quad element is not created anywhere in the front, then the next option of creating a triangle is attempted.

3.3.2 Option 2

In this option, a triangle is created instead of a quadrilateral. The procedure to create a triangle is similar to option 1, except one node instead of two is created for element construction at the frontal segment. After the local coordinate transformation is performed on the frontal segment AB as in option 1, a third point C is located which is equidistant from A and B as shown in Fig. 11b. The distance between C and A or B is same as δ in option 1 so that an element close to an equilateral triangle is created in the transformed coordinate system. A set of active nodes on the front are identified which are within a circle of radius $k\delta$ where k is a constant (in the current work k is chosen as 0.8) with the center at C and are ordered according to the distance between their location and the point C in an increasing manner. Point C is added in the end of this set. A point N1 is chosen from this set, starting from the first node in the set, so that an acceptable triangle ABN1 can be formed. The following checks are made for this purpose:

(a) No other frontal segment intersects the line joining N1 and M (midpoint of AB)

(b) No other active node exists within the triangle ABN1 .

If an element cannot be formed with N1, the next point in the set is chosen as N1 and the checks are repeated. If an element cannot be created at this frontal segment, another segment is chosen as the base AB and the entire process is repeated. Once an element is created the coordinates are transformed back and the element is included in the list of elements. The front is updated and the minimum frontal segment on the new front is determined. Then option 1 is attempted again to create a quad element.

3.4 STRUCTURED REMESHING

The unstructured remeshing method discussed above is directly applicable to several problems including inviscid flows. Any flow variable that undergoes significant changes either in a shock, expansion fan or a shear layer can be used as an indicator. The unstructured meshes generated in this manner are not the optimal meshes for boundary layers. A structured mesh stretched in the streamwise direction of the flow, with the spacings gradually increased in the transverse direction, is a highly desirable mesh for boundary layers. A stretched mesh is desirable because gradients in the flow variables dominate in the transverse direction compared to the streamwise direction. In high-speed flow applications, the elements near the wall where these gradients are maximum sometimes require aspect ratios of the order of 1000. It was found that it is difficult to generate such a mesh in the boundary layer using unstructured remeshing. Even if such a mesh is created it consists of high-aspect ratio triangles which show poor performance in the analysis.

Thareja et al. [44] used a structured mesh generating algorithm to create well stretched meshes in the viscous region and generated unstructured meshes in the inviscid region by remeshing. The edge of the boundary layer was explicitly defined in this method. A method of adaptively determining the boundary layer region is desirable. Determination of the edge of the boundary layer is difficult in problems where there is a strong shock-boundary layer interaction. Moreover, if the structured mesh is not created in an adaptive manner, there is a possibility of over-prescribing the number of mesh points in the boundary layer. Data management is also cumbersome in using two mesh generation schemes to generate a single mesh for the whole domain.

A new adaptive method of generating a structured mesh in the boundary layer region has been developed. In this method, as the initial front is set up along the boundary of the domain, mesh generation begins on the part of the front covering the "no-slip" surface of the boundary. Imaginary lines are drawn at all nodal points on the "no-slip" surface in a direction normal to the surface. A layer of quadrilateral elements is placed on this surface by locating points on these imaginary lines. The location of these points is based on the local spacing values that are interpolated from the background mesh. Mesh construction proceeds as more layers of elements are created one above the other. Every time a new layer is started, a check is made whether the edge of this layer falls within the boundary layer. This check is adaptively based on a viscous indicator from a solution obtained on an earlier mesh. Thus as the edge of the boundary layer is reached, no more elements are created. The front is updated as in the case of unstructured remeshing during this process at the end of which the front defines the edge of the inviscid flow-field. Generation of elements in a simple boundary layer and the final front defining the inviscid region is shown in Fig. 12. After the structured mesh in the boundary layer is complete, the unstructured mesh for the inviscid region is then generated in the usual manner. So the viscous-inviscid interface is determined adaptively, and the same indicator can be used for remeshing in both regions.

In developing the adaptive structured remeshing method for boundary layers, the following assumptions are made: (1) the maximum gradient of the flow variable used as an indicator occurs at the no-slip surface, and (2) principal directions of the second derivative tensor of the indicator coincide with the tangential and normal directions of the no-slip surface within the boundary layer .

3.5 MESH SMOOTHING

The mesh that is generated may have some distorted elements because of maximum angle tolerances and/or aspect ratios permitted. To minimize element distortions, the nodes are repositioned. This operation is referred to by some researchers as mesh smoothing. In the smoothing process every interior node is moved to the centroid of the polygon formed by the nodes that are connected to it. Boundary nodes are constrained to move along the boundary edge on which they lie .

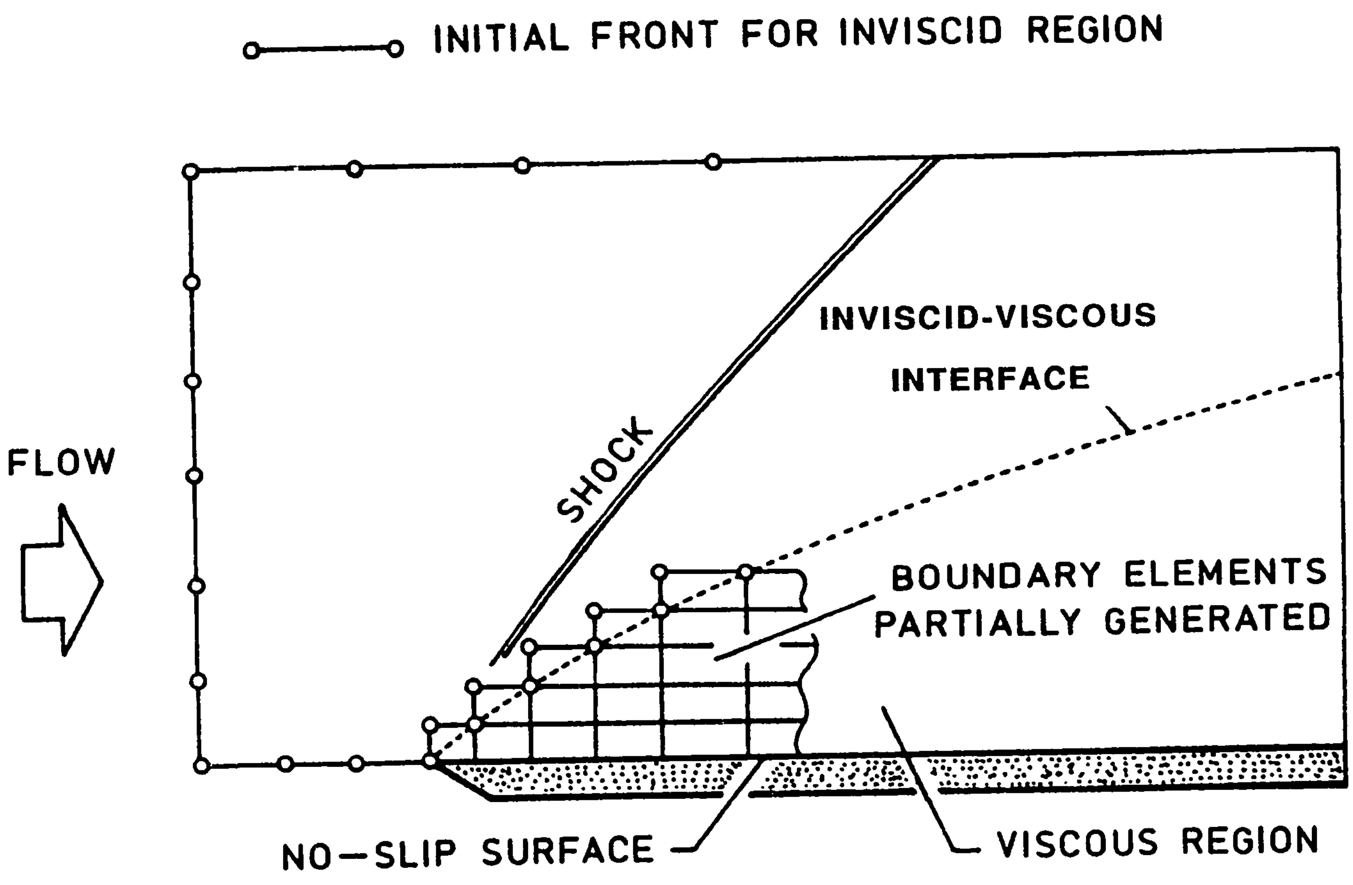


Fig. 12. Adaptive remeshing for a simple viscous flow.

Let C denote the node that is to be repositioned, and let M denote the number of elements that share C as a common node. Then the new position vector of node C R_c, is computed from

$$\left(\sum_{i=1}^{M} A_i \right) \overline{R_c} = \sum_{i=1}^{M} A_i \overline{R_i} \qquad (3.22)$$

where A_i denotes the area of the ith connected element, and R_i is the position vector of the centroid of the ith element. The new position of all nodes are computed in this fashion. Then the process is repeated for several iterations until the nodes reach their final positions. Typically ten iterations are sufficient to bring the nodes to their final positions. When this repositioning is done adaptively the smoothing process becomes the "r-refinement" method discussed earlier.

Chapter 4

ADAPTIVE REMESHING FOR FINITE ELEMENT THERMAL ANALYSIS

The new remeshing method described in Chapter 3 was tested for convergence and effectiveness before it was used for high speed viscous flow applications. Thermal analysis was chosen because the governing equation, i.e. the steady state heat conduction equation, is elliptic, and convergence rates with respect to error norms have been established [3]. Three steady state heat conduction example problems are solved to study the convergence rates of the L_2-norm of the temperature error. All three problems have exact solutions and error norms are computed based on the exact and finite element solutions. Geometry for this problems is selected to be simple since Chapter 3 shows that the new method can handle complicated geometries.

4.1 Applications

For heat conduction problems, let $T_E(x,y)$ be the exact solution, and $T_{FE}(x,y)$ be the finite element solution. The error $E(x,y)$ in the approximation at any point is the function

$$E(x,y) = T_E(x,y) - T_{FE}(x,y) \tag{4.1}$$

In the examples that follow, the global error in the approximate solution is measured in the L_2-norm defined by

$$\|E\| = \left(\frac{\int_A (T_E - T_{FE})^2 \, dxdy}{A} \right)^{\frac{1}{2}} \qquad (4.2)$$

where A is the area of the solution domain. The integral is evaluated on an element basis and summed over all elements in the domain. Four point Gauss integration is used to evaluate the integrals; a check with nine point integration for a solution with steep gradients verified the accuracy of the four point approach.

4.1.1 Example 1

The first example (Fig. 13) consists of the classical problem of heat conduction in a square plate where T(x,y) is a solution of the steady state equation,

$$k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + Q = 0 \qquad (4.3)$$

subject to the boundary conditions,

$$\begin{aligned} \frac{\partial T}{\partial x}(0,y) &= 0 \quad ; \quad T(a,y) = 0 \\ \frac{\partial T}{\partial y}(x,0) &= 0 \quad ; \quad T(x,a) = 0 \end{aligned} \qquad (4.4)$$

for constant thermal conductivity k and heat generation Q. The exact solution given by Carslaw and Jaeger [45] is

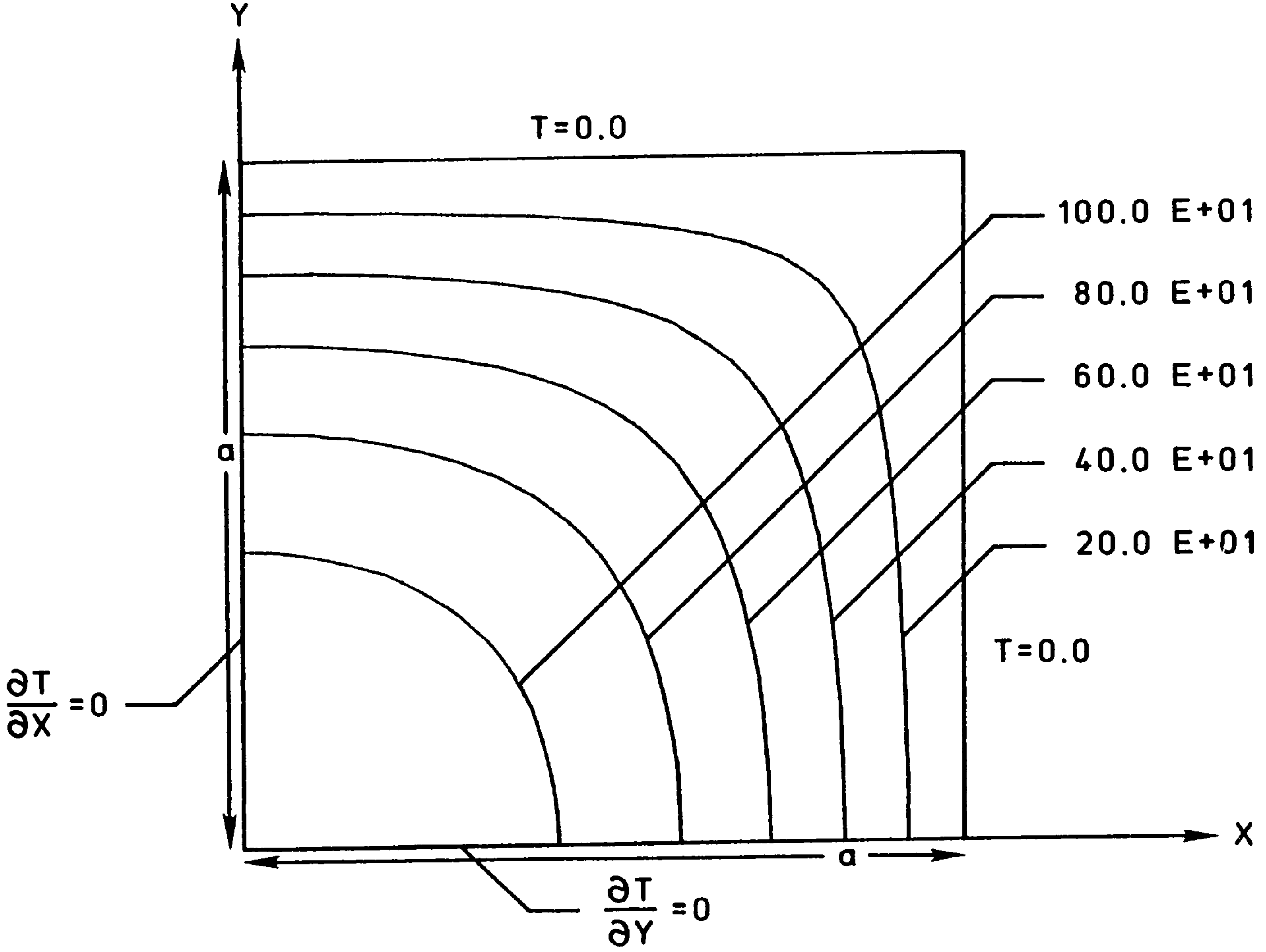

Y
T=0.0
100.0 E+01
80.0 E+01
60.0 E+01
40.0 E+01
20.0 E+01
T=0.0
a
$\frac{\partial T}{\partial X} = 0$
X
a
$\frac{\partial T}{\partial Y} = 0$

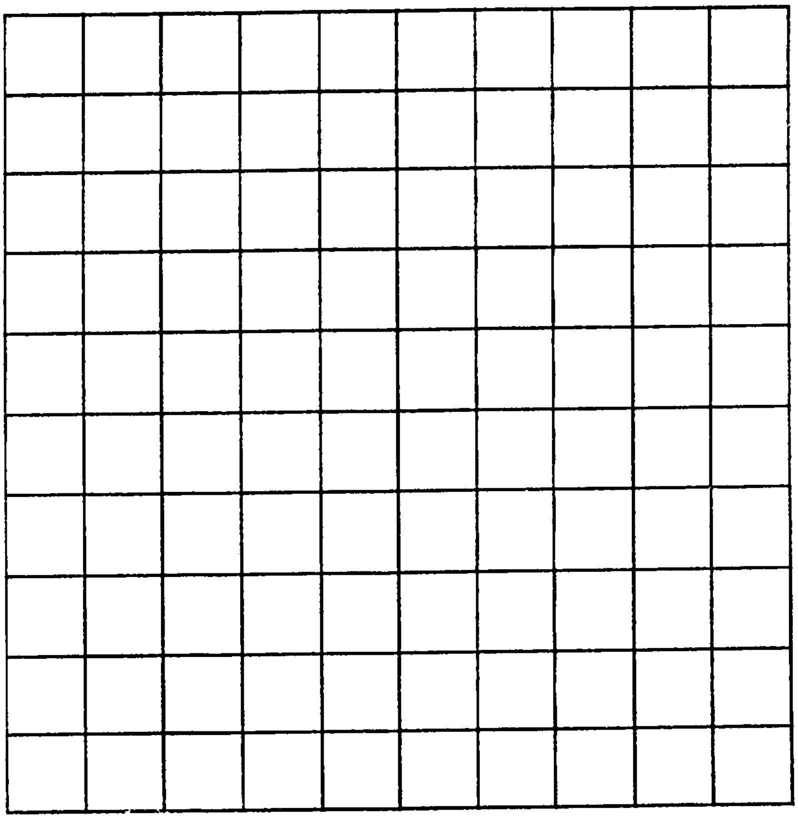
Fig. 13. Problem statement for 2D heat conduction problem with internal heat generation, Example 1.

$$T(x,y) = \frac{Q(a^2 - x^2)}{2k} - \frac{16Qa^2}{k\pi^3} \sum_{n=0}^{\infty} \frac{(-1)^n \cos(2n+1)\frac{\pi x}{2a} \cosh(2n+1)\frac{\pi y}{2a}}{(2n+1)^3 \cosh(2n+1)\frac{\pi}{2}} \tag{4.5}$$

Finite element solutions were obtained for: (1) three meshes with uniform h refinement of square elements, and (2) two adaptive remeshes where the remeshing started from the solution on uniform mesh 2. The meshes used in the remeshing approach are shown in Fig. 14, and solution errors are tabulated in Table 1 .

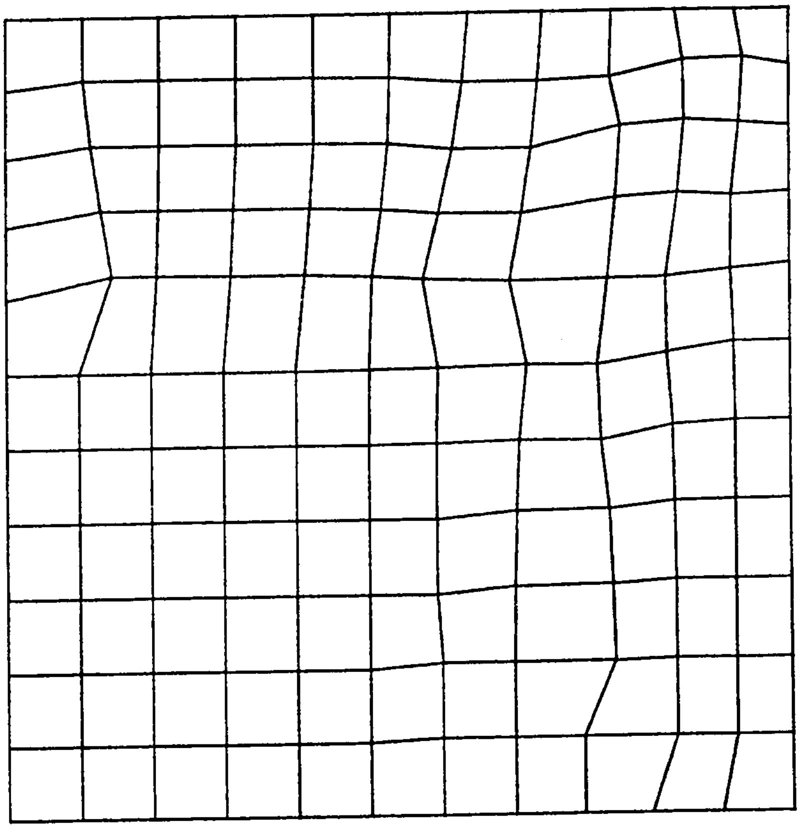
The remeshes (meshes 2 and 3) shown in Fig. 14, do not differ greatly from those obtained in a uniform h refinement. The exception is the two triangles that appear in mesh 3, Fig. 14c. The triangles are generated spuriously by the program, possibly due to the tolerance for element distortions. Some studies of mesh movement (not shown) indicated that a nearly optimum mesh for this problem consists of square elements slightly distorted into quads but with very smooth element interface curves. The remeshes shown in Fig. 14 appear to have some features of the optimum meshes, but fall short of producing smooth optimum meshes. Nevertheless, when solution errors are considered, the remeshes of Fig. 14 gave acceptable results.

Considering Table 1, note that uniform h refinement reduces the global solution error according to h^2 as predicted by finite element theory. Using the average element size h_a, the solutions based on the remeshes follow the same trend; that is, the solution is $O(h_a^2)$. Thus in this problem, remeshing is improving the solution quality at the same rate as a uniform h refinement. The



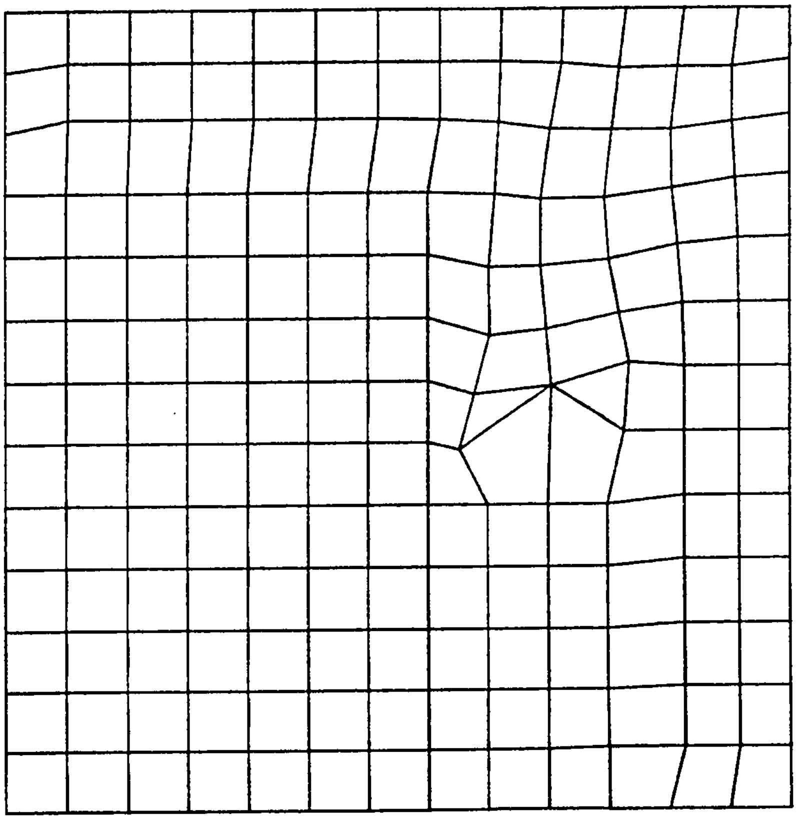
(a) Initial mesh

Fig. 14 Successive remeshes for Example1.



(b) Adaptive remesh 1

Fig. 14 Successive remeshes for Example1 (continued).



(c) Adaptive remesh 2

Fig. 14 Successive remeshes for Example1 (concluded).

Table 1 Comparative Solution Errors for Example 1

Uniform h Refinement

Mesh	Unknowns	h	Error, $\|\| e \|\|$
1	36	4.0	4.348
2	121	2.0	1.087
3	441	1.0	0.2717

Remeshing

Mesh	Unknowns	h*	Error, $\|\| e \|\|$
1	121	2.0	1.087
2	144	1.818	0.8816
3	196	1.539	0.6381

h* denotes an average element size defined as the square root of the area A divided by the number of elements in the remesh.

reason that remeshing shows no advantage over uniform h refinement is that the problem solution is smooth with no steep gradients and the remeshes look similar to uniform meshes.

4.1.2 Example 2

This model problem [41] assumes a solution that satisfies Eq. 4.3 and homogeneous Dirichlet boundary conditions. The solution takes the form

$$T(x,y) = \psi(x, x_0, c_x)\ \psi(y, y_0, c_y)$$

$$\psi(x, x_0, c_x) = (x + x_0)^{c_x} + Ax + B \tag{4.6}$$

where A and B are selected so that the homogeneous boundary conditions are satisfied. In the example, $x_0 = y_0 = 0.03$, and $c_x = c_y = -0.25$. The solution is formulated to give steep gradients near the coordinate axes. For the finite element computations, the solution, Eq. 4.6, is substituted into the governing equation 4.3 which is solved for Q(x,y). The heat generation Q is then integrated over each element, and nodal temperatures are computed. Temperature contours for the exact solution , Eq. 4.6 are presented in Fig. 15.

Finite element solutions were obtained for: (1) three meshes with uniform h refinement of square elements, and (2) three adaptive remeshes where the remeshes started from the solution on uniform mesh 2. The meshes used in the remeshing approach are shown in Fig. 16, and solution errors are tabulated in Table 2.

The remeshes shown in Fig. 16 consist of a mixture of quadrilateral and triangular elements generally of good proportions. The elements do not follow

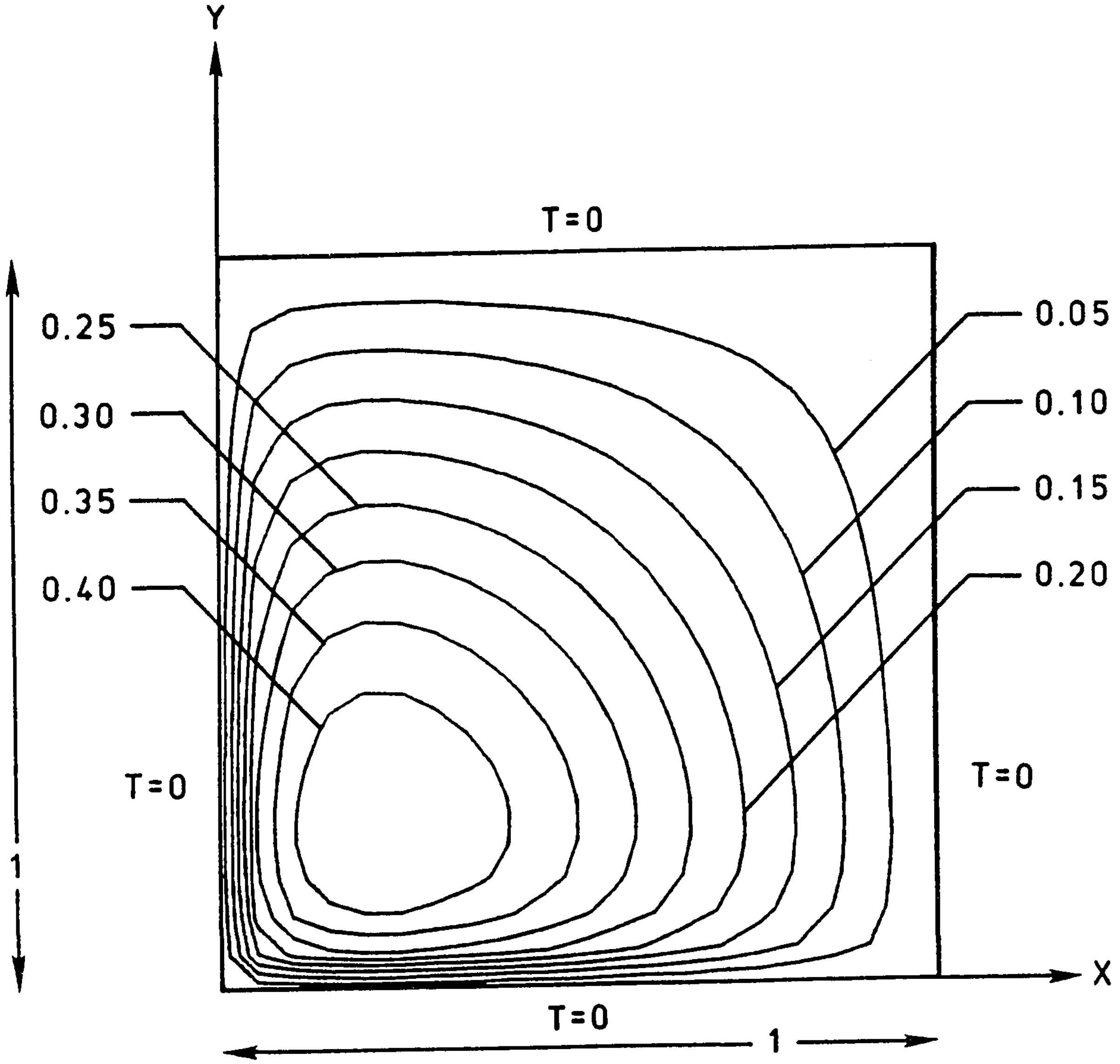
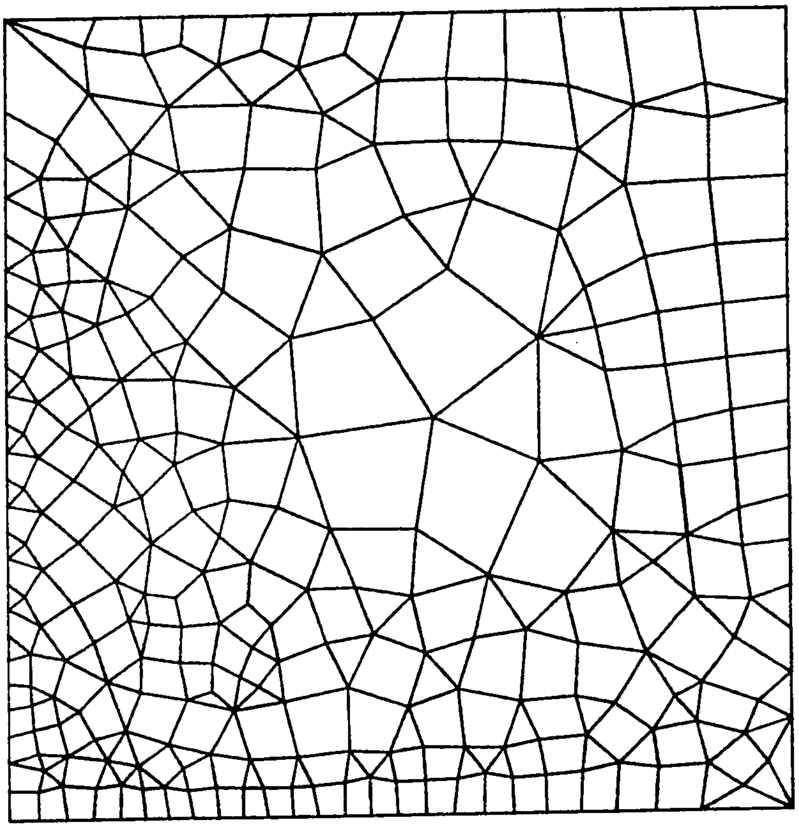
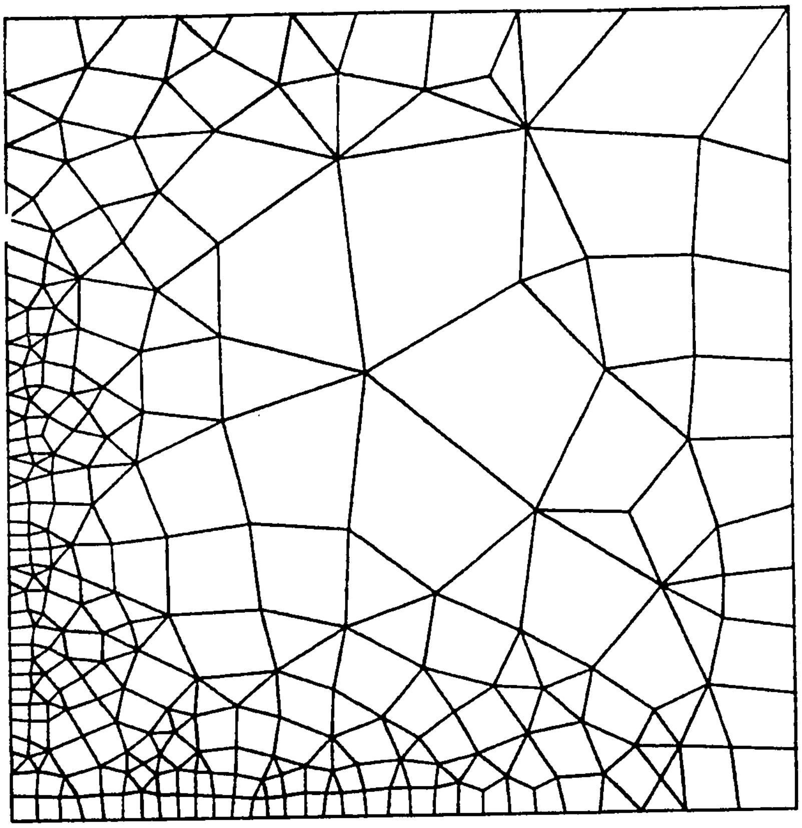


Fig. 15. Problem statement for 2D heat conduction, Example 2.



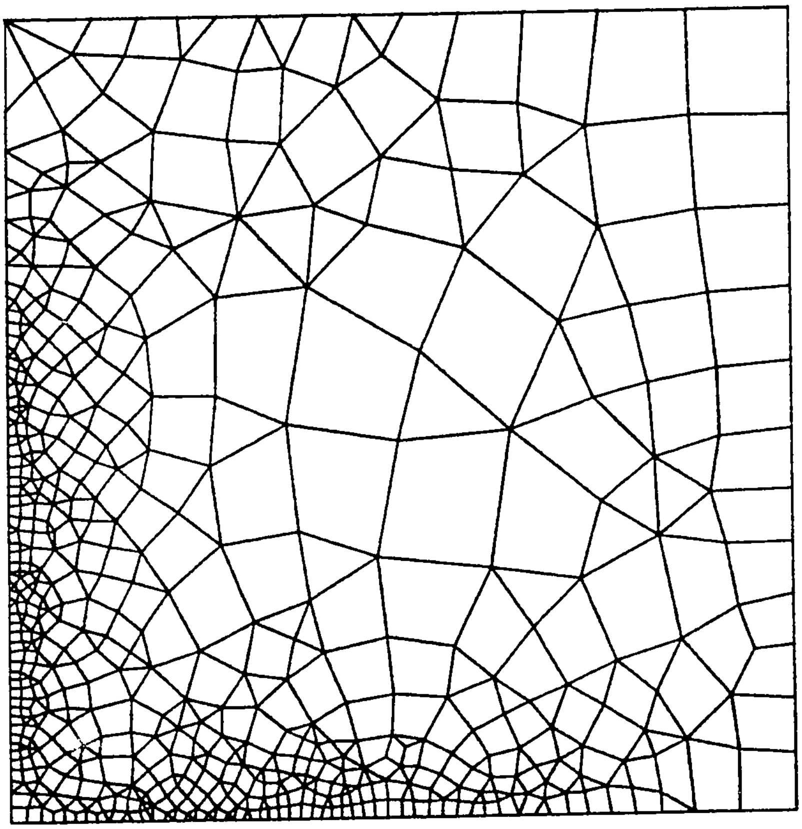
(a) Adaptive remesh 1

Fig. 16 Adaptive remeshes for Example 2.



(b) Adaptive remesh 2

Fig. 16 Adaptive remeshes for Example 2 (continued).



(c) Adaptive remesh 3

Fig. 16 Adaptive remeshes for Example 2 (concluded).

Table 2 Comparative Solution Errors for Example 2

Uniform h Refinement

Mesh	Unknowns	h	Error, || e ||
1	36	1.0	0.1272
2	121	0.5	0.07982
3	441	0.25	0.03305

Remeshing

Mesh	Unknowns	Error, || e ||
1	121	0.1272
2	299	0.01969
3	288	0.01270
4	572	0.00550

any clear pattern that might be intuitively suggested, but as the meshes are adaptively created there is clear refinement of elements at the sharp gradients near the coordinate axes. At the same time larger elements are being created away from the boundaries in regions of smaller gradients. Although the remeshes are not aesthetically satisfying, they are effective in producing high quality solution convergence as demonstrated in Table 2.

From Table 2, note that uniform h refinement reduces the global solution error at a rate less than the optimum rate, $O(h_a^2)$. This appears to be caused by the steep gradients present in the solution. Additionally, the optimum convergence rate is $O(h_a^2)$ only as an asymptotic limit which apparently has not been reached.

The errors based on the remeshes are reduced at a faster rate than for uniform refinement. The first remesh, which produced mesh 2 with 299 unknowns, gives an error $|| E || =$ 0.01969 which is smaller than the error produced on uniform mesh 3 which has 441 unknowns. In addition, the second remesh which produces mesh 3 has reduced the number of unknowns from 299 to 288 and simultaneously has reduced the error. Finally, if one more uniform h refinement were made assuming an optimal convergence rate of four, the solution error would reduce from 0.03305 to 0.00826 for over 1600 unknowns. Yet, the third remesh which produced mesh 4 has an error of 0.00550 for only 572 unknowns. Clearly for this problem with steep gradients the remeshing approach produces higher solution quality for reduced computational effort than uniform h refinement.

4.1.3 Example 3

In example 2, the gradients were roughly one-dimensional with the solution varying steeply normal to the x and y axes. For this final example, a model problem [41] is solved where the gradients are more two-dimensional. As before the model problem assumes a solution that satisfies Eq 4.3 and homogeneous boundary conditions. The solution takes the form

$$T(x,y) = \phi(x,x_0,c_x)\ \phi(y,y_0,c_y)$$

$$\phi(x,x_0,c) = \frac{1}{\exp(c^{-1})} \exp\left(\frac{1}{(x-x_0)^2 + c}\right) + Ax + B \qquad (4.7)$$

where again A and B are selected so that homogeneous boundary conditions are satisfied. In the example, $x_0 = 0.55$, $y_0 = 0.50$, and $c_x = 0.02$, $c_y = 0.05$. The solution gives very steep gradients near (x_0,y_0). As in the previous example, the solution is substituted into the governing equation, Eq. 4.3, to give a variable heating rate Q(x,y) for the finite element solution. Temperature contours for the exact solution, Eq. 4.7, are presented in Fig. 17.

Finite element solutions were obtained for two remeshes starting from a solution on a uniform mesh. The meshes obtained from remeshing are shown in Figs. 18(a) and 19(a), and the finite element solutions on these meshes are shown in Figs. 18(b) and 719(b). The solution errors are tabulated in Table 3.

Table 3 shows that for the first remesh there was an error reduction of about 15 even though the number of unknowns was reduced by more than one-half. For the second remesh, the number of unknowns are less than 75% of the

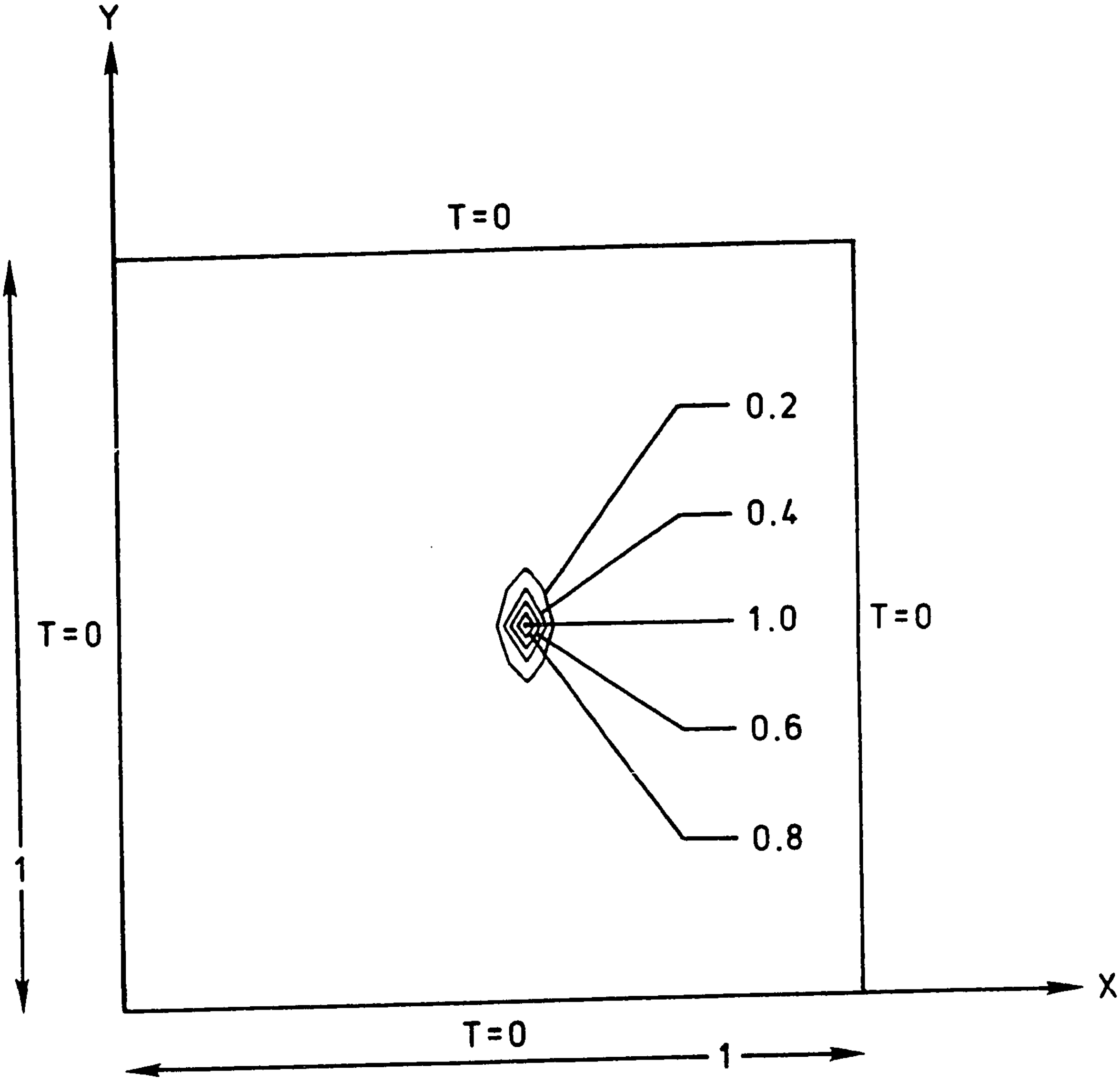
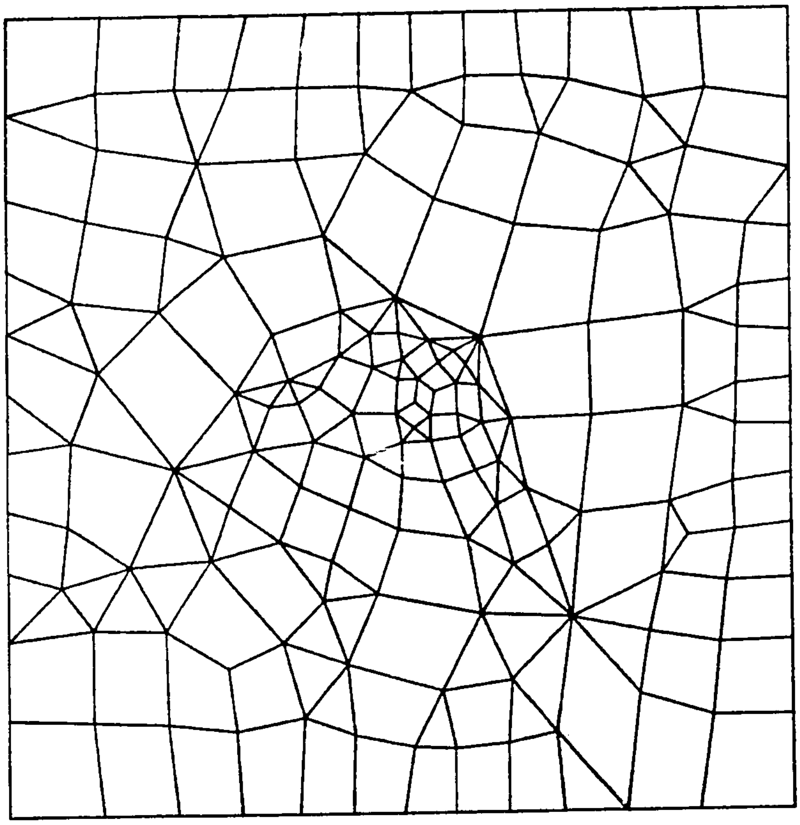
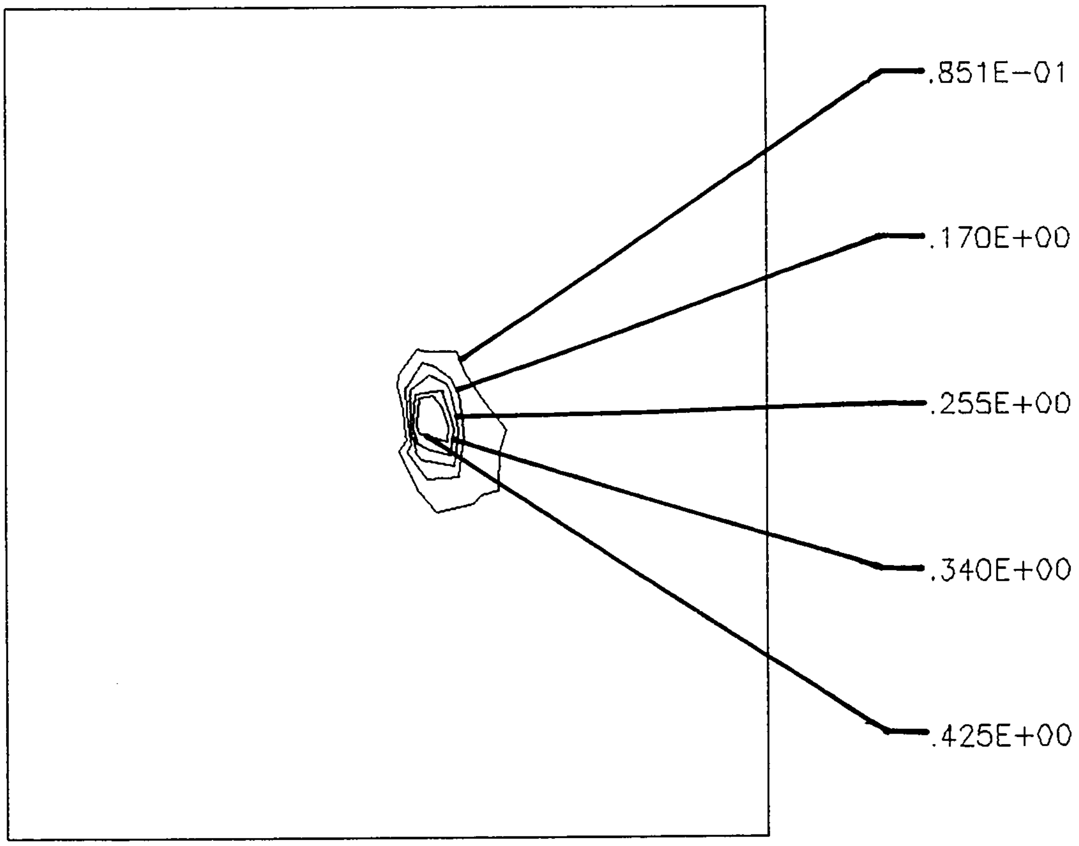


Fig. 17. Problem statement for heat conduction, Example 3.



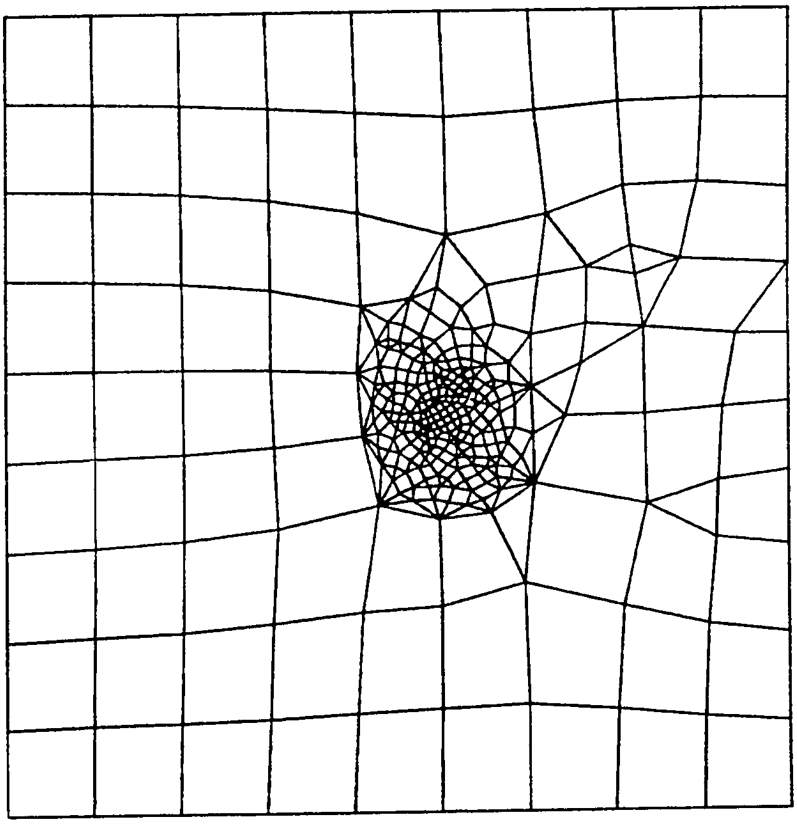
(a) Remesh 1

Fig. 18 Adaptive remesh 1 for Example 3.



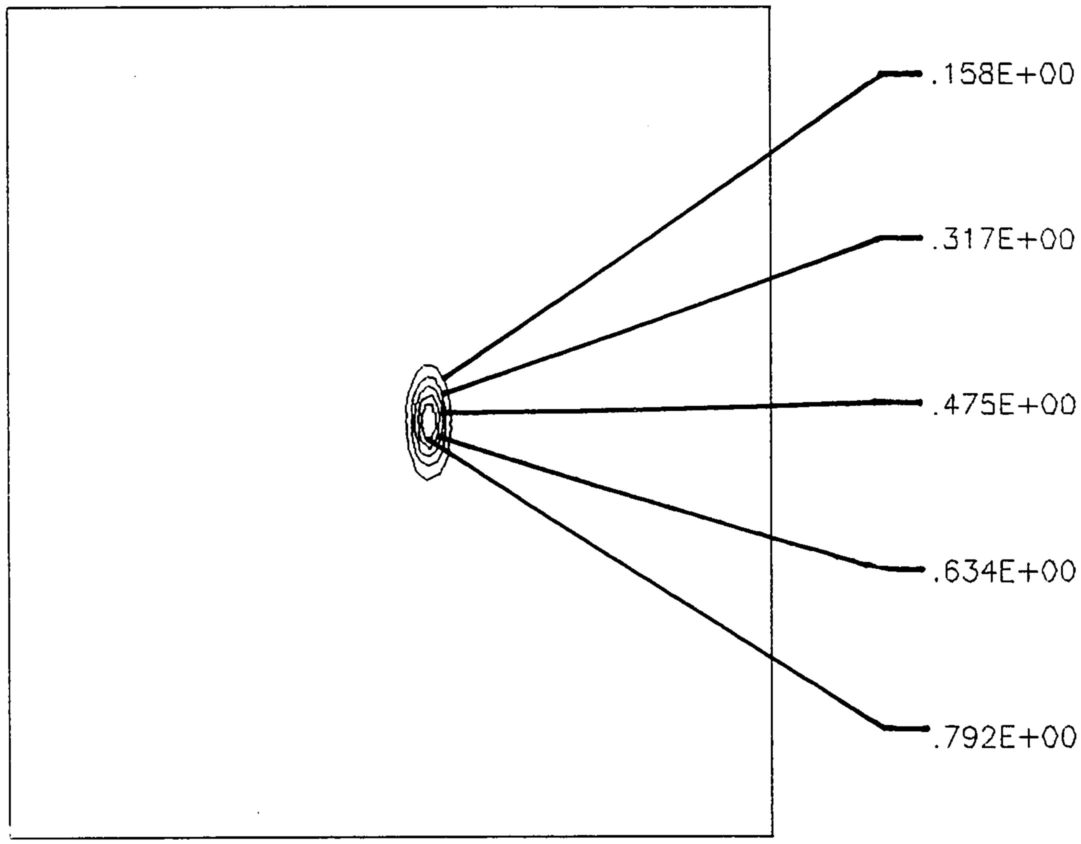
(b) Temperature contours of the finite element solution

Fig. 18 Adaptive remesh 1 for Example 3 (concluded).



(a) Remesh 2

Fig. 19 Adaptive remesh 2 for Example 3.



(b) Temperature contours of the finite element solution

Fig. 19 Adaptive remesh 2 for Example 3 (concluded).

Table 3 Comparative Solution Errors for Example 3

Type	Unknowns	Error, $\| e \|$
Uniform h, (21x21)	441	0.4150
Remesh 1	176	0.02870
Remesh 2	322	0.002868

unknowns in the original mesh, yet the solution error has been reduced by more than two orders of magnitude. The example shows the strong benefits that were obtained by adaptively refining a mesh in regions of steep gradients.

4.2 Closure

For a problem with a smooth solution with no steep gradients, the remeshing approach produced convergence rates of order h2 the same as uniform refinement. Hence, for problems with few gradients the approach offers no advantage . However, for two examples with steep gradients the approach consistently out-performed uniform refinement. In both examples, the remeshing approach gave much faster convergence rates. For these problems, it may be possible to generate regular meshes with proper gradations near the regions with large variations in temperature. Such graded meshes may give faster convergence compared to uniform meshes. But in general, solution features may not exhibit symmetry and a priori knowledge is not available regarding the location of steep gradients. Therefore generation of a graded regular mesh is not a viable task in many situations. On the other hand the remeshing method proved to capture gradients in these examples adaptively and is applicable to any problem in general. Moreover, in both cases there were instances where the error in the solution was reduced even though the number of unknowns was actually smaller than in the previous mesh. This desirable behavior has also been observed by Peraire and Morgan [36] who had previously developed the method using all triangles. The present implementation of the method has some advantages over an all triangle mesh including the fact that fewer elements are required with quadrilaterals.

Chapter 5

GOVERNING EQUATIONS AND SOLUTION ALGORITHM FOR COMPRESSIBLE VISCOUS FLOWS

This chapter presents the governing equations and the solution algorithm for compressible viscous flows. Numerical algorithms [22,23,24,32,36] for solving problems related to high speed flows using unstructured meshes have received attention recently because of their capability for handling arbitrary geometries and complex localized flow features. Taylor-Galerkin [7] and Runge-Kutta [23] schemes proved to be successful finite element algorithms for analyzing compressible flows. Although, these schemes are second-order accurate, because of their explicit time marching nature solution convergence to steady state may be slow for viscous problems. In addition, a two step Taylor-Galerkin [23] scheme was unstable for certain unstructured meshes.

The implicit upwind differencing algorithm proposed by Gnoffo [47] proved to be a robust finite-volume scheme that can run at very high Courant numbers (of the order of 1,000) resulting in faster convergence. Thareja et al. [44] extended this scheme in the context of unstructured meshes and implemented the viscous effects in a finite element sense. Dechaumphai et al. [5] modified this algorithm presented by Thareja et al. [44] by incorporating dimensional quantities and non-uniform boundary conditions. This modified algorithm is used in the current work and is described in this chapter after a formal introduction of the governing equations.

5.1 Governing Equations

The governing equations for a laminar viscous compressible flow are the conservation of mass, momentum and energy equations. These equations are written in conservation form as

$$\frac{\partial\{U\}}{\partial t}+\frac{\partial\{F_1+G_1\}}{\partial x}+\frac{\partial\{F_2+G_2\}}{\partial y} = 0 \tag{5.1}$$

where {U} is the vector of conservation variables, $\{F_1\}$ and $\{F_2\}$ are inviscid flux components; and $\{G_1\}$ and $\{G_2\}$ are viscous flux components.

These vectors are given by

$$\{U\}=\begin{Bmatrix}\rho\\ \rho u\\ \rho v\\ \rho\varepsilon\end{Bmatrix} \tag{5.2}$$

$$\{F_1\}=\begin{Bmatrix}\rho u\\ \rho u^2+p\\ \rho u v\\ (\rho\varepsilon+p)u\end{Bmatrix}\ ;\ \{F_2\}=\begin{Bmatrix}\rho v\\ \rho u v\\ \rho u^2+p\\ (\rho\varepsilon+p)v\end{Bmatrix} \tag{5.3}$$

$$\{G_1\} = -\left\{\begin{matrix} 0 \\ \sigma_{xx} \\ \tau_{xy} \\ u\,\sigma_{xx} + v\,\tau_{xy} - q_x \end{matrix}\right\} \; ; \; \{G_2\} = -\left\{\begin{matrix} 0 \\ \tau_{xy} \\ \sigma_{yy} \\ u\,\tau_{xy} + v\,\sigma_{yy} - q_y \end{matrix}\right\} \qquad (5.4)$$

where ρ is density, u and v are velocity components, ε is the total energy, σ_{xx}, σ_{yy} and τ_{xy} are the stress components, and q_x and q_y are the components of the heat-flux vector. A calorically perfect gas is assumed where

$$p = \rho RT \qquad (5.5)$$

The stress components are given by

$$\sigma_{xx} = \frac{2}{3}\mu(T)\left[2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right] \qquad (5.6)$$

$$\tau_{xy} = \mu(T)\left[\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right] \qquad (5.7)$$

and

$$\sigma_{yy} = \frac{2}{3}\mu(T)\left[2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right] \qquad (5.8)$$

where $\mu(T)$ is the temperature dependent viscosity, computed from Sutherland's law ,

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0}\right)^{3/2} \frac{T_0 + S}{T + S} \qquad (5.9)$$

where subscript 0 refers to a reference value and S is Sutherland's constant.

The components of the heat-flux vector are given by

$$q_x = - k\,(T)\,\frac{\partial T}{\partial x} \tag{5.10}$$

$$q_y = - k\,(T)\,\frac{\partial T}{\partial y} \tag{5.11}$$

where k(T) is the thermal conductivity computed from the viscosity and a constant Prandtl number of 0.72.

This set of governing equations is solved subject to appropriate initial and boundary conditions. The initial conditions consist of specifying the distributions of the conservation variables at time zero. Typical boundary conditions may include:

1. Specifying all conservation variables for supersonic inflow,
2. No-slip boundary condition on a fluid-solid interface ,
3. Specified wall temperature, and
4. Supersonic outflow where the flow variables are free to change.

5.2 Solution Algorithm

The governing equations expressed in equation (5.1) can also be written as

$$\frac{\partial \{U\}}{\partial t} + \frac{\partial \{F_1\}}{\partial x} + \frac{\partial \{F_2\}}{\partial y} = - \frac{\partial \{G_1\}}{\partial x} - \frac{\partial \{G_2\}}{\partial y} \tag{5.12}$$

by combining viscous and inviscid fluxes together. Then equation (5.12) can be expressed as

$$\frac{\partial\{U\}}{\partial t}+\frac{\partial\{F_i\}}{\partial x_i}=-\frac{\partial\{G_i\}}{\partial x_i} \qquad (5.13)$$

in the indicial notation where $\{F_i\}$ is the inviscid flux vector, $\{G_i\}$ is the viscous flux vector, and $\{U\}$ is the conservation variable vector. On direct application of the Green's theorem , equation (5.13) yields the following expression for a typical element (E).

$$\int_{\Omega_E}\frac{\partial\{U\}}{\partial t}d\Omega=\int_{\Omega_E}\left(-\frac{\partial\{F_i\}}{\partial x_i}-\frac{\partial\{G_i\}}{\partial x_i}\right)d\Omega=\int_{\Gamma_E}n_i\,(-\{F_i\}-\{G_i\})\,d\Gamma \qquad (5.14)$$

where Ω_E is the area of the element (E), n_i are the direction cosines and Γ_E is the edge of the element.

Assuming that the conservation variables are constant over the element, equation (5.14) can be written in the form

$$\{\Delta U_E\}=\{U_E\}^{m+1}-\{U_E\}^{m}=\frac{\Delta t}{\Omega_E}(\{IC\}^{m+1}+\{VC\}^{m+1}) \qquad (5.15)$$

where $\{U_E\}$ is the element level conservation variable vector, $\{\Delta U_E\}$ is the vector representing the element level changes in the conservation variables, Δt is the local time step for the element (E) and superscript m is the time step index . {IC} is the element level inviscid flux vector and {VC} is the element level viscous flux vector and are expressed as,

$$\{IC\}^{m+1} = -\int_{\Gamma_E} n_i \{F_i\}^{m+1} d\Gamma = -\int_{\Gamma_E} \{F_n\}^{m+1} d\Gamma \tag{5.16}$$

$$\{VC\}^{m+1} = -\int_{\Gamma_E} n_i \{G_i\}^{m+1} d\Gamma = -\int_{\Gamma_E} \{G_n\}^{m+1} d\Gamma \tag{5.17}$$

where subscript n denotes flux components normal to the element edge. Thus the total element level flux contribution is evaluated by summing the individual contribution from each of the element sides Γ_{es}. Assuming normal fluxes are constant over an element edge, the analytical normal flux vectors $\{F_n\}$ and $\{G_n\}$ are replaced by the numerical normal flux vectors $\{\tilde{F}_n\}$ and $\{\tilde{G}_n\}$ as

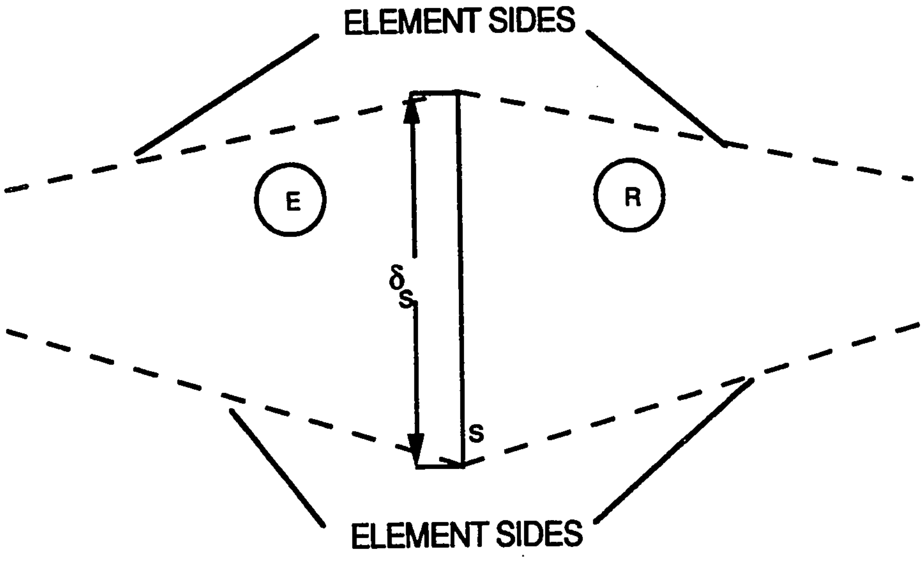
$$\{IC\}^{m+1} = -\sum_{S=1}^{NSIDE} \{\tilde{F}_n\}^{m+1} \delta_S \tag{5.18}$$

$$\{VC\}^{m+1} = -\sum_{S=1}^{NSIDE} \{\tilde{G}_n\}^{m+1} \delta_S \tag{5.19}$$

where NSIDE is the number of sides in the element (E), and δ_S is the length of the side s.

5.2.1 Inviscid solution procedure

Consider the inviscid contribution alone for a typical side s of the element (E) whose length is δ_S which interfaces element E on the left side and element R on the right side as shown in the sketch given below.



Roe's averaging procedure [46] is used to construct the Jacobian matrix $[A_s]$ such that

$$\{F_{nR}\} - \{F_{nE}\} = [A_S]\left(\{U_R\} - \{U_E\}\right) \qquad (5.20)$$

where subscripts R and E denote elements R and E respectively. Matrix $[A_s]$ can be factored as

$$[A_S] = [R]^{-1}[\Lambda]\ [R] \qquad (5.21)$$

where [R] is the eigenvector matrix of $[A_s]$ and $[\Lambda]$ is the diagonal eigen-value matrix of $[A_s]$. The components of each of the matrices in equation (5.21) are given in Appendix C.

Gnoffo [47] used a first-order dissipation term defined as

$$\{D_S\} = [|A_S|]\,(\{U_R\} - \{U_E\}) \tag{5.22}$$

where $[\,|A_S|\,]$ is given by the following expression,

$$[|A_s|] = [R]^{-1}[\,|\Lambda|\,]\,[R] \tag{5.23}$$

where $[\,|\Lambda|\,]$ is the diagonal absolute eigenvalue matrix of $[A_s]$. Using this dissipation term as given by equation (5.22), Gnoffo [47] formulated the numerical normal flux vector associated with the side under consideration as

$$\{\tilde{F}_n\} = \frac{1}{2}[\{F_{nR}\} + \{F_{nE}\} - [|A_s|]\,(\{U_R\} - \{U_E\})] \tag{5.24}$$

The minimum allowable value for the eigenvalues λ_i are restricted according to Yee and Harten [48] and is such that

$$|\lambda_i| = \left\{ \begin{array}{ll} |\lambda_i|, & |\lambda_i| > \epsilon_\lambda \\ 0.5\left(\dfrac{\lambda_i^2}{\epsilon_\lambda} + \epsilon_\lambda\right), & |\lambda_i| < \epsilon_\lambda \end{array} \right\} \tag{5.25}$$

where ϵ_λ is the eigenvalue limiter. In this work a value of 0.5 is used for the ϵ_λ for inviscid problems. Thus for a typical side "s" , the inviscid contribution at the time instant m+1 can be written as

$$\{IC\}_S^{m+1} = -\delta_S \{\tilde{F}_n\}^{m+1} = -\frac{\delta_S}{2}\Big[\{F_{nR}\}^{m+1} + \{F_{nE}\}^{m+1} - [|A_s|]^{m+1}\big(\{U_R\}^{m+1} - \{U_E\}^{m+1}\big)\Big] \tag{5.26}$$

Considering the inviscid contributions alone as expressed in equation (5.18), equation (5.15) yields the following implicit time marching scheme for inviscid flows.

$$\{\Delta U_E\} = -\frac{\Delta t}{2\Omega_E}\sum_{s=1}^{NSIDE} \delta_S \Big[\{F_{nR}\}^{m+1} + \{F_{nE}\}^{m+1} - [|A_s|]^{m+1}\big(\{U_R\}^{m+1} - \{U_E\}^{m+1}\big)\Big] \tag{5.27}$$

The scheme represented by equation (5.27) is implicit because of the terms on the RHS correspond to the m+1 instant of time, and hence the system of equations it represents are coupled. The scheme can be made explicit by taking the value of these terms on the RHS at time m instead of m+1 so that equations become uncoupled. So an explicit time marching scheme can be expressed as

$$\{\Delta U_E\} = -\frac{\Delta t}{2\Omega_E}\sum_{s=1}^{NSIDE} \delta_S \Big[\{F_{nR}\}^{m} + \{F_{nE}\}^{m} - [|A_s|]^{m}\big(\{U_R\}^{m} - \{U_E\}^{m}\big)\Big]$$

and

$$\{U_E\}^{m+1} = \{U_E\}^{m} + \{\Delta U_E\} \tag{5.28}$$

To improve the speed of convergence of the scheme, the implicit time stepping scheme as given in equation (5.27) can be converted into a point-implicit scheme, where a 4x4 matrix system is solved for each element, one element at a time. This conversion is made by taking: (1) the element level conservation variables corresponding to the element (E) at instant m, (2) all the quantities corresponding to elements surrounding (E) from the latest available values, and (3) the flux vector of element (E) at instant m+1, as shown below.

$$\{F_{nE}\}^{m+1} = \{F_{nE}\}^{m} + [|A_s|]^{m+1} \{\Delta U_E\} \tag{5.29}$$

After including the above changes, equation (5.27) can be written as

$$\{\Delta U_E\} = -\frac{\Delta t}{2\Omega_E} \sum_{s=1}^{NSIDE} \delta_S \left[\{F_{nR}\}^{*} + \{F_{nE}\}^{m} + [|A_s|]^{*} \{\Delta U_E\} - [|A_s|]^{*} \left(\{U_R\}^{*} - \{U_E\}^{*}\right)\right] \tag{5.30}$$

where superscript * refers to the latest value available. By taking the underlined term in equation (5.30) to the LHS of the equation we obtain

$$\left[I + \frac{\Delta t}{2\Omega_E} \sum_{s=1}^{NSIDE} \delta_S \, [|A_s|]^{*} \right] \{\Delta U_E\} =$$

$$-\frac{\Delta t}{2\Omega_E} \sum_{s=1}^{NSIDE} \delta_S \left[\{F_{nR}\}^{*} + \{F_{nE}\}^{m} - [|A_s|]^{*} \left(\{U_R\}^{*} - \{U_E\}^{*}\right)\right] \tag{5.31}$$

The above point -implicit scheme is solved using an iterative method which can be considered as a point Gauss-Seidel method where the most current solution is used for the elements surrounding (E). The above scheme is marched in time, and at each time step the 4x4 system of element equations are solved. The L_2-norm of $\{\Delta U_E\}$ values at each time step is computed for the four variables and is used as a measure of convergence. A Courant number of 300 is used for inviscid problems in computing the time step Δt for each element. Once the desired convergence is achieved all element level quantities are converted to nodal quantities.

5.2.2 Explicit Viscous Solution Procedure

The viscous fluxes given by equation (5.19) can be easily accommodated in the explicit inviscid procedure given in equation (5.28) by just adding them to the right hand side of the equation. This explicit viscous solution procedure can be expressed as

$$\{\Delta U_E\} = -\frac{\Delta t}{2\Omega_E}\sum_{s=1}^{NSIDE} \delta s\left[\{F_{nR}\}^m+\{F_{nE}\}^m-\left[|A_s|\right]^m\left(\{U_R\}^m-\{U_E\}^m\right)+2\{G_{ns}\}\right]$$

and

$$\{U_E\}^{m+1} = \{U_E\}^m + \{\Delta U_E\} \qquad (5.32)$$

where $\{G_{ns}\}$ represents the normal viscous flux vector. The evaluation of the numerical viscous fluxes involves computation of first derivatives of primitive variables like the components of velocity and the temperature at the nodes. In the cell-centered scheme discussed above, the primitive variables are assumed to be piecewise constant, that is, they are assumed to be constant for an element. So a bilinear variation in quadrilateral elements and a linear variation in triangular elements are assumed for the first derivatives, in order to compute them at the nodes. Under these assumptions, the first derivative of a variable, for example, derivatives of temperature T at any point in the element (E) can be approximated as,

$$\frac{\partial T}{\partial x} = [N]\left\{\frac{\partial T}{\partial x}\right\} \qquad (5.33)$$

$$\frac{\partial T}{\partial y} = [N]\left\{\frac{\partial T}{\partial y}\right\} \qquad (5.34)$$

where [N] is the shape functions and $\left\{\frac{\partial T}{\partial y}\right\}$ is the nodal first derivative vector of element (E). By multiplying the LHS of equation (5.33) by {N} and integrating by parts we obtain

$$\int_{\Omega_E} \frac{\partial T}{\partial x}\{N\}\, d\Omega = \int_{\Gamma_E} n_x T\{N\}\, d\Gamma - \int_{\Omega_E} T\left\{\frac{\partial N}{\partial x}\right\} d\Omega \tag{5.35}$$

where n are the direction cosines. Substituting for $\left\{\frac{\partial T}{\partial x}\right\}$ using equation (5.33), the LHS becomes

$$\begin{aligned} \text{LHS} &= \int_{\Omega_E} \{N\}[N]\left\{\frac{\partial T}{\partial x}\right\} d\Omega = \int_{\Omega_E} \{N\}[N]\, d\Omega \left\{\frac{\partial T}{\partial x}\right\} \\ &= [M_L]\left\{\frac{\partial T}{\partial x}\right\} \end{aligned} \tag{5.36}$$

where $[M_L]$ is the lumped mass matrix approximation for the integral. By equating equations (5.35) and (5.36) we have the following expression

$$[M_L]\left\{\frac{\partial T}{\partial x}\right\} = \left[\int_{\Gamma_E} n_x T\{N\}\, d\Gamma - \int_{\Omega_E} T\left\{\frac{\partial N}{\partial x}\right\} d\Omega\right] \tag{5.37}$$

To compute the first derivative at a particular node K, an assembly operation has to be performed where contributions from all the elements that have K as a common node have to be summed. This operation results in the following relation for the first derivative at node K.

$$\left.\frac{\partial T}{\partial x}\right|_K = \frac{1}{M_{LK}}\left[BT_{xK} - \sum_{E=1}^{NE} \int_{\Omega_E} T_E \left\{\frac{\partial N}{\partial x}\right\} d\Omega \right] \qquad (5.38)$$

where M_{LK} is the coefficient of the global (assembled) lumped mass matrix for node K, BT_{xK} is the assembled boundary term including contributions from all the elements NE surrounding node K, and T_E is the temperature of the element E. Similarly we can write

$$\left.\frac{\partial T}{\partial y}\right|_K = \frac{1}{M_{LK}}\left[BT_{yK} - \sum_{E=1}^{NE} \int_{\Omega_E} T_E \left\{\frac{\partial N}{\partial y}\right\} d\Omega \right] \qquad (5.39)$$

Since the partial derivatives associated with the viscous fluxes are the derivatives of primitive variables, the following conversion is required in terms of ΔU_E, i.e. the conservative variables.

$$\Delta u_E = \frac{\left(\Delta(\rho\, u)_E - u_E \Delta \rho_E \right)}{\rho_E} \qquad (5.40)$$

$$\Delta v_E = \frac{\left(\Delta(\rho\, v)_E - v_E \Delta \rho_E \right)}{\rho_E} \qquad (5.41)$$

$$\Delta T_E = \frac{\gamma}{c_P}\left[\begin{array}{l} \left(\frac{u^2+v^2}{\rho} - \frac{\varepsilon}{\rho}\right) \Delta(\rho\,)_E - \frac{u}{\rho}\Delta(\rho\, u)_E \\ \qquad - \frac{v}{\rho}\Delta(\rho\, v)_E + \frac{1}{\rho}\Delta(\rho\, \varepsilon)_E \end{array} \right] \qquad (5.42)$$

Now equation (5.32) is solved explicitly at each time step and marched in time until the desired convergence over the error norm is obtained. This explicit scheme for viscous flows was found to be unstable at high Courant numbers and hence resulted in slower convergence compared to the explicit inviscid procedure.

5.2.3 Point-implicit viscous solution procedure

The point-implicit scheme for inviscid flows as represented by equation (5.31) can be extended to viscous flows to obtain faster convergence in the solution procedure compared to the explicit scheme discussed above. This extension is possible by breaking the element level viscous flux vector into two parts as shown below. This procedure is suggested by Thareja et al. [44].

$$\{G_{nS}\}^{m+1} = \{G_{nS}\}^{*} + [B_S]\,\{\Delta U_E\} \qquad (5.43)$$

where superscript * refers to the latest values available, and the matrix $[B_S]$ is the viscous analogue of the matrix $[A_S]$. The components of the matrix $[B_S]$ are given in Appendix D. Due to the complexities involved in evaluating $\{G_{nS}\}^{*}$, in the current implementation it is assumed that

$$\{G_{nS}\}^{*} = \{G_{nS}\}^{m} \qquad (5.44)$$

Incorporating equations (5.43) and (5.44) in equation (5.31) we obtain the following expression.

$$\left[I + \frac{\Delta t}{2\Omega_E} \sum_{s=1}^{NSIDE} \delta_S \, [|A_s|]^* \right] \{\Delta U_E\} =$$

$$- \frac{\Delta t}{2\Omega_E} \sum_{s=1}^{NSIDE} \delta_S \left[\{F_{nR}\}^* + \{F_{nE}\}^m - [|A_s|]^* \left(\{U_R\}^* - \{U_E\}^* \right) + 2\left(\{G_{nS}\}^m + [B_S]\{\Delta U_E\} \right) \right] \tag{5.45}$$

By taking the underlined term on the RHS of equation (5.45) to the LHS we have the following point-implicit scheme for viscous flow.

$$\left[I + \frac{\Delta t}{2\Omega_E} \sum_{s=1}^{NSIDE} \delta_S \left([|A_s|]^* + 2[B_S] \right) \right] \{\Delta U_E\} =$$

$$- \frac{\Delta t}{2\Omega_E} \sum_{s=1}^{NSIDE} \delta_S \left[\{F_{nR}\}^* + \{F_{nE}\}^m - [|A_s|]^* \left(\{U_R\}^* - \{U_E\}^* \right) + 2 \, \{G_{nS}\}^m \right] \tag{5.46}$$

Because of the assumption given in equation (5.44) the solution algorithm is like a point Jacobi method for the viscous contributions unlike for inviscid contributions where the algorithm is like a point Gauss-Seidel method [44]. Higher Courant numbers are permissible with this point-implicit scheme for viscous flows compared to the explicit viscous scheme but yet are much smaller than that used for inviscid procedures. In the current work a Courant number of 3.0 is used for viscous flows. In order to remove the artificial diffusion due to the upwinding, the eigenvalue limiter ϵ_λ is made very small (1.E-07) within the boundary layer.

5.2.4 Boundary condition treatment

For a supersonic inflow boundary, $\{U_R\} = \{U_{SP}\}$ is employed where subscript SP refers to specified boundary conditions . For a supersonic outflow boundary $\{U_R\} = \{U_E\}$ is employed. For an inviscid flow the following set of wall boundary conditions are used,

$$p_R = p_E$$
$$u_{nR} = -u_{nE}$$
$$u_{tR} = u_{tE}$$
$$\rho_R = \rho_E$$

where u_n is the normal velocity, and u_t is the tangential velocity. For a no-slip boundary for solid-fluid interface in a viscous problem the following set of conditions are imposed.

$$p_R = p_E$$
$$u_R = -u_E$$
$$v_R = -v_E$$
$$T_R = T_{SP}$$

The above set of boundary conditions impose the normal pressure gradient to be zero which is a fairly reasonable assumption for most boundary layer flows.

5.3 Closure

Input data for the computer program developed based on the numerical algorithm discussed above include properties of the fluid, connectivity information for the elements in the mesh, nodal coordinates, nodal initial conditions and the boundary conditions. Connectivity information for the elements is the global numbers of the nodes that are associated with each element. For triangular elements, the number for the fourth node is entered as zero. The boundary conditions are specified for each of the boundary segments joining any two consecutive nodes on the boundary in counter-clockwise manner. The initial front information given by the remeshing program can be used for this purpose. A boundary condition code is set up for each of the boundary conditions described in section 5.1.

Although the viscous terms in the algorithm are formally second-order accurate (by the assumption that the first derivatives of the primitive variables are linear in an element), the overall numerical scheme is not second-order accurate since the the convective terms are formally first-order accurate. A grid refinement study on a simple viscous problem with a flow over a flat plate showed that the order of accuracy of the scheme for a viscous flow with no strong inviscid effects is about 1.5 (details are given in appendix E). Research is underway to implement formally second-order accurate upwind algorithms in the finite element context. These schemes may require less number of nodes and be more accurate compared to the first-order scheme, but questions regarding their stability are yet to be answered. On the otherhand the point-implicit algorithm is stable at large Courant numbers (of the order of 1000) for inviscid flows and residuals on the conservation variables reduce to machine zero within 500 time steps. The maximum Courant number used for viscous flows is 3. In this work it was observed that the residuals of the conservation variables reduced 5 to 6 orders of magnitude within about 10,000 time steps for a typical viscous problem. The point-implicit scheme for the viscous flows took 0.000612 CPU seconds per node per time step on the CRAY-YMP machine. The results obtained for the flow problems using this program are discussed in the next chapter.

Chapter 6

ADAPTIVE REMESHING FOR HIGH SPEED COMPRESSIBLE FLOWS

The adaptive remeshing method is employed to solve inviscid and viscous problems for high speed compressible flows. Comparisons are made with either analytical or experimental results. The discussion of the results focusses on the adaptation procedures and the method of mesh generation. The point-implicit algorithm described in the preceding chapter is used for the flow analysis.

6.1 Inviscid Shock Reflection

This is an inviscid flow problem that is solved to demonstrate the adaptation of the method to shocks. The problem statement is given in Fig. 20. A Mach 3 flow intersects another supersonic flow coming at an angle producing an inviscid shock that reflects on an inviscid wall. This is a classical test problem referred to as the "Collela" problem in the literature [49].

The computational domain is a rectangle with dimensions of 4x1 and the boundary conditions are shown in Fig. 20. Two analysis sequences were performed for this problem, one using a uniform refinement starting from a uniform mesh consisting of 400 elements and the other using adaptive remeshing, starting from a uniform mesh consisting of 1600 elements.

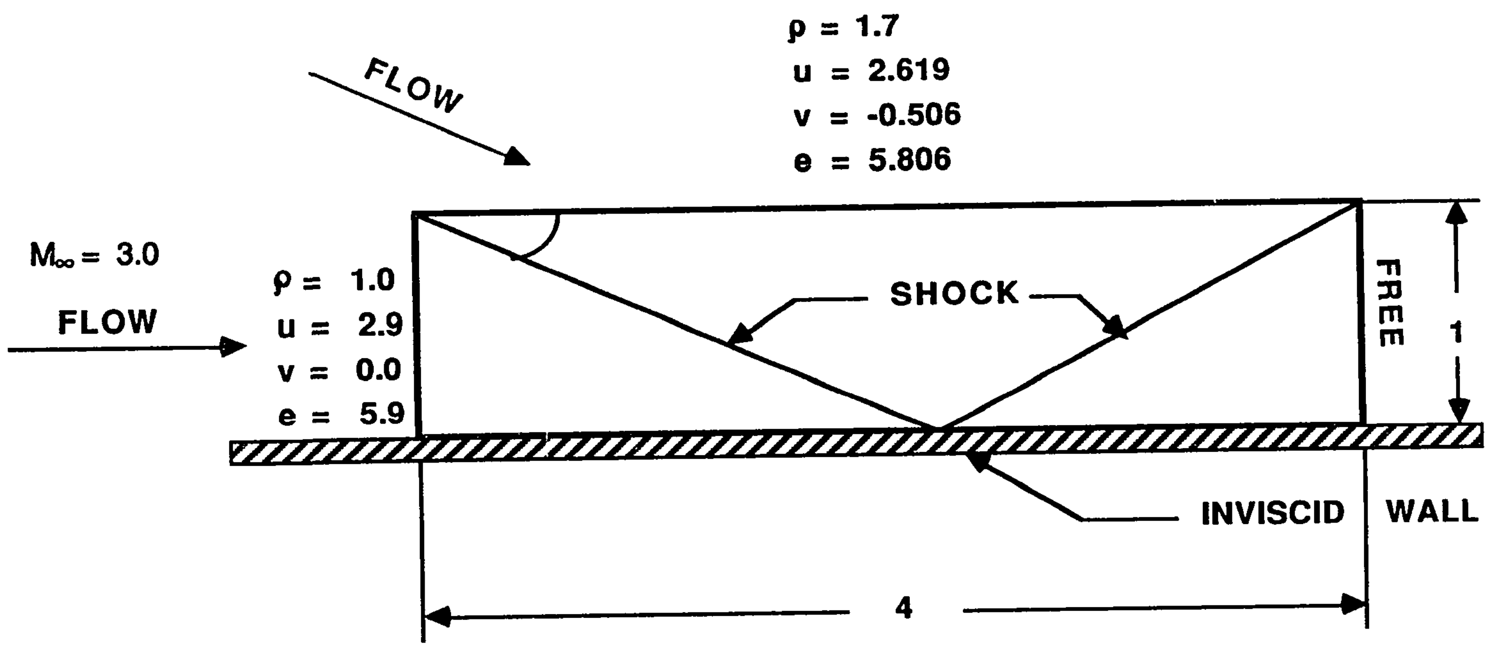
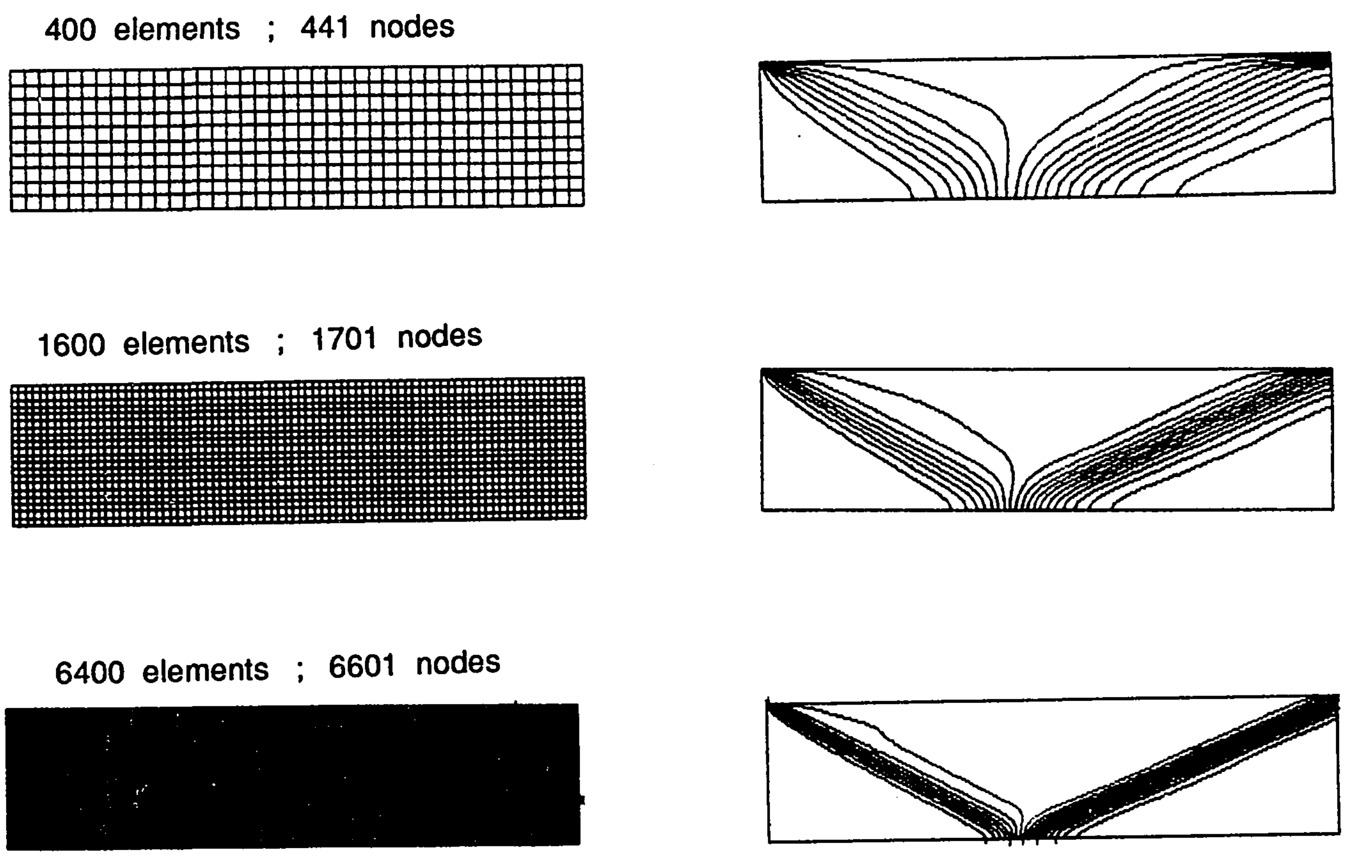


Fig. 20. Problem statement for the inviscid shock reflection.

Fig. 21 shows the uniform h refinement and the corresponding finite element solution for the density distribution. The relatively large thickness of the shock even in the most refined mesh occurs because the solution algorithm is first-order accurate. Thareja et al. [44] observed a similar result with the same algorithm and showed a higher order extension would reduce the thickness of the shock. However, they observed that such an extension would give rise to convergence related problems when applied to viscous flows. Since the objective of the current research is to analyze viscous flows, the first order algorithm is used for all analyses.

Starting from the solution on the uniform mesh of 1600 elements, adaptive remeshing was performed. Initially, density was used as an error indicator. The finite element solution had a steep gradient of density at the core of the shock, and hence second derivatives were small in this region. When the principle of equi-distribution of error was employed in computing the spacings, bigger spacings were computed at the core of the shock and hence bigger elements resulted in the core of the shock as shown in Fig. 22a. This mesh in turn produced a more diffused solution near the shock, and subsequent adaptation was polluted.

To rectify this problem a concept of "multiple indicators" was used where a combination of flow variables was used to compute the spacings required for mesh generation. In this concept, the mesh generation parameters are computed independently based on two or more flow variables and on any node on the background mesh, the set of mesh generation parameters with the smallest h_2 value is taken for adaptation. Combinations of density and Mach number and density and temperature were attempted in generating meshes.


400 elements ; 441 nodes

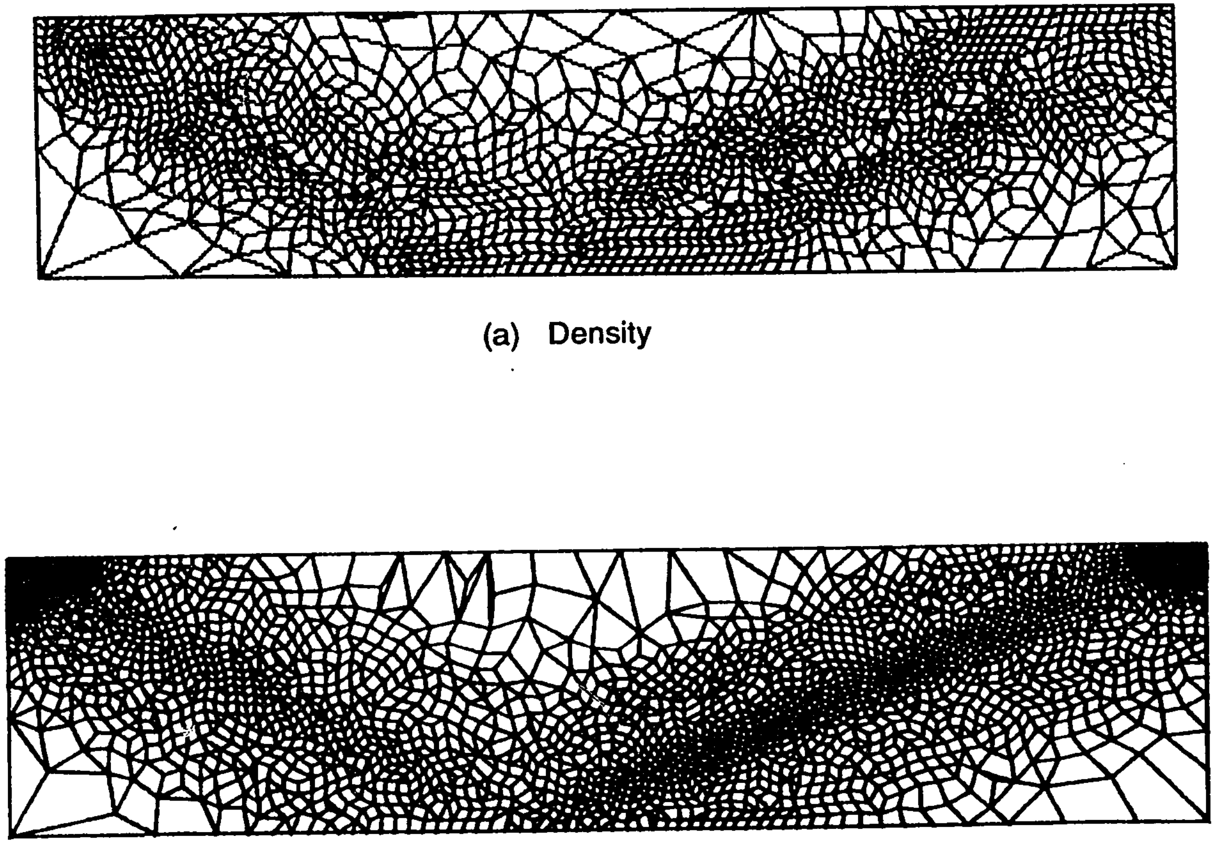
1600 elements ; 1701 nodes

6400 elements ; 6601 nodes

(a) Uniform meshes

(b) Density distributions

Fig. 21. Uniform refinement for the inviscid shock reflection.



(a) Density

(b) Density + magnitude of the gradient of density

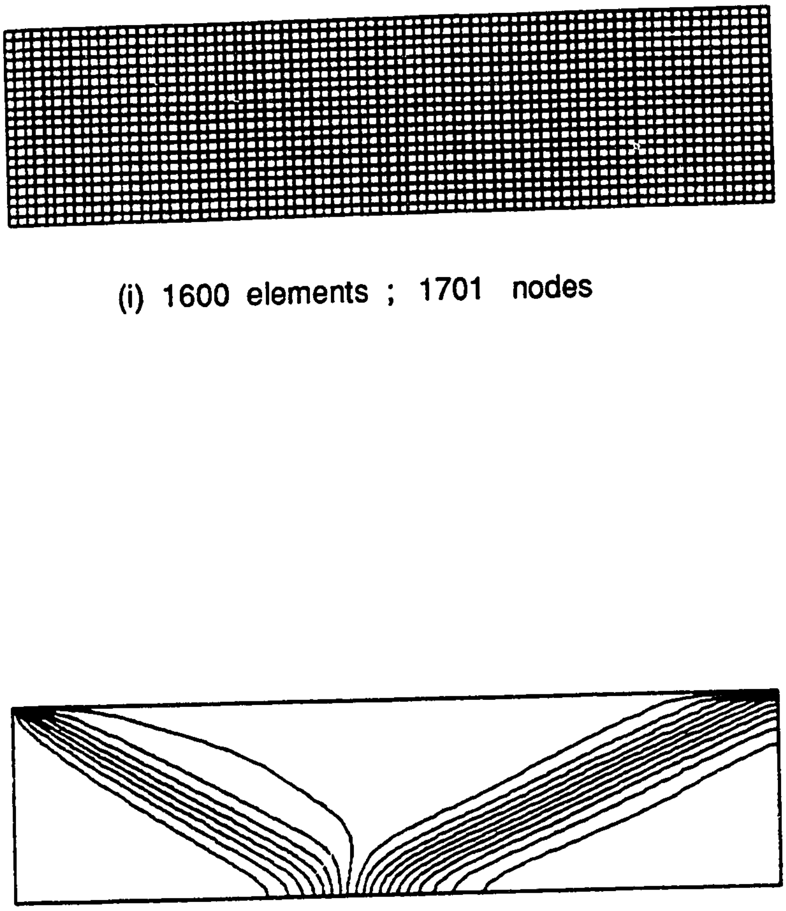
Fig. 22. Influence of the error indicators on remeshing for inviscid shock reflection.

These combinations resulted in similar looking meshes as in Fig. 22a, since the Mach number and temperature distributions showed similar behavior near the shock as the density distribution. A combination of density and the absolute value of the gradient of density as multiple indicators resulted in a mesh as shown in Fig. 22b which rectified the problem with using density alone as an indicator. This result may be due to the distribution of the absolute value of the gradient of density near the core of the shock being different from that of density.

Two successive adaptive remeshes were created with 2809 elements and 4466 elements respectively consisting of predominantly quadrilaterals. The adaptive remeshes and the corresponding finite element solution for density are shown in Fig. 23. Table 4 gives a comparison of adaptive remeshing with the uniform refinement in terms of an L_2 - norm of the error in density. The L_2 - norm is computed from an exact solution obtained from oblique shock relations and the finite element solution. Results show with adaptive remeshing the error was reduced less than that of uniform refinement with one-third less number of unknowns.

6.2 Hypersonic boundary layer flow

This problem is solved to demonstrate the remeshing capability for viscous adaptation. The problem statement is given in Fig. 24. The inflow profiles were obtained from a boundary layer solution [50]. This solution at the inflow plane was obtained at a distance away from the leading edge such that the leading edge shock effects are eliminated from the computational domain.The computational domain includes a laminar boundary layer region and an undisturbed inviscid region.

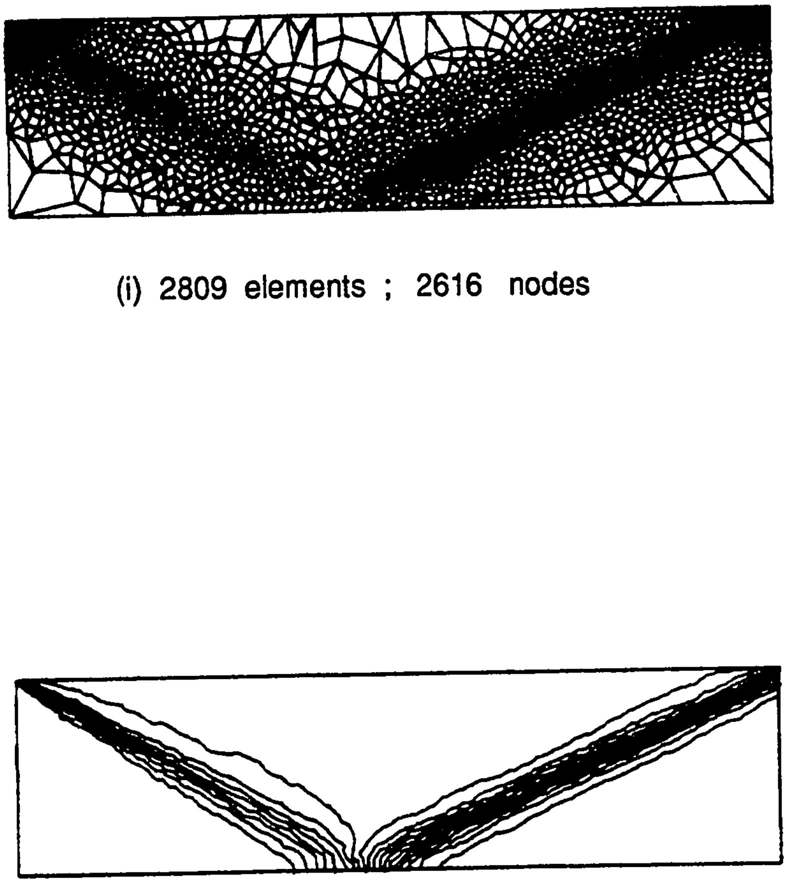


(i) 1600 elements ; 1701 nodes

(ii) Density distribution

(a) Initial mesh

Fig. 23 Adaptive remeshing for the inviscid shock reflection.

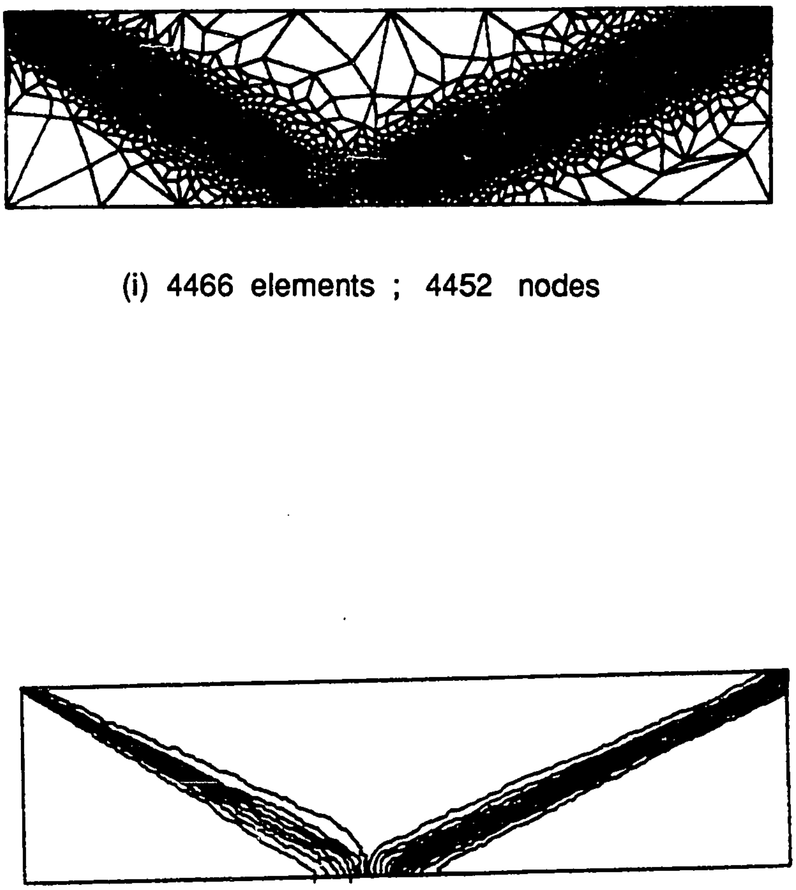


(i) 2809 elements ; 2616 nodes

(ii) Density distribution

(b) First remesh

Fig. 23 Adaptive remeshing for the inviscid shock reflection (continued).



(i) 4466 elements ; 4452 nodes

(ii) Density distribution

(c) second remesh

Fig. 23 Adaptive remeshing for the inviscid shock reflection (concluded).

Table 4 Comparative Solution Errors for Inviscid Shock Reflection

Uniform h Refinement

Mesh	Nodes	h	Error, $\|\| e \|\|$ in Density
1	451	0.1	0.43124
2	1701	0.05	0.33011
3	6601	0.025	0.25164

Adaptive Remeshing

Mesh	Nodes	Error, $\|\| e \|\|$ in Density
1(uniform)	1701	0.33011
2	2616	0.29357
3	4452	0.23822

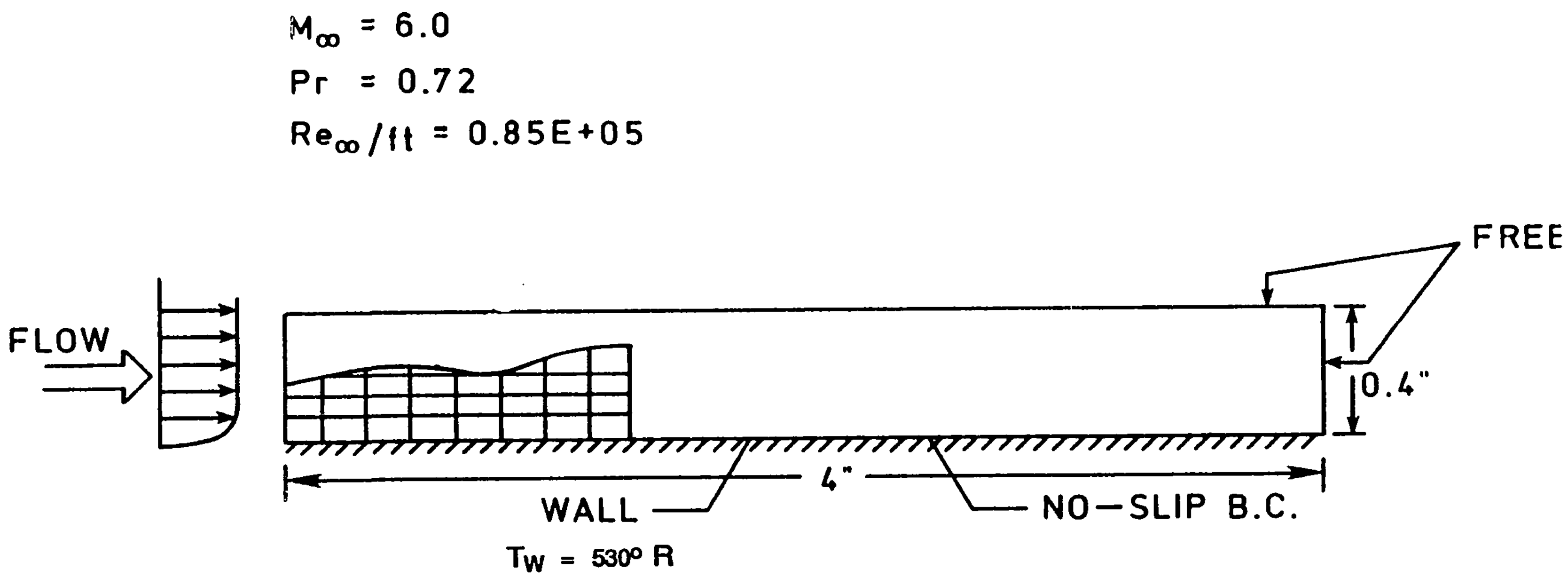
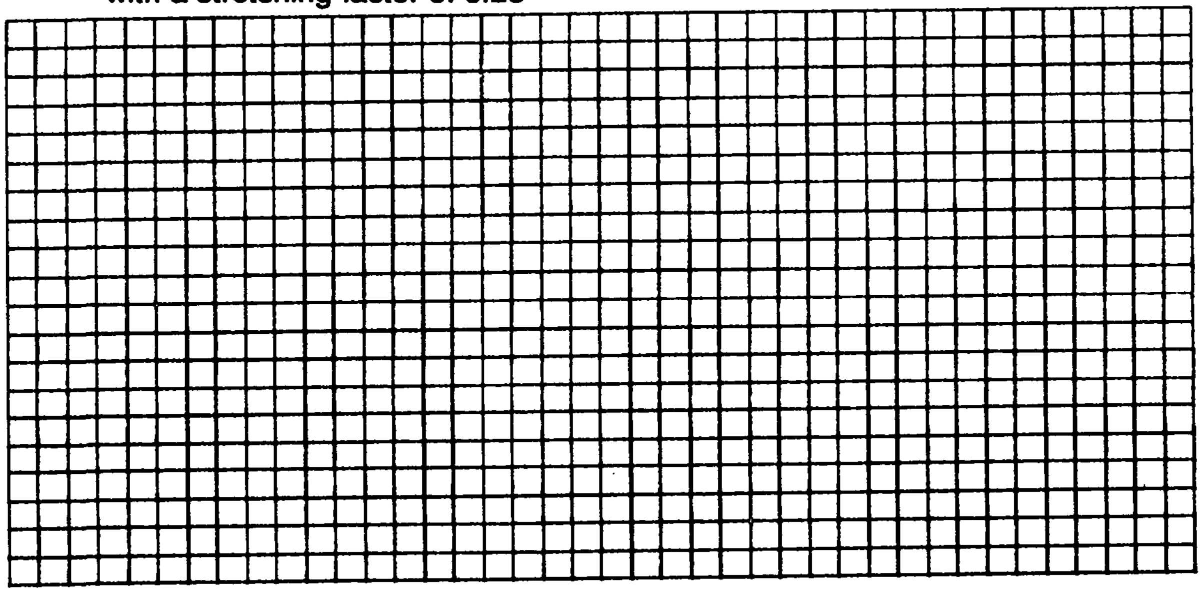


Fig. 24. Problem statement for hypersonic boundary layer flow

A fairly uniform mesh consisting of 800 elements was used as an initial mesh. Fig. 25 shows this mesh which is scaled in the y direction. The inlet profiles were input as initial conditions throughout the domain. The finite element solution was obtained with a Courant number of 3. The solution converged within 1200 iterations. Though this problem appears to be simple, it involves important flow features such as a very high temperature gradient near the wall. The heat flux distribution(magnitude of the heat flux vector) normal to the wall at the inlet, is shown in Fig. 26a. The shearing stress , τ_{XY}, distribution normal to the wall at the same section is shown in Fig. 26b. The converged solution on this mesh gave a wall heat flux distribution as shown in Fig. 27. The finite element solution on this mesh does not agree well with the boundary layer solution [50]. The shearing stress τ_{XY} was computed at every node, and this information was used to identify the boundary layer edge in the next mesh by making use of the fact that the shear stresses are predominant in the boundary layer. It can be viewed from Fig. 26b, as the boundary layer edge is reached the shear stress approaches zero. A limiting value of 1.0E-07 for the numerical shear stress was used to identify the boundary layer region. Again the derivatives in the shear stress computation were evaluated by the numerical procedure given in Appendix B. After two successive remeshings the final mesh as shown in Fig. 28 was obtained. This mesh is again scaled in the y direction so that the thin elements near the wall are visible. Absolute velocity was used as an error indicator for remeshing in the boundary layer. The number of wall nodes in this mesh reduced compared to the initial mesh and so did the number of nodes and the number of elements in the entire domain. While the boundary layer region was discretized by structured remeshing, the inviscid region was discretized in an unstructured manner using density as an indicator.

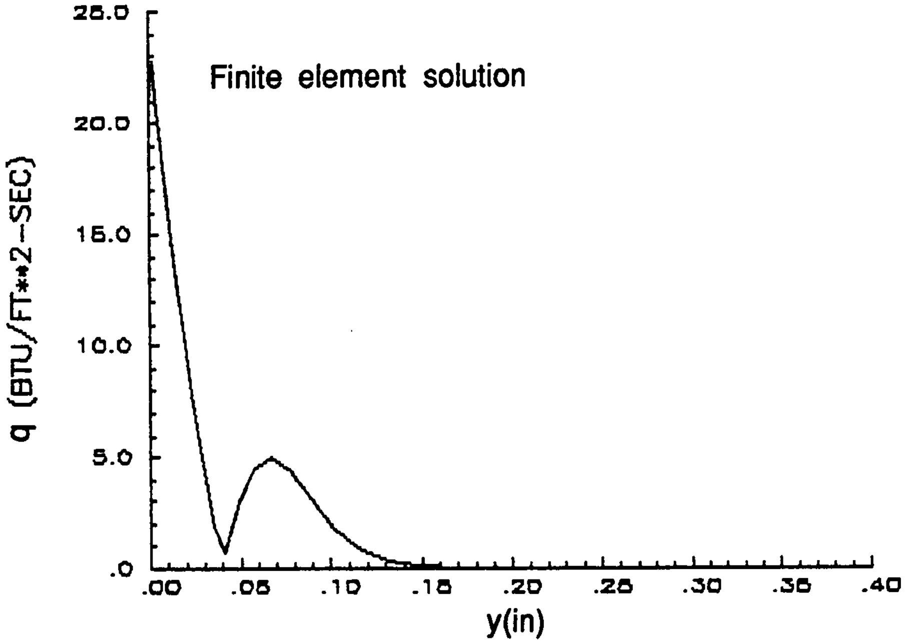


800 elements and 861 nodes.

Mesh stretched uniformly in the y direction

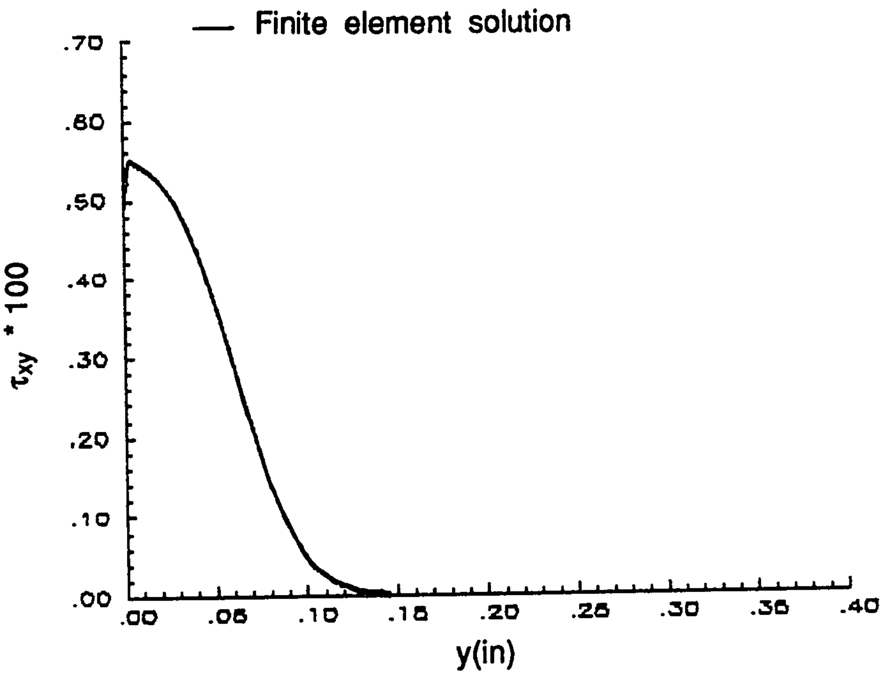
with a stretching factor of 5.25

Fig. 25. Initial mesh for the boundary layer problem.



(a) Heat flux distribution

Fig. 26 Viscous fluxes normal to the wall at the inflow section of the flat plate.



(b) Shear stress distribution

Fig. 26 Viscous fluxes normal to the wall at the inflow section of the flat plate.

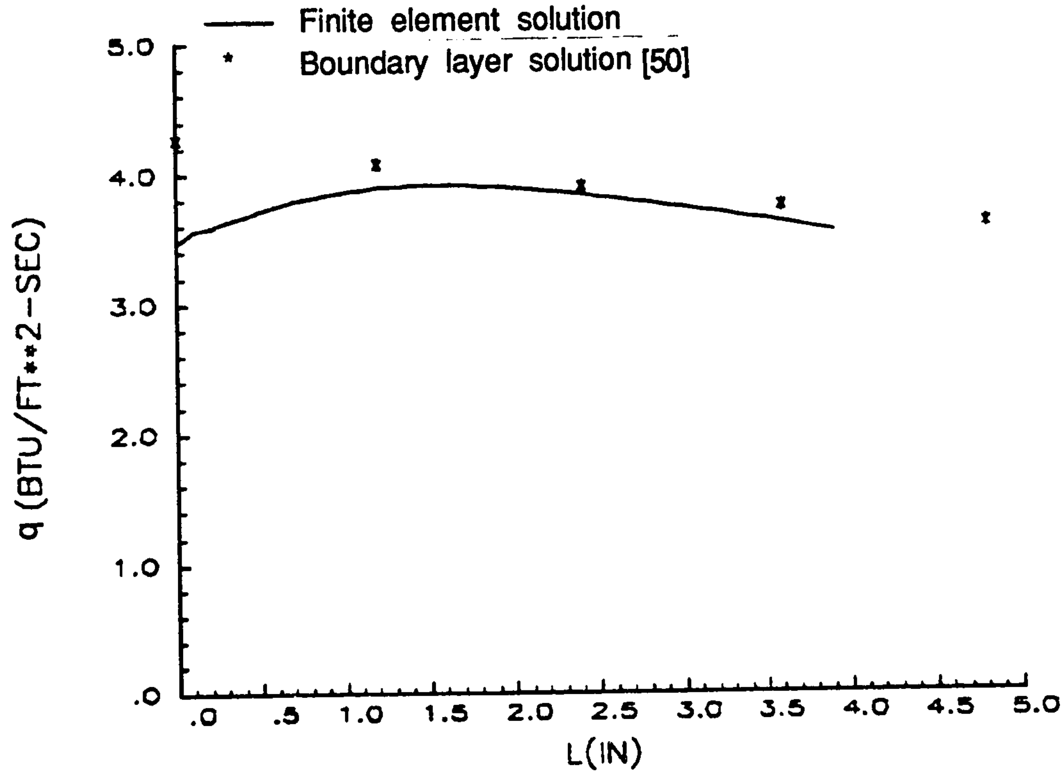


Fig. 27. Wall heat flux distribution on the initial mesh.

746 elements and 797 nodes.

Mesh stretched in the y direction with a stretching factor of 10.45

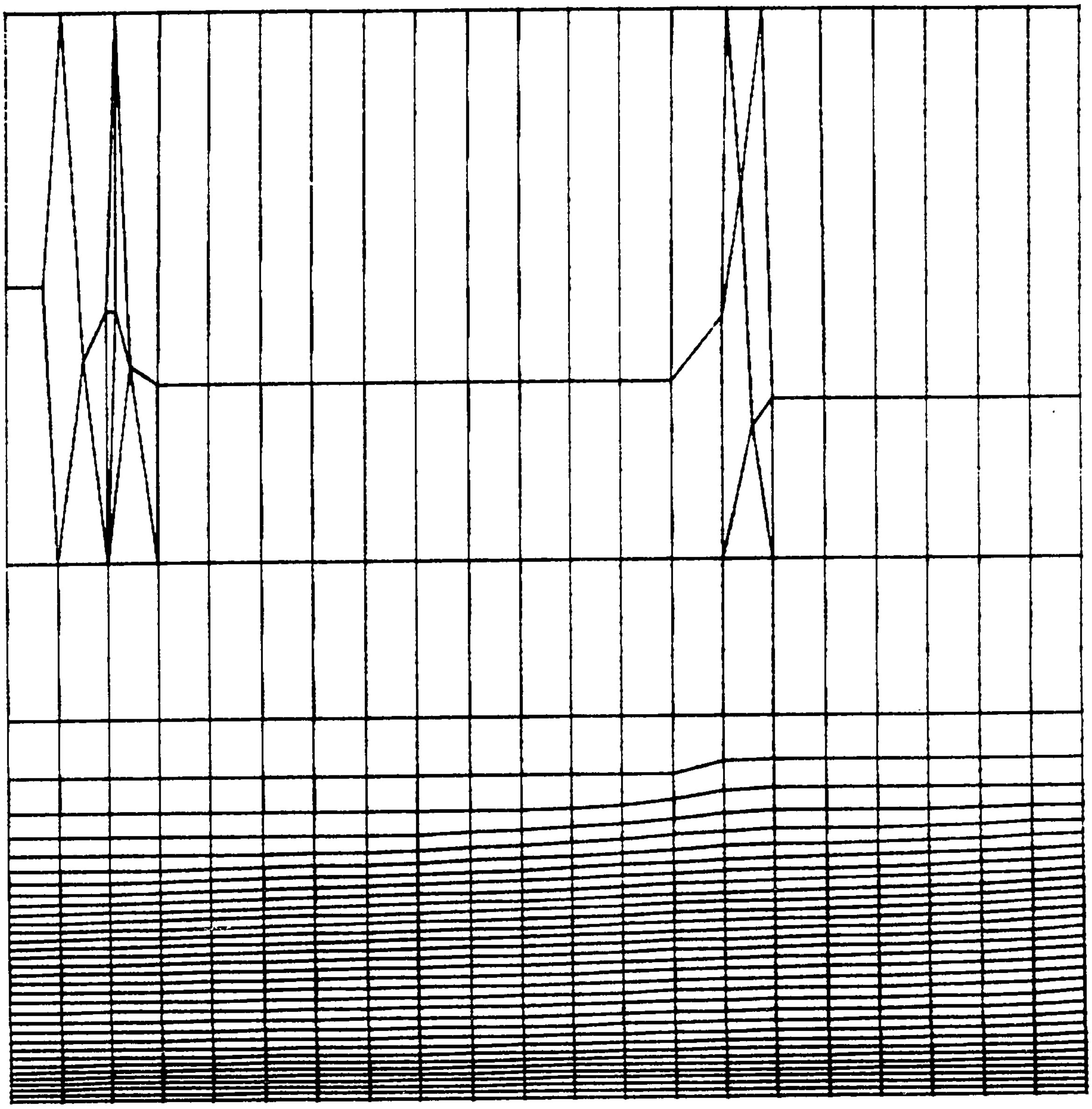


Fig. 28. Stretched final remesh for the boundary layer problem.

The mesh without the scaling, shown in Fig. 29, looks more or less uniform in the inviscid region except for a few triangles that were created as transition elements to accommodate a small change in the spacing resulted from numerical inaccuracy.

The wall heat flux distribution from a converged finite element solution on the final mesh is shown in Fig. 30. The finite element solution compares well with the boundary layer solution. As a matter of fact the remeshing improved the solution quality with respect to the wall heat flux distribution on the initial mesh as shown in Fig. 27 and yet reduced the number of nodes from 841 to 746 in this viscous problem.

6.3 Hypersonic flow over a compression corner

6.3.1. Problem description

The adaptive remeshing capability to predict flow details for a viscous problem with strong inviscid interaction is illustrated by solving the hypersonic flow over a compression corner. This kind of interaction is common in the design of control surfaces for high speed vehicles such as the aerospace plane. The flow features in a hypersonic flow over a 15^0 compression corner are shown in Fig. 31. This problem has been experimentally investigated by Holden [51]. Inflow at Mach 11.68 interacts with a sharp leading edge at zero angle of attack producing a weak shock due to the displacement thickness of the boundary layer. The boundary layer separates ahead of the compression corner due to a strong adverse pressure gradient and reattaches on the ramp. The compression fan generated in the separation region eventually coalesces to form a strong induced shock. This shock interacts with the leading edge shock to produce a resultant shock, an expansion fan and a shear layer.

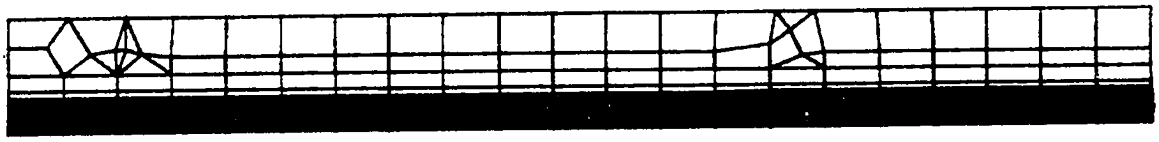


Fig. 29. Final remesh for the boundary layer problem.

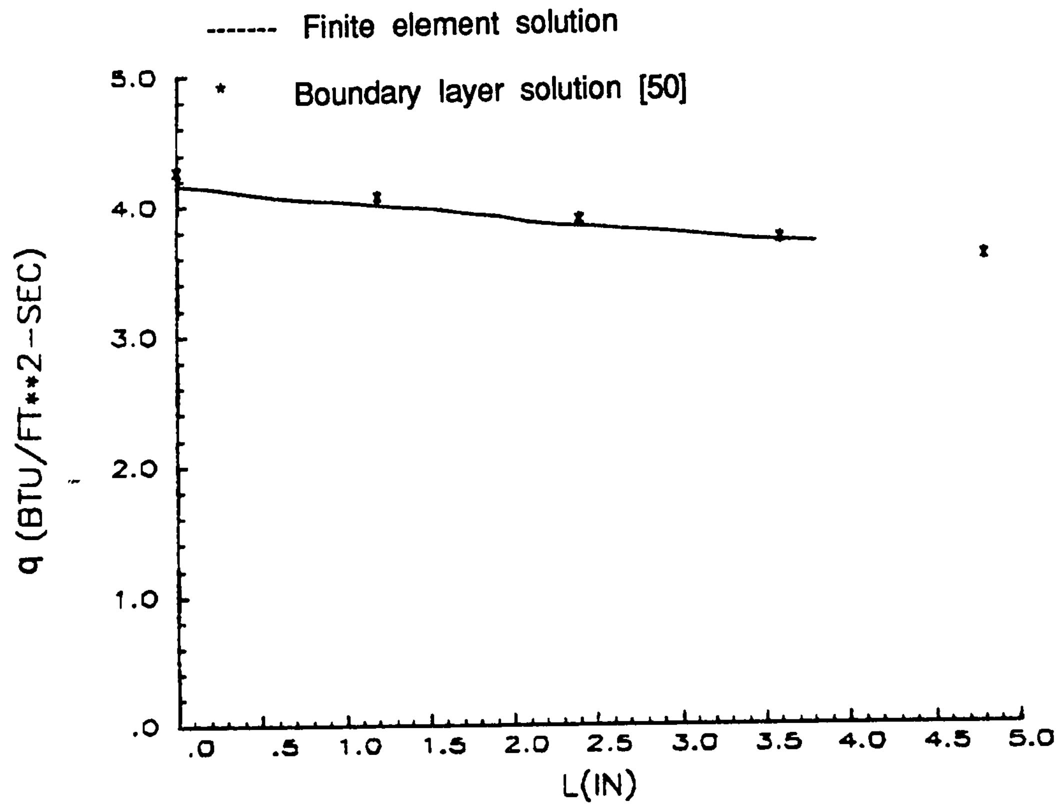


Fig. 30. Wall heat flux distribution on the final remesh.

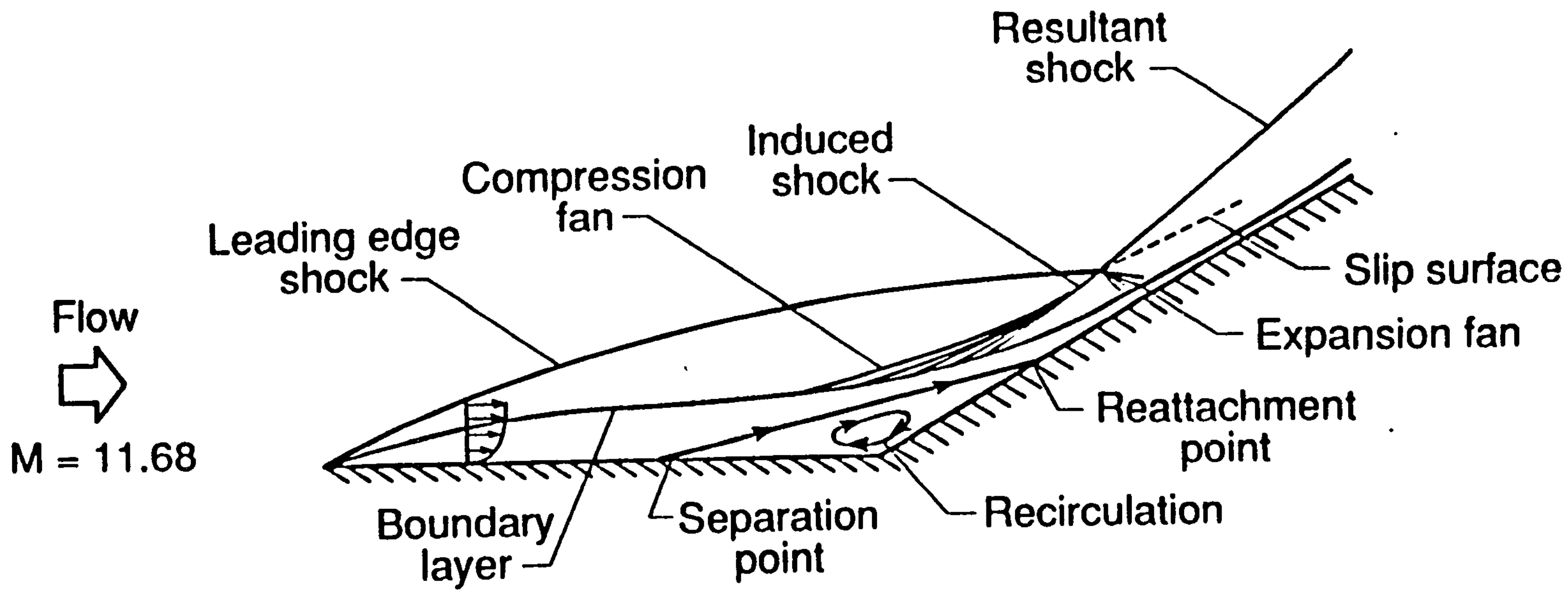


Fig. 31. Flow description for the hypersonic flow over a compression corner.

Ramakrishnan et al. [52] have solved the problem by dividing the entire flow domain into three regions. These three regions are: (1) the sharp leading edge section, (2) the flat plate section, and (3) the ramp section. The results predicted by Ramakrishnan et al. were in good agreement with the strong interaction theory [53] in region 1 and with the experimental results of Holden [54] in region 3. The region of interest among the three discussed above is region 3 where the boundary layer separates and thickens due to the coupling with the strong induced shock and thins down over the ramp where the gradients become larger. This region requires the solution of the full Navier-Stokes equations, and it has been chosen for the adaptive analysis. This region starts at about the midpoint of the flat plate section and goes to the location of shock-shock interaction on the ramp. The inlet profiles for this region are obtained from the results of Ramakrishnan et al. [52]. This inlet section is chosen sufficiently far upstream such that the flow separation effects do not influence the inflow.

The problem statement is shown in Fig. 32. An initial mesh was constructed with 5283 nodes and 5142 elements. A structured mesh is constructed to a height of 0.4 from the wall and the rest of the region is discretized in an unstructured manner. User specified values on a crude background mesh for the mesh generation parameters are used to generate the entire mesh. This initial mesh is shown in Fig. 33. The interpolated values at the inlet section from the solution of Ramakrishnan et al. are shown in Fig. 34.

A finite element solution was obtained after 5000 iterations until the L_2-norm of the residuals of all the conservation variables reduced at least three

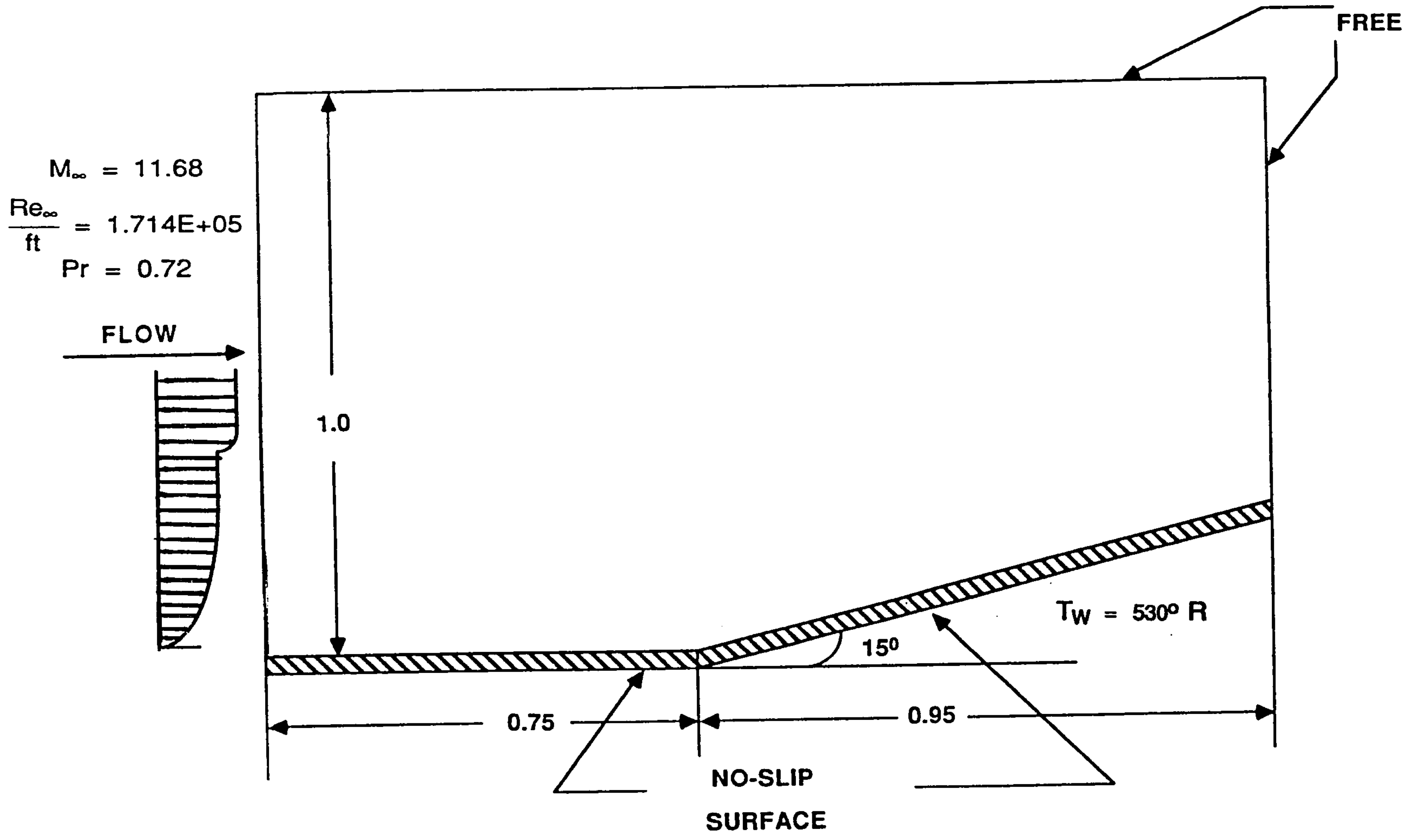


Fig. 32. Problem statement for the hypersonic flow over a compression corner.

5142 elements ; 5283 nodes

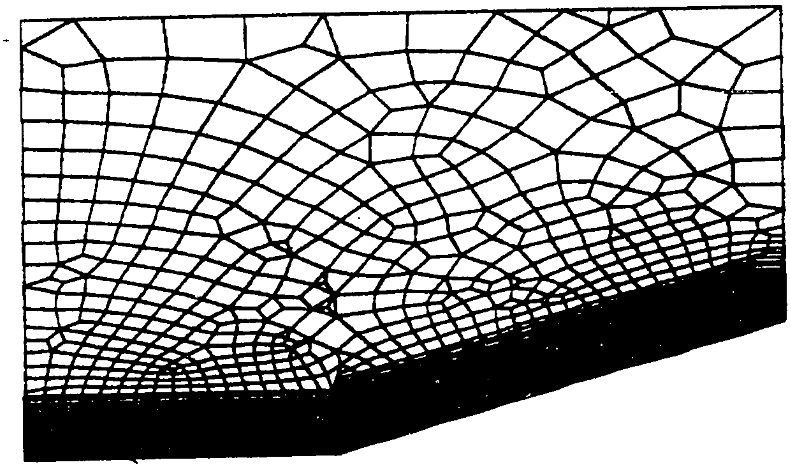
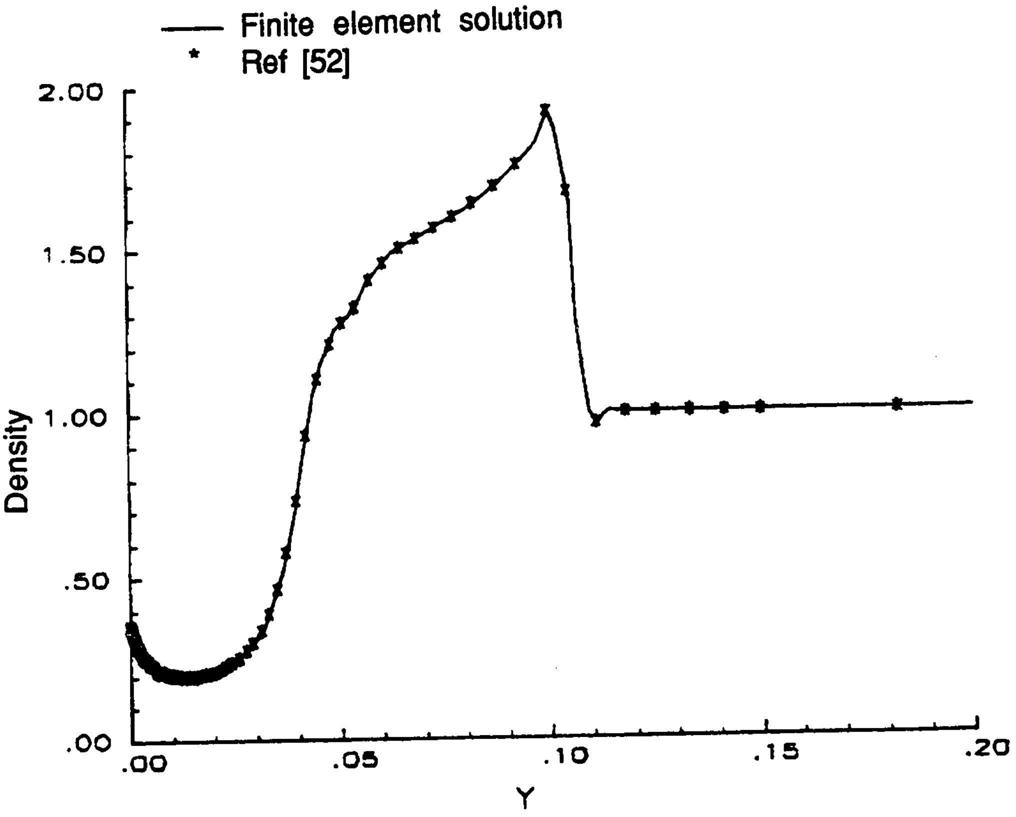
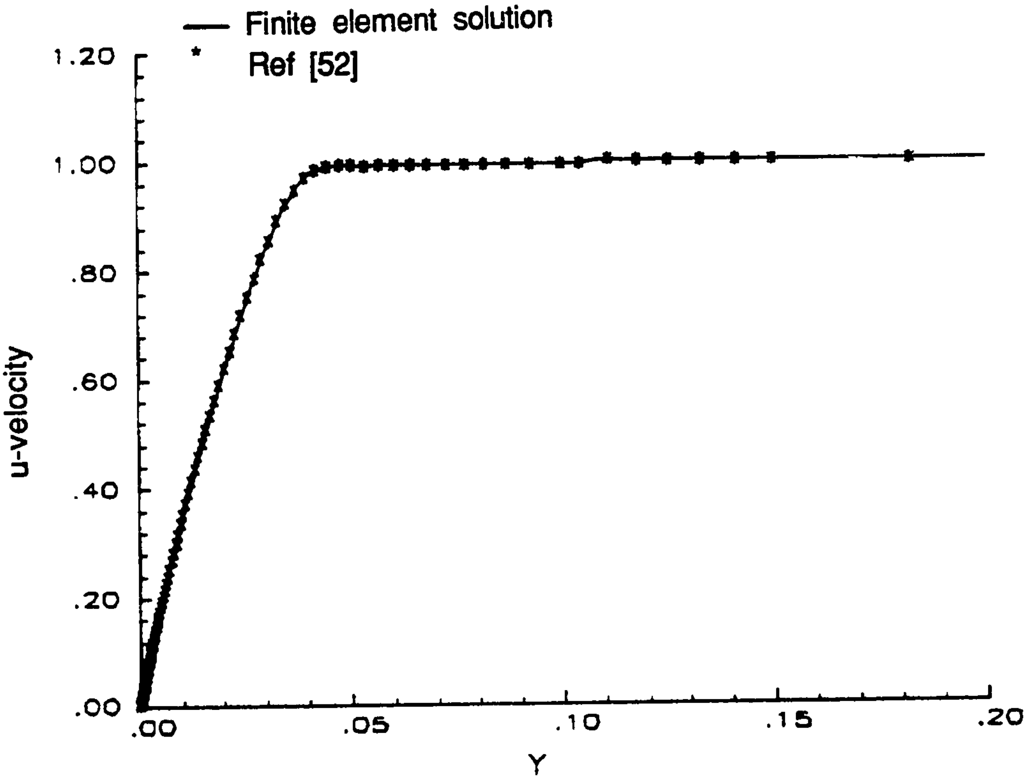


Fig. 33. Initial mesh for the hypersonic flow over a compression corner.



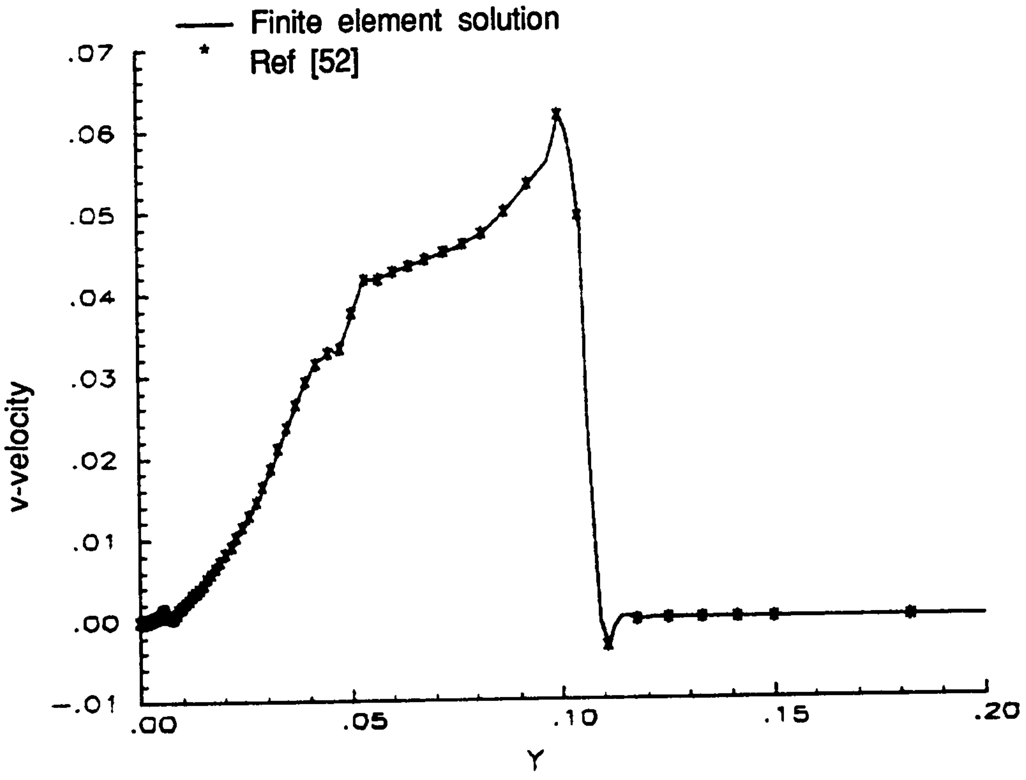
(a) Density

Fig. 34 Interpolation of flow variables at the inflow section of the compression corner.



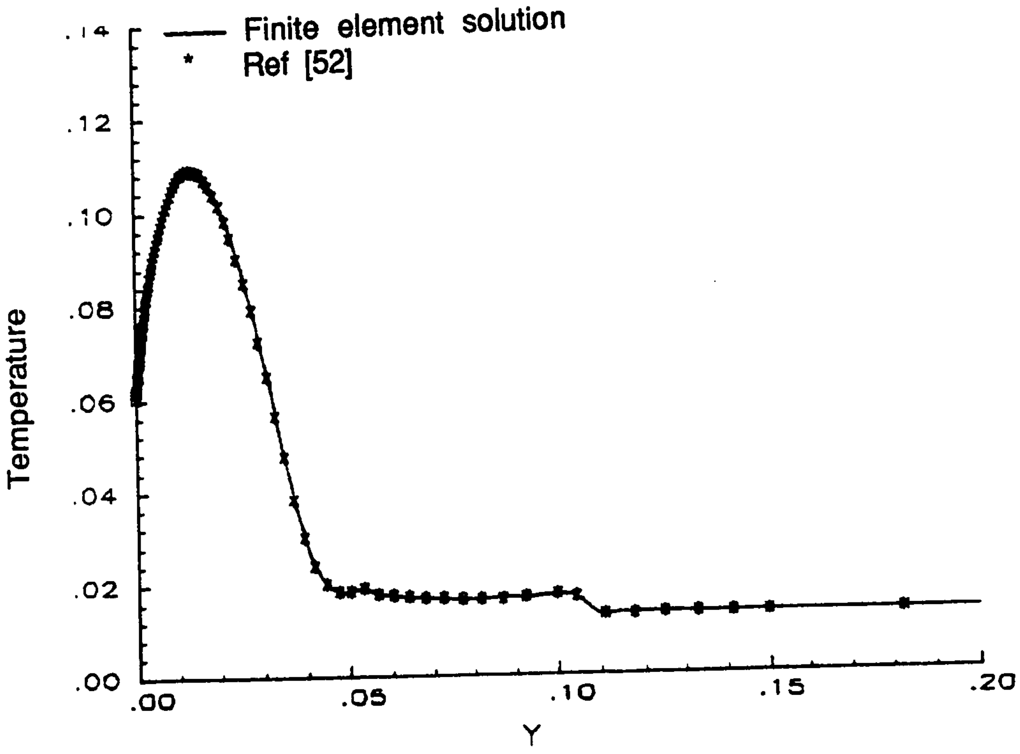
(b) u-velocity

Fig. 34 Interpolation of flow variables at the inflow section of the compression corner (continued).



(c) v-velocity

Fig. 34 Interpolation of flow variables at the inflow section of the compression corner (continued).



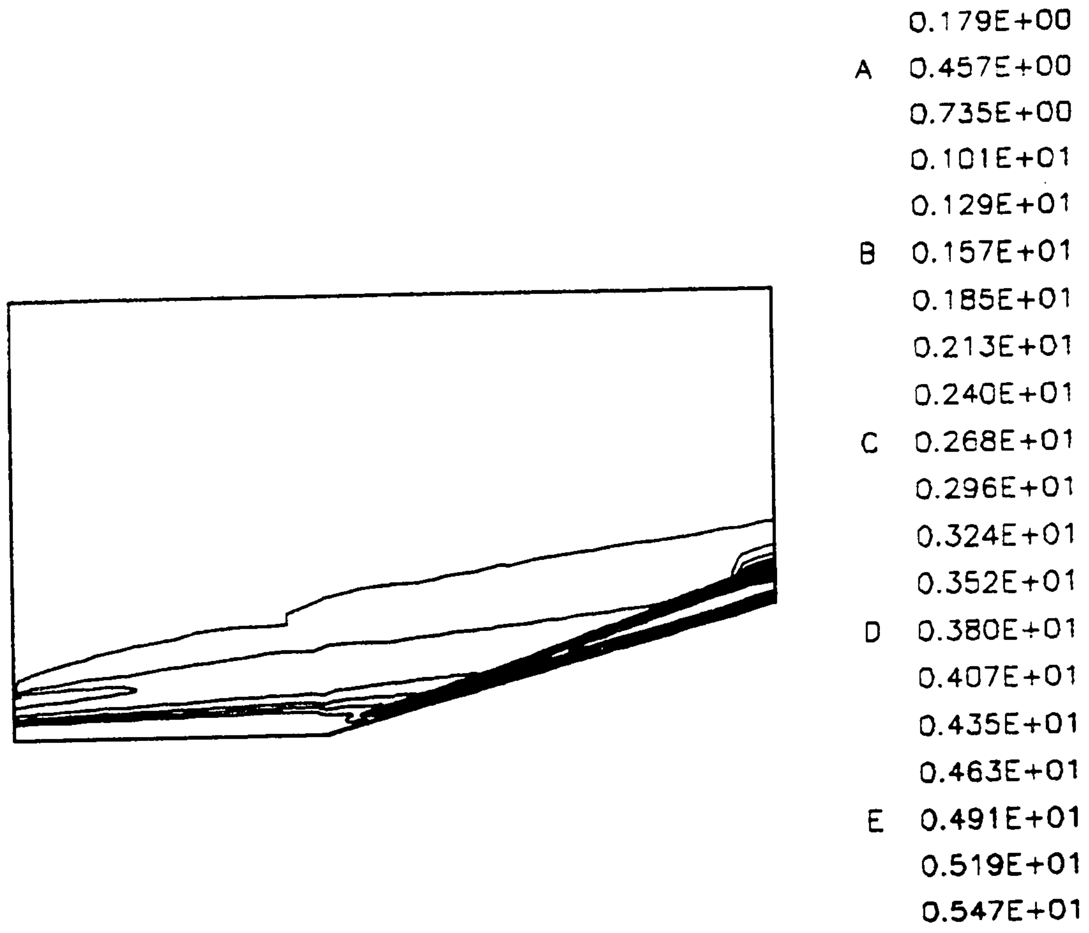
(d) Temperature

Fig. 34 Interpolation of flow variables at the inflow section of the compression corner (concluded).

orders of magnitude. The solution features are shown in Fig. 35. Some oscillations near the exit section are probably due to the larger elements in the shock region in this section

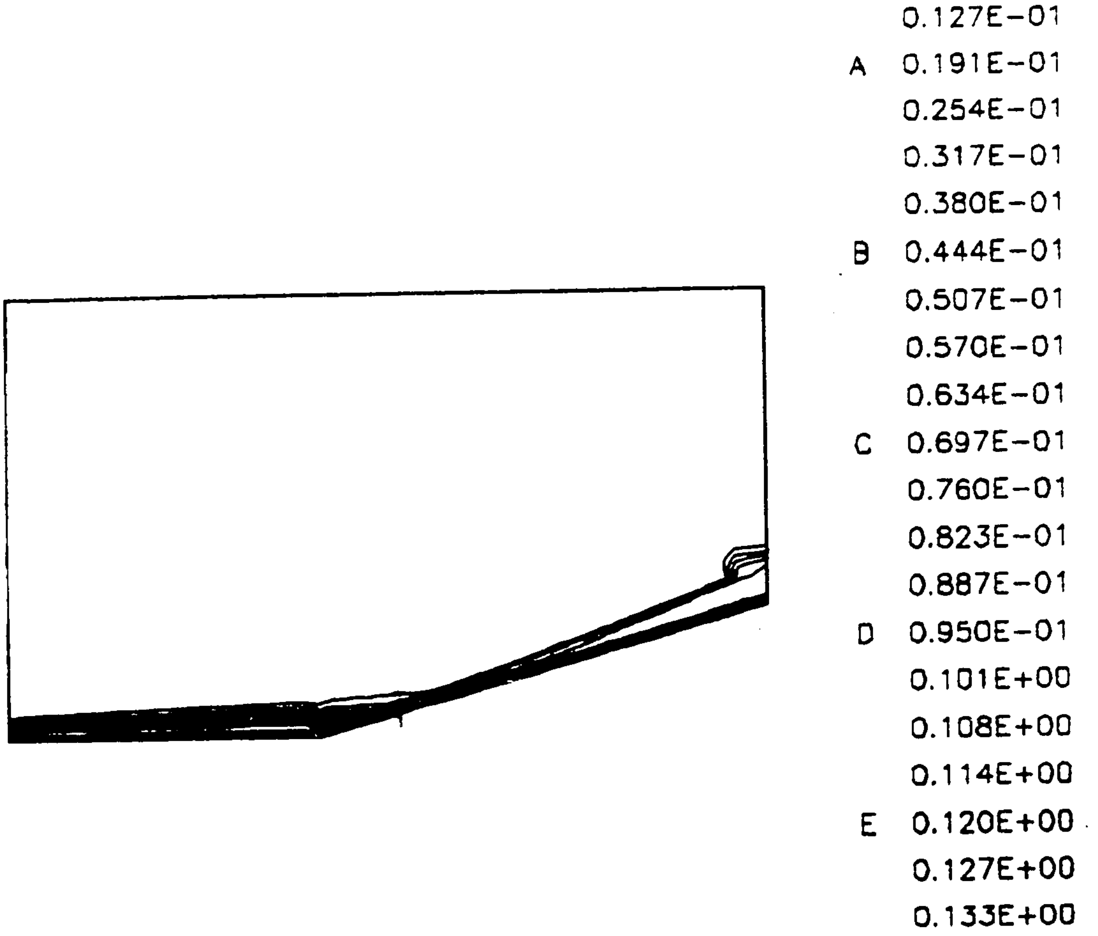
6.3.2 Details of adaptation

The two variables that were predominant in the boundary layer region were the absolute velocity and heat flux distribution of the finite element solution on the initial mesh. These contour plots are shown in Figs 36a and 36b. Both of these variables did not show the shocks because the absolute velocity change across the shock was not as significant as it was in the boundary layer and heat flux was predominant in the viscous boundary layer region compared to the shock region. Though the heat flux distribution identified the thermal boundary layer alone, it was sufficient to capture the high gradients at the wall. On the initial mesh, the concept of "multiple indicators" was employed again, but in this example in a slightly different sense. The mesh generation parameters are computed based on the second derivatives of absolute velocity in the boundary layer identified by the heat flux distribution and in the inviscid region the mesh generation parameters are computed based on the second derivatives of density. During the mesh generation process, the structured mesh is created in the region where the heat flux is predominant and unstructured mesh is generated in the inviscid region. This resulted in the first adaptive remesh as shown in Fig. 37. For the sake of clarity, only the nodal distribution is shown in this mesh where the boundary layer region and the shock region are identifiable. This initial mesh consisted of 3823 elements and 3470 nodes. Initial conditions for this mesh are interpolated linearly from the initial mesh.



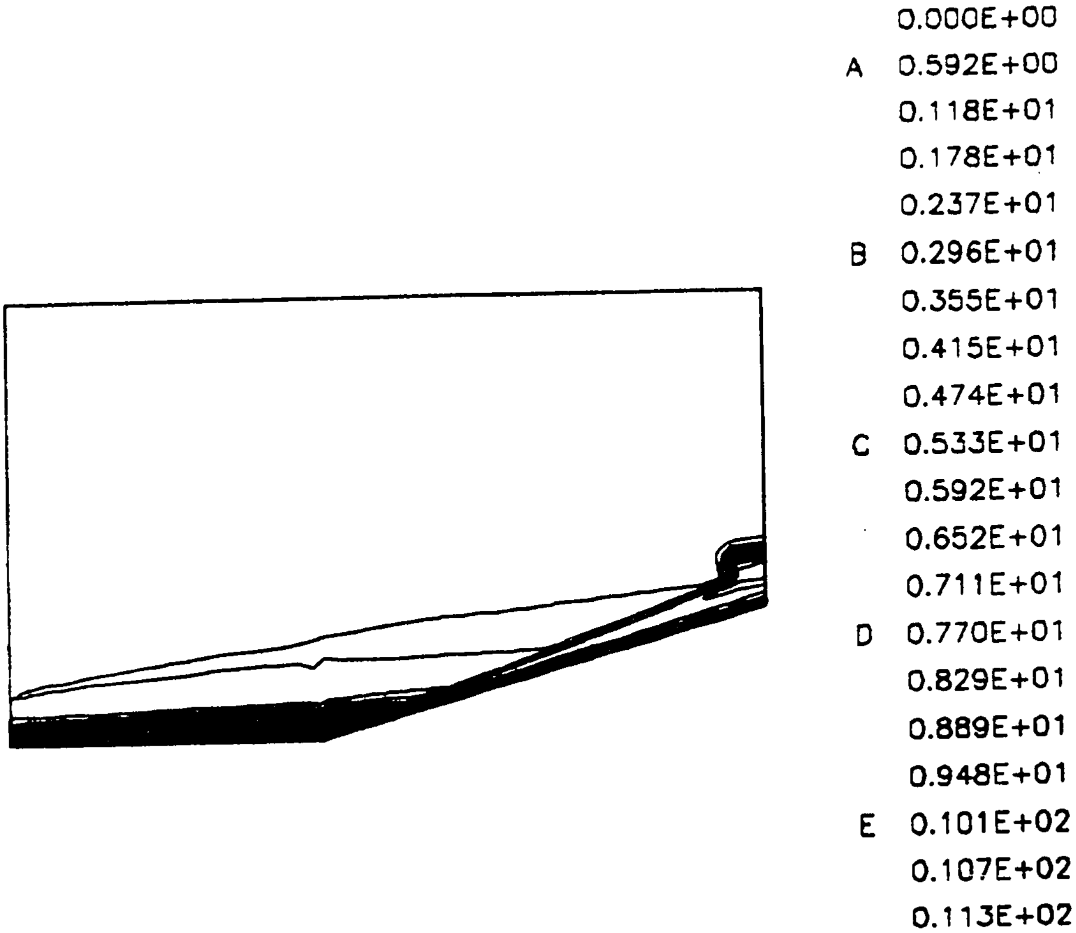
(a) Density contours

Fig. 35 Finite element solution features on the initial mesh for the compression corner.



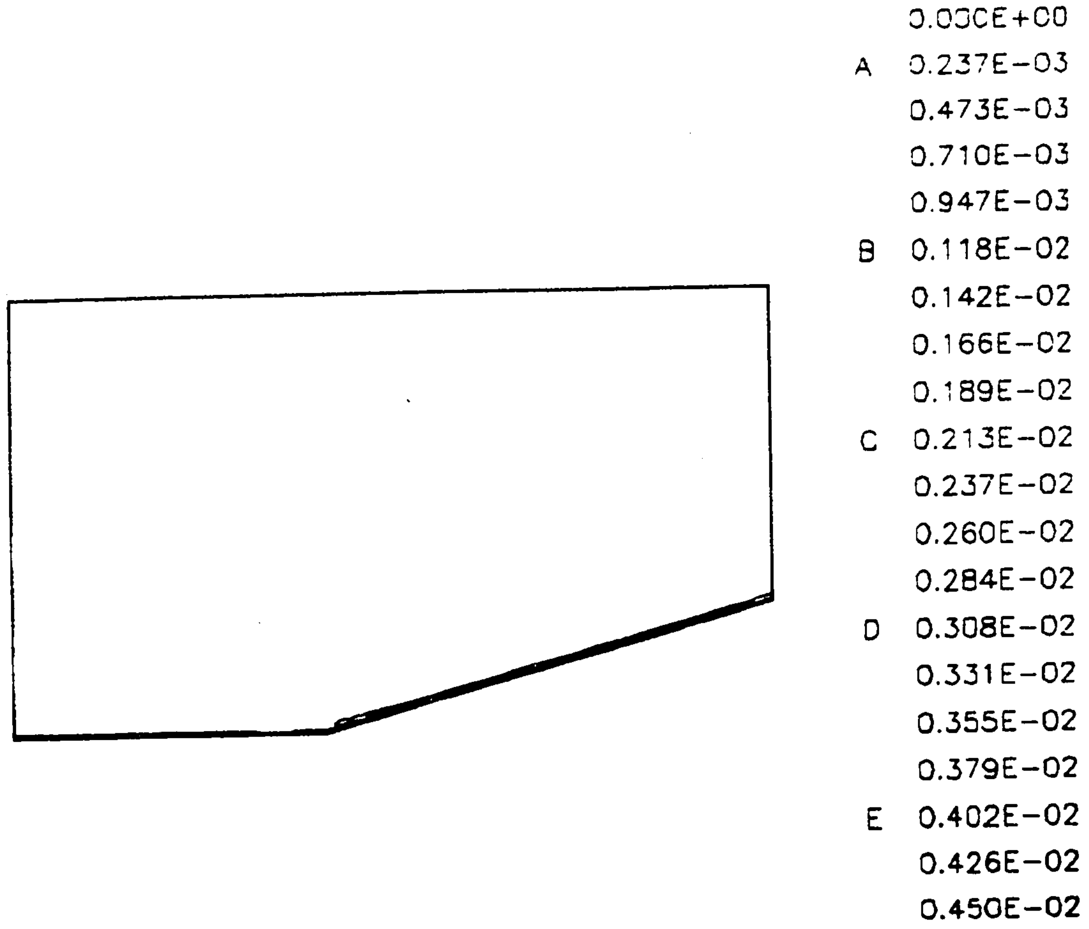
(b) Temperature contours

Fig. 35 Finite element solution features on the initial mesh for the compression corner (continued).



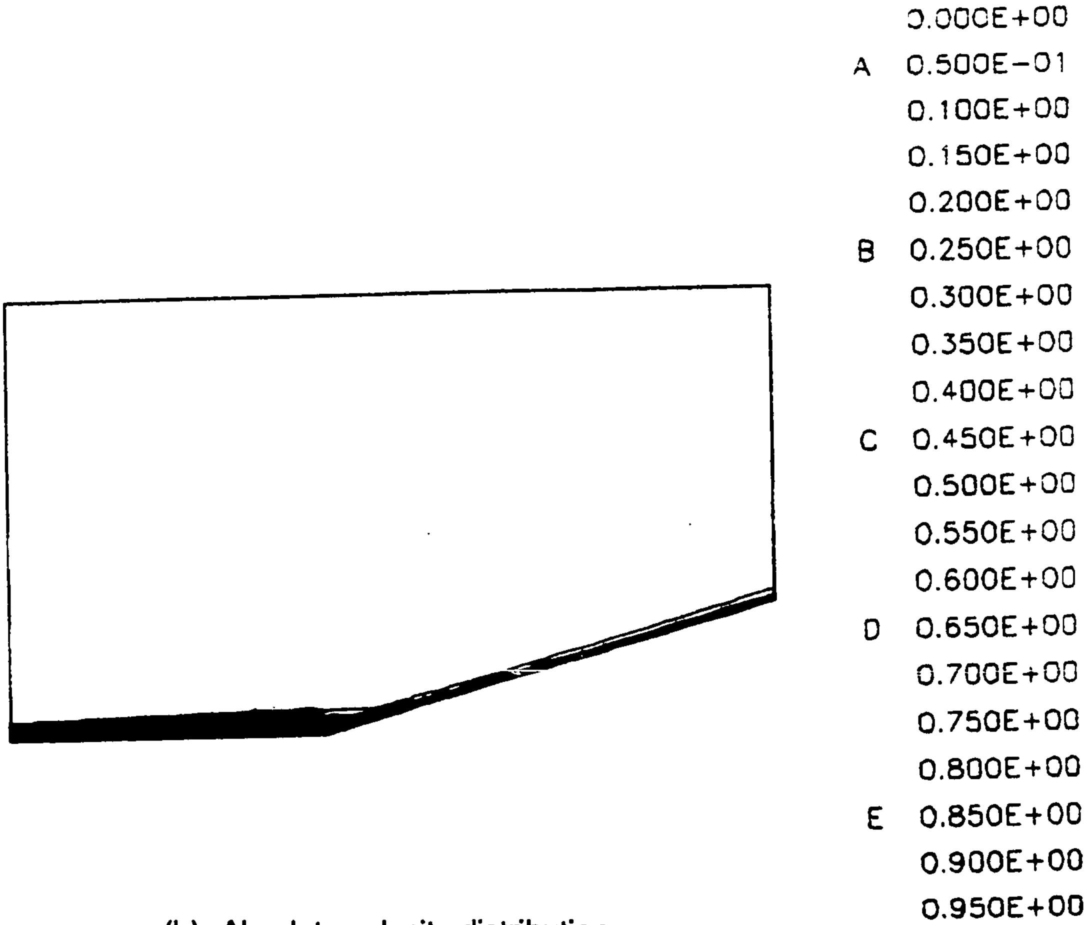
(c) Mach number contours

Fig. 35 Finite element solution features on the initial mesh for the compression corner (concluded).



(a) Heat flux distribution

Fig. 36 Boundary layer indicators based on the finite element solution on the initial mesh for the compression corner.



(b) Absolute velocity distribution

Fig. 36 Boundary layer indicators based on the finite element solution on the initial mesh for the compression corner (concluded).

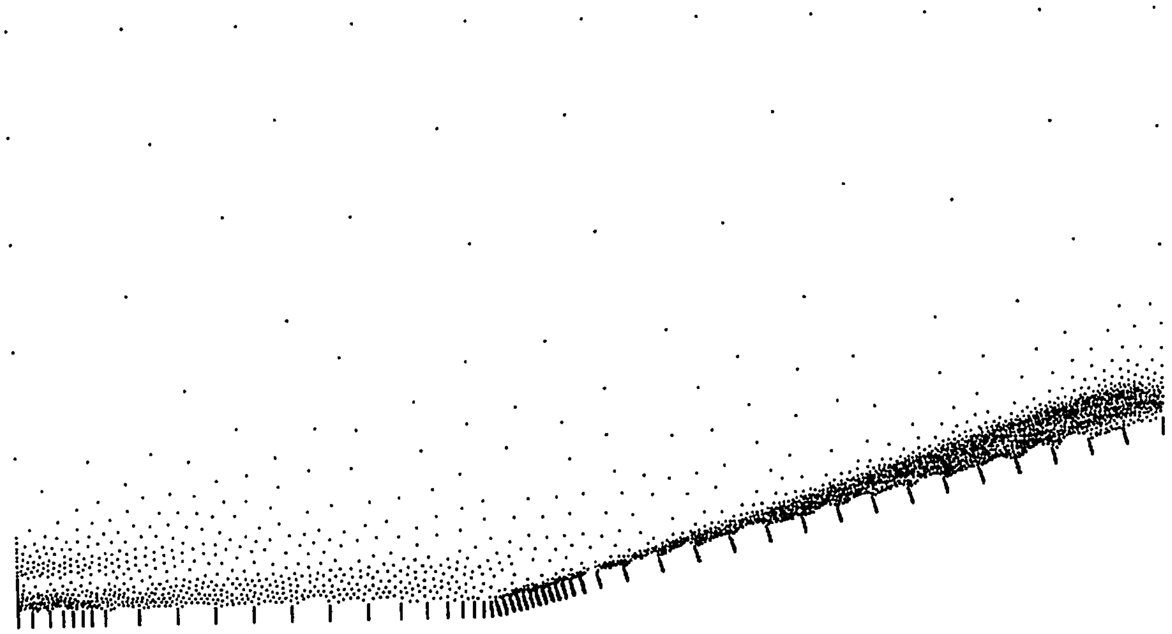


Fig. 37. Node distribution on the first remesh.

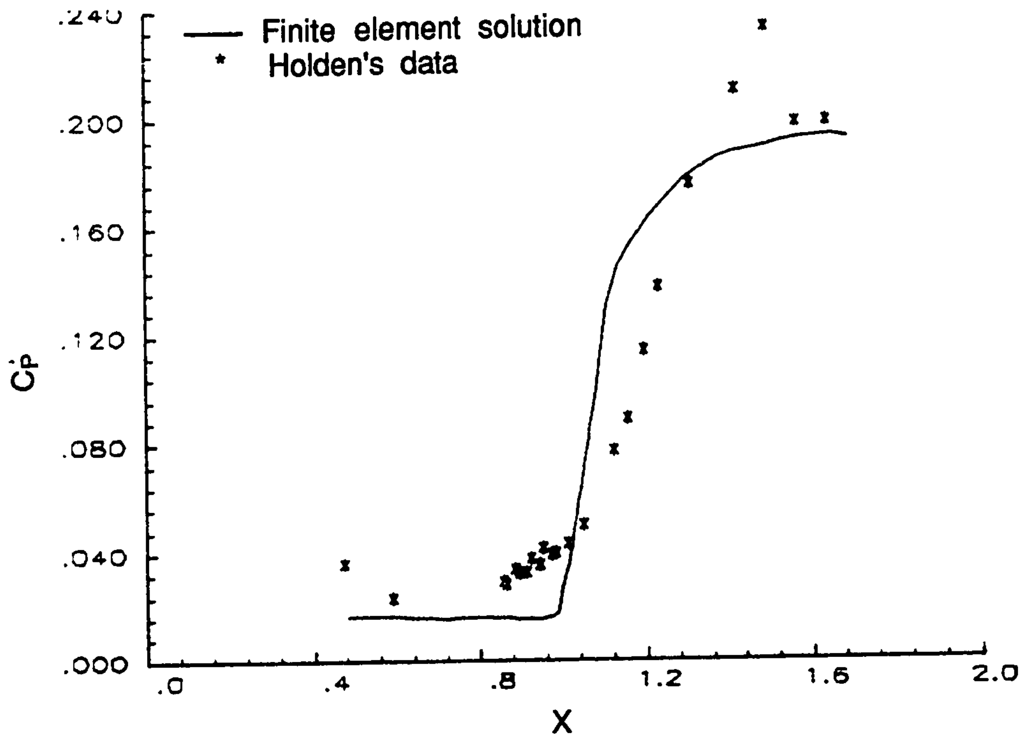
A finite element solution was obtained after the L_2-norm on the residuals of the conservation variables reduced three orders of magnitude. The key solution features of the wall pressure distribution and wall heat flux distribution are shown in Figs 38a and 38b. In these figures, the coefficient of pressure is given by

$$C_P = \frac{p_w}{\left(\frac{1}{2}\rho_\infty u_\infty^2\right)} \tag{6.1}$$

where subscripts w,∞ correspond to wall and free stream conditions respectively, and C_H is the heat-transfer coefficient given by

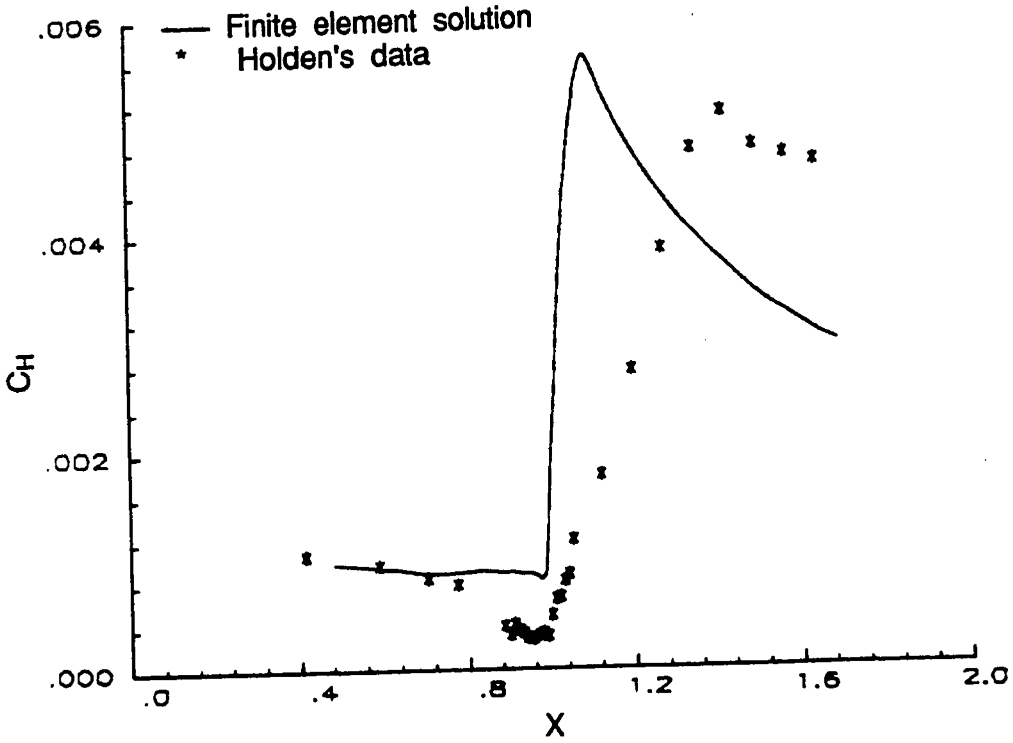
$$C_H = \frac{q_w}{\rho_\infty u_\infty \gamma c_v [T_{res} - T_w]} \tag{6.2}$$

where subscript "res" corresponds to free-stream stagnation conditions. The finite element solution did not contain a separation region and hence predicted higher heating rates near the corner as well as on the ramp section. A second remesh was generated based on the finite element solution on the first remesh. The second mesh had 4807 elements and 4177 nodes. The mesh generation was in a similar manner as in the case of the first remesh, that is, the boundary layer was identified from the heat flux distribution. The inviscid region is discretized using the second derivatives of density and the boundary layer region was discretized using the second derivatives of absolute velocity. The node distribution in this mesh is shown in Fig. 39. A finite element solution is obtained on this mesh and the key solution features are shown in Fig. 40. Though this solution predicted flow separation, it predicted the separation point at a location closer to the corner compared to the experimental value. Though the wall pressure distribution agreed well with the experimental



(a) Wall pressure distribution

Fig. 38 Finite element solution features on the first remesh for the compression corner.



(b) Wall heat flux distribution

Fig. 38 Finite element solution features on the first remesh for the compression corner (concluded).

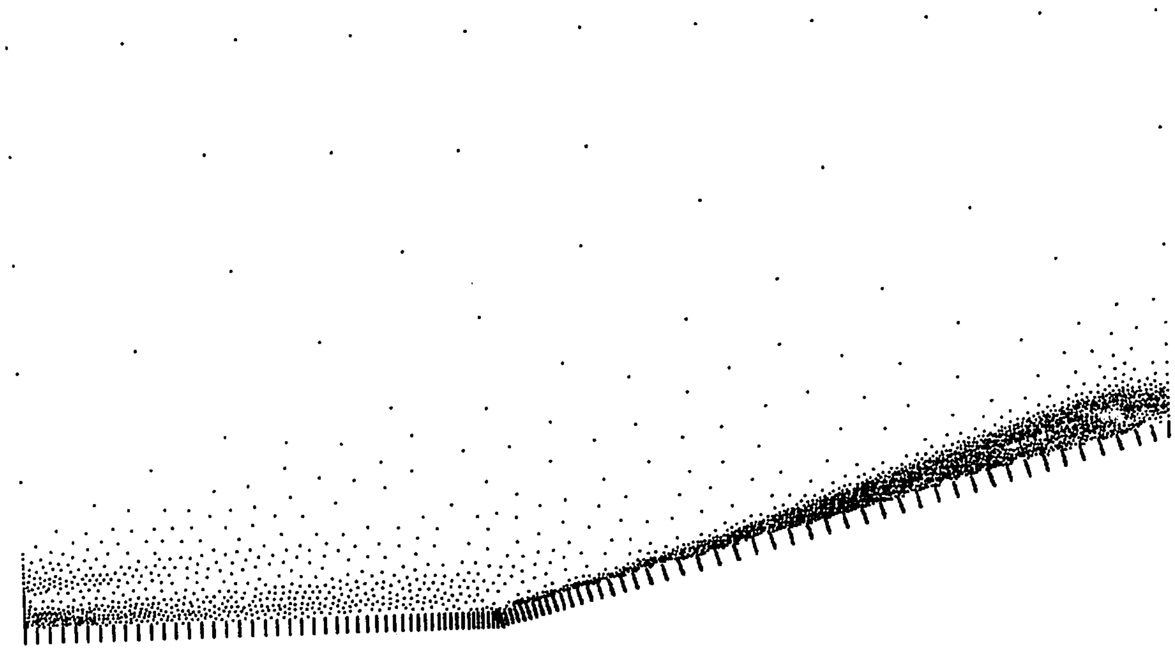
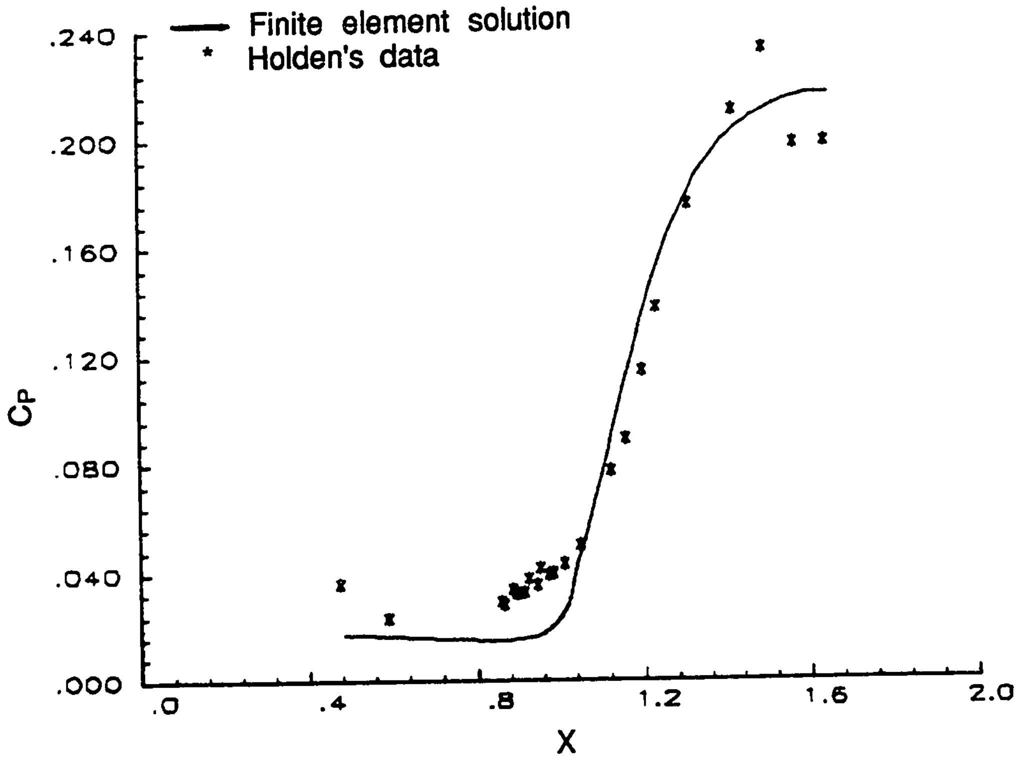
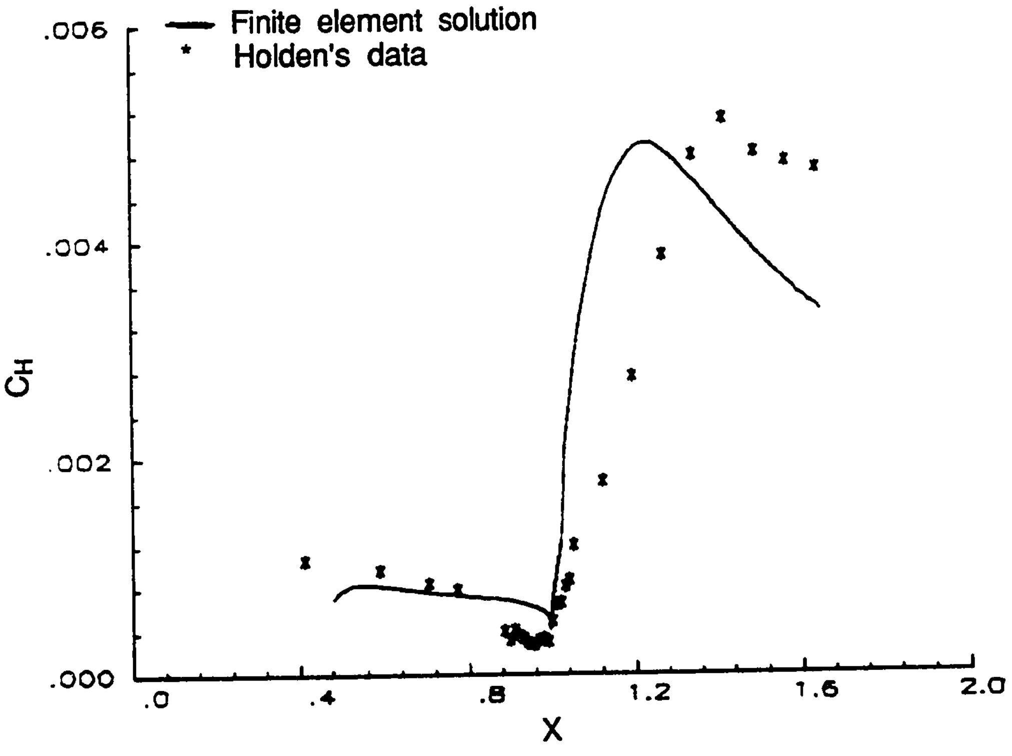


Fig. 39. Node distribution on the second remesh.



(a) Wall pressure distribution

Fig. 40 Finite element solution features on the second remesh for the compression corner.



(b) Wall heat flux distribution

Fig. 40 Finite element solution features on the second remesh for the compression corner (concluded).

data on the ramp section, it failed to predict the plateau in the separation region. A similar trend was observed in the wall heat flux distribution in the region of separation and subsequently predicted a higher wall heating rate over the ramp section.

At this point it was observed that the tangential velocity is a better indicator in refining the separation region than the absolute velocity. The reason being that the gradients of tangential velocity are much stronger than that of absolute velocity in the separation region. Fig. 41 shows the value of h_1 spacing along the wall when tangential velocity is used as an indicator. This figure shows that the spacing along the wall dropped two orders of magnitude near the separation region, indicating a large number of wall nodes are required in this region.

A third remesh was created with tangential velocity as an indicator in the boundary layer and density as an indicator in the inviscid region. This mesh consisted of 14626 nodes and 14663 elements and is shown in Fig. 42. A finite element solution is obtained on this mesh after 10,000 iterations until the error norm on the residuals of the conservation variables reduced at least 4 orders of magnitude. Distribution plots of different flow variables are shown in Fig. 43. The wall pressure distribution and the heat flux distribution are shown in Fig. 44. Though the solution is predicting separation further down stream than the experimental value, the heat flux distribution is closer to the experimental data compared to that of earlier meshes. The finite element solution is predicting the peak heating rate within 5% of the peak experimental value. The number of elements in the boundary layer are about 9000 out of which nearly 6000 elements are placed in the region where the wall spacing value was two orders

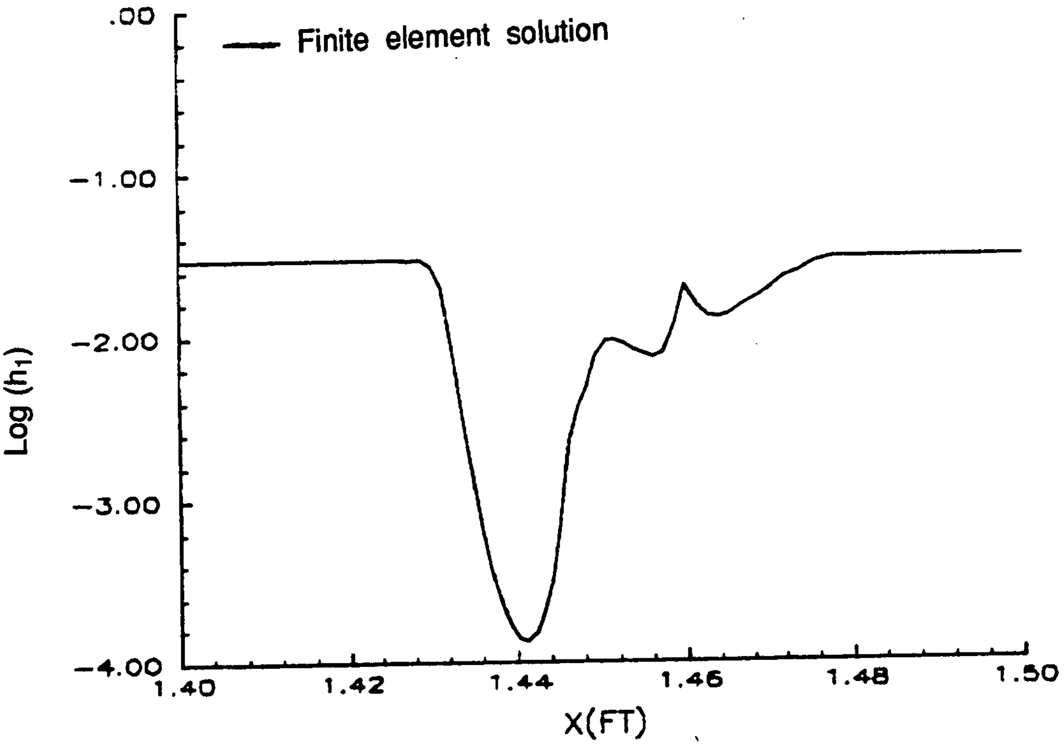


Fig. 41. Wall spacing distribution computed based on tangential velocity as an error indicator.

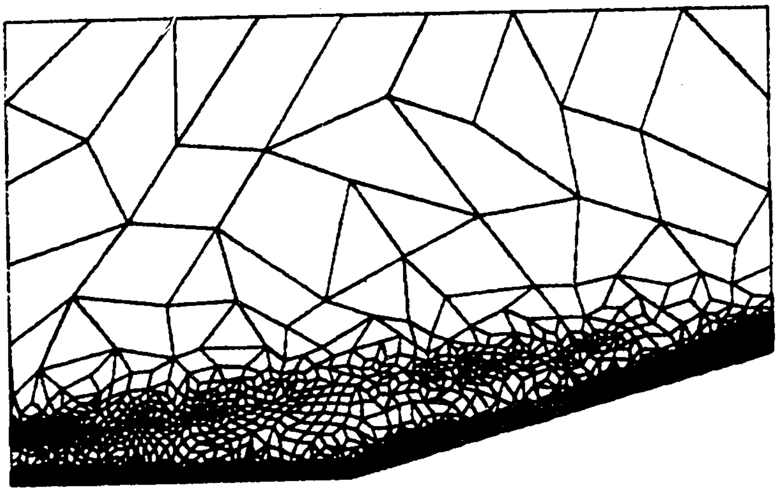
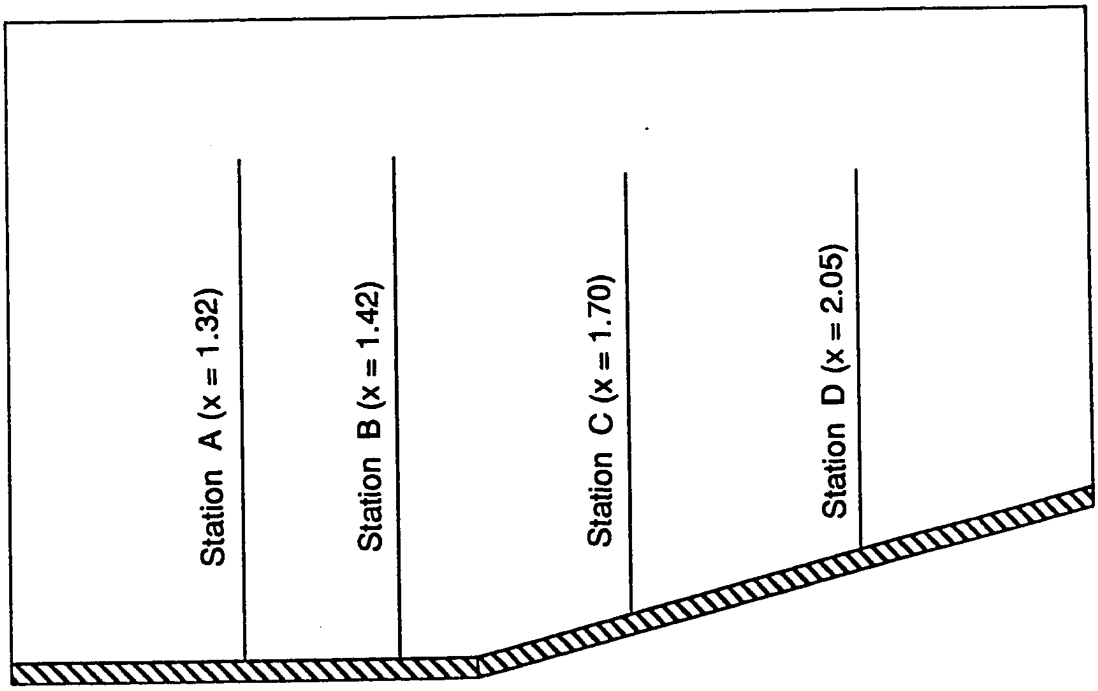


Fig. 42. Third remesh.



(a) Station locations

Fig. 43 Boundary layer profiles from the finite element solution on the third remesh for the compression corner

(b) u-velocity

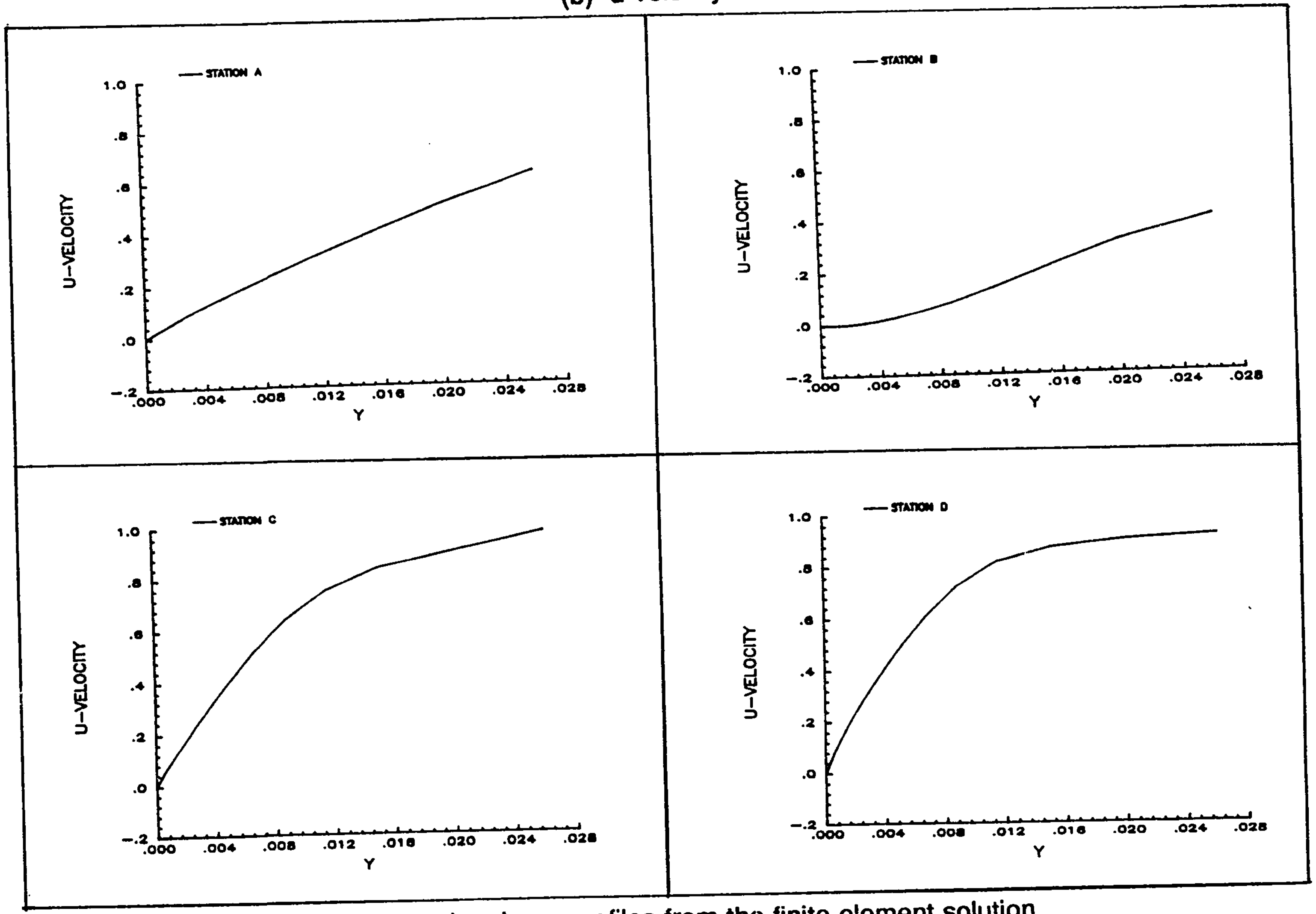


Fig. 43 Boundary layer profiles from the finite element solution on the third remesh for the compression corner(continued)

(c) Pressure

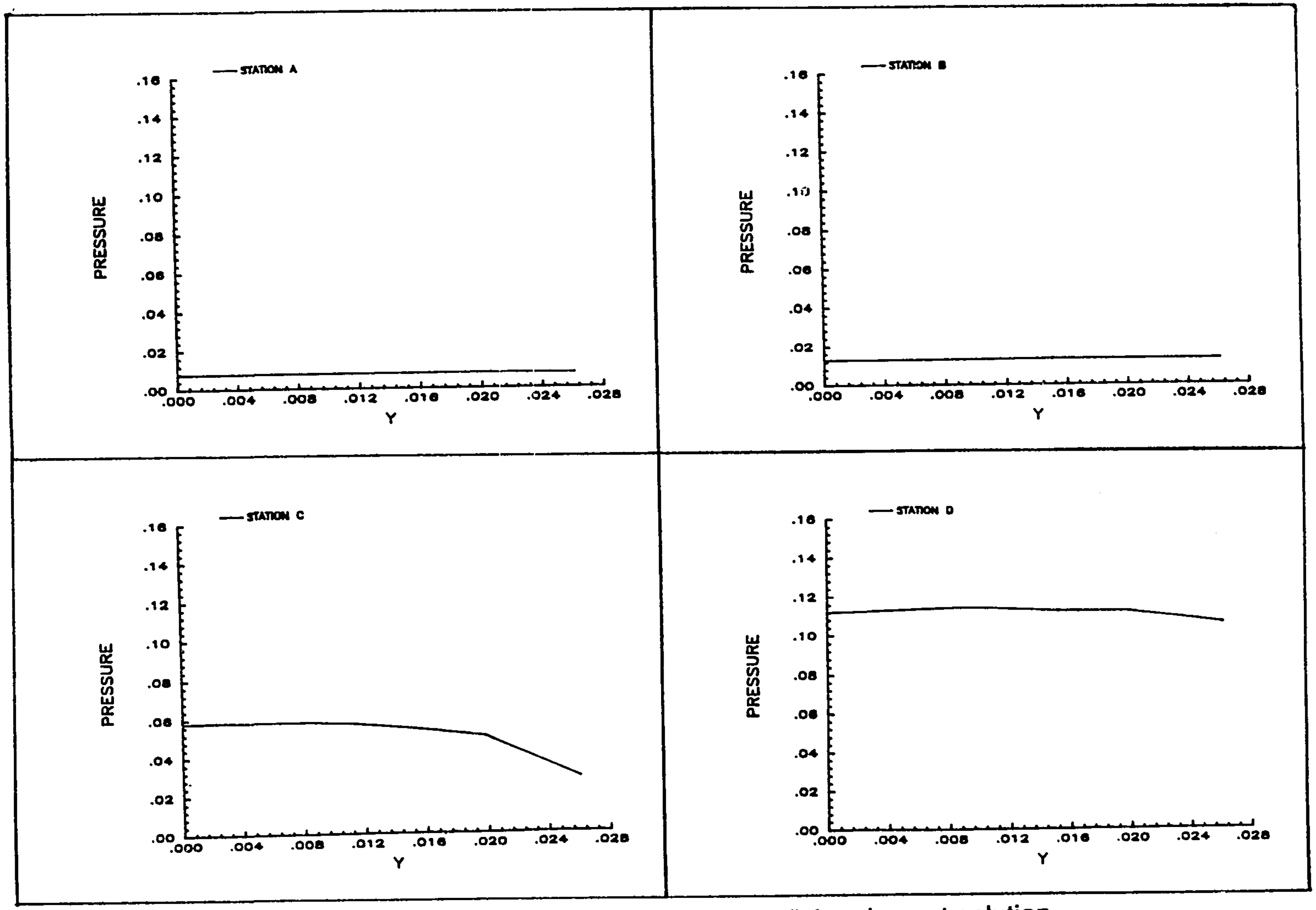


Fig. 43 Boundary layer profiles from the finite element solution on the third remesh for the compression corner(continued)

(d) Temperature

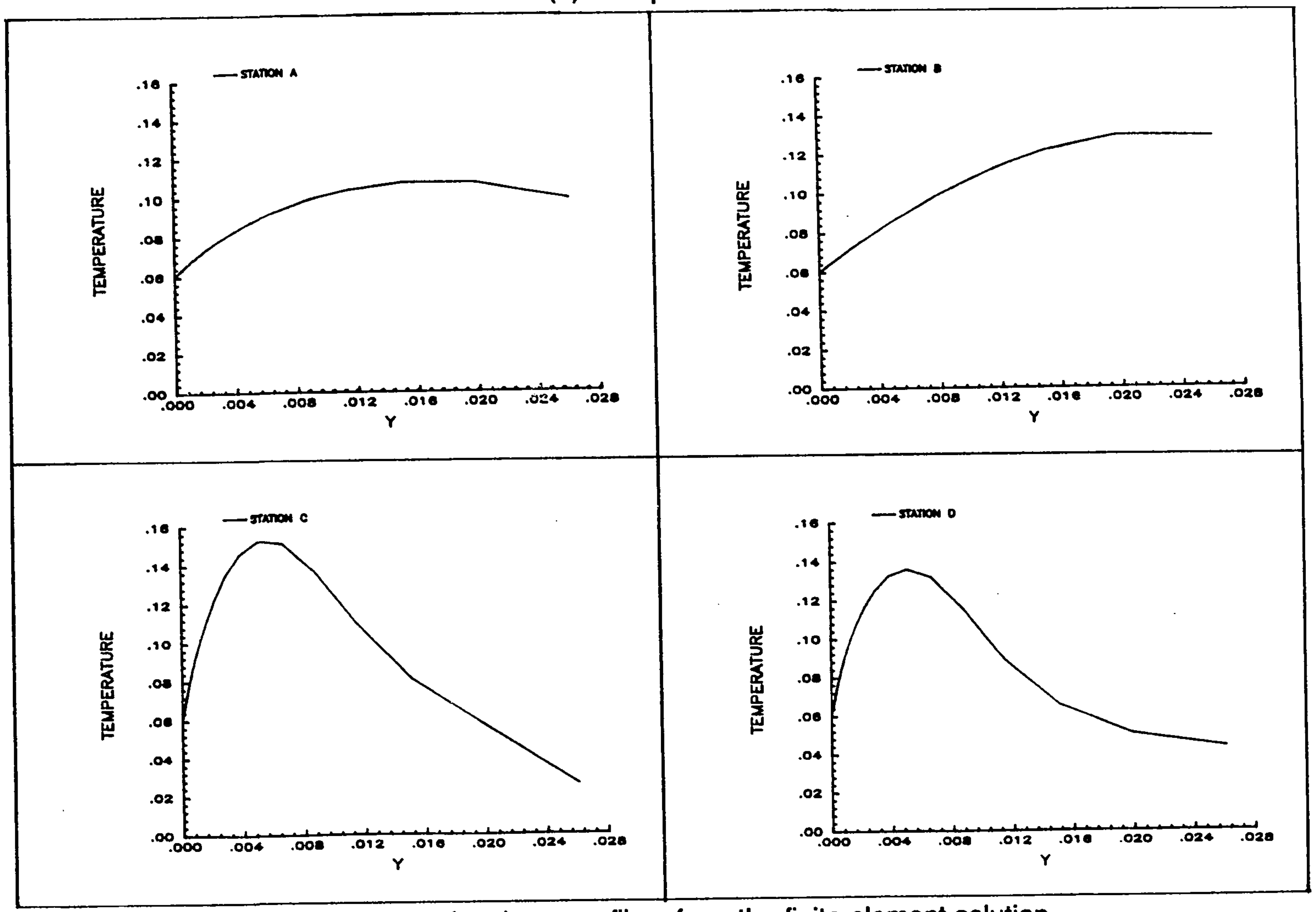
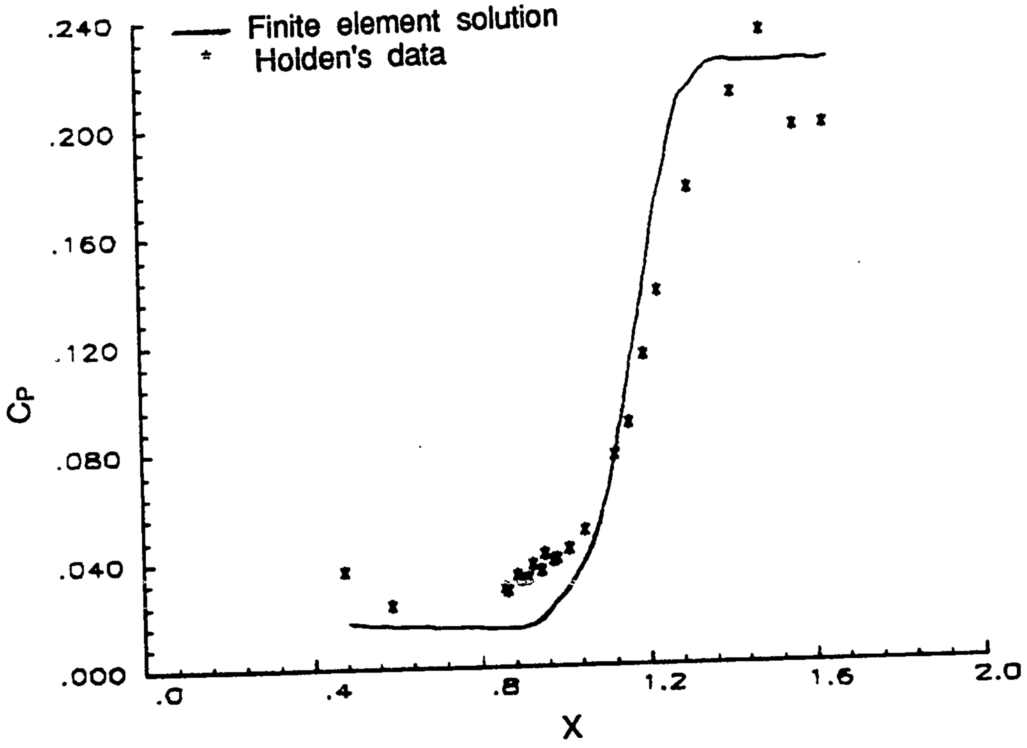
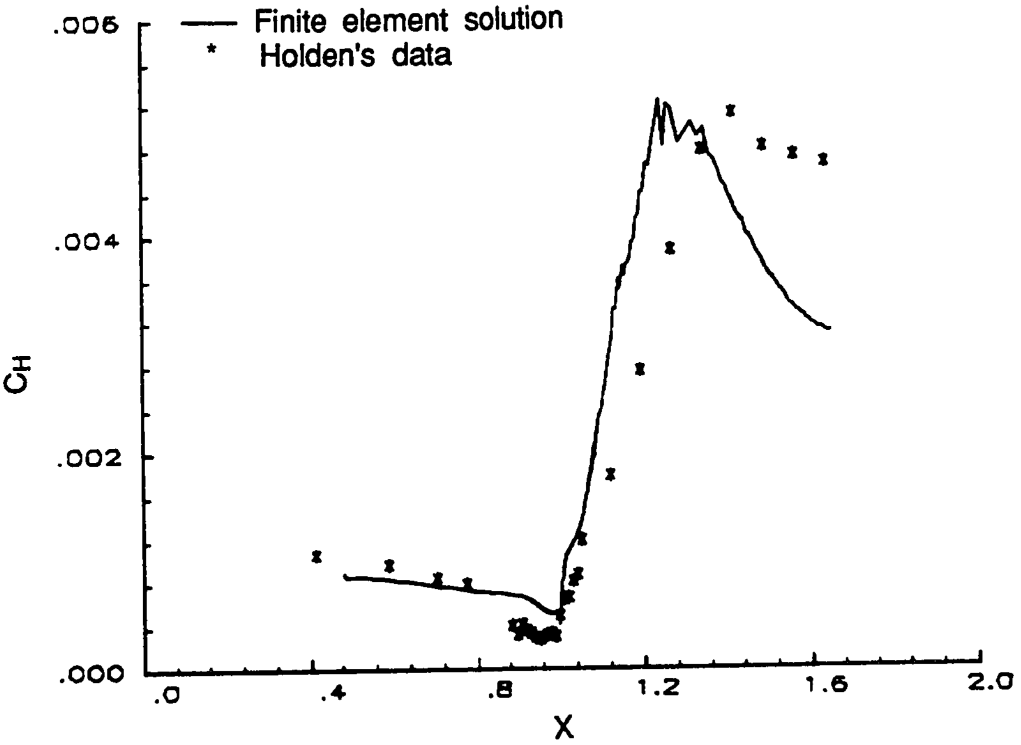


Fig. 43 Boundary layer profiles from the finite element solution on the third remesh for the compression corner(concluded)



(a) Wall pressure distribution

Fig. 44 Finite element solution features on the third remesh for the compression corner.



(b) Wall heat flux distribution

Fig. 44 Finite element solution features on the third remesh for the compression corner (concluded).

of magnitude less than that on the flat plate portion of the domain. There is a significant improvement in the overall prediction of the heat flux distribution through successive remeshing. Further refinement may improve the heat flux prediction in the separation region which would subsequently lower the heat flux on the ramp portion.

Chapter 7

CONCLUDING REMARKS

7.1 Recapitulation

This dissertation presents a new adaptive remeshing method for high speed compressible flows. The method uses quadrilateral elements where possible, and triangles are introduced as needed. The primary goal of this study was to develop a new remeshing method which uses both the concepts of unstructured and structured meshes for the finite element analysis to predict accurate aerodynamic heating in problems related to high speed viscous flows. Though the method was developed for high speed compressible flows, it can be used for any field problem with steep gradients.

The remeshing method uses a solution based on an old mesh to create a new mesh. Second derivatives of the previous solution are used to compute parameters that determine the size and orientation of elements on the new mesh. The new mesh is created one element at a time using an advancing front technique. The front begins from the discretized boundary of the solution domain and advances into the domain as the mesh evolves. As the mesh construction continues, the front continuously changes its shape and vanishes when the mesh is complete. In the present implementation for viscous flows, a structured mesh is created in the boundary layer based on the solution on a previous mesh and the front covering the "no-slip" surface will move to the edge of the boundary layer and the remaining domain is remeshed in an unstructured manner.

The remeshing method is first applied to three elliptic problems for steady state heat conduction, to study the convergence rates compared with uniform h refinement. Next the method is applied to three high speed compressible flow problems in association with an upwind, cell-centered finite element scheme to demonstrate its capability of adaptation to critical flow features such as shocks and thin boundary layers.

7.2 Conclusions on Applications

The remeshing method is applied to three pure conduction problems with exact analytical solutions. To assess the effectiveness of the approach, a global error measure was used to study solution convergence rates as meshes were adaptively refined. Comparisons were also made with convergence rates for solutions obtained on successively refined uniform, structured meshes. For the problem with a smooth solution with no steep gradients, the method offered no advantage. However for the problems with steep gradients, the method gave much faster convergence compared to uniform refinement. Moreover in these problems with steep gradients the error in the solution reduced even though the number of unknowns decreased during remeshing.

The remeshing method is applied on three high speed compressible flow problems which include complexities characteristic of these flows. Results for an inviscid shock reflection showed a reduction in the error by adaptive remeshing compared to uniform refinement. This problem has an exact solution computed from oblique shock relations. The error norm in this example is not as reliable as in the thermal problems since a certain amount of artificial diffusion is always associated with shock capturing and since the solution algorithm is first-order accurate the optimal error-reduction is O(h). For the hypersonic boundary layer problem the adaptive remeshing method identified the boundary layer edge using the shear stress distribution in the flow and created a mesh that had a structured portion within the boundary layer and an unstructured portion in the inviscid region. The final remesh for this problem consisted of fewer number of elements and unknowns compared to the initial mesh. The results for this problem showed very good agreement on the wall heat flux distribution with the boundary layer solution .

For the problem of hypersonic flow over a compression corner, the adaptive remeshing demonstrated the capability of handling problems with strong viscous-inviscid interactions. Density was used as an indicator for the inviscid region of the flow and the flow heat flux distribution is used to identify the boundary layer region. Tangential velocity was used as an indicator for the structured remeshing within the boundary layer because the tangential velocity refined the region near the separation point better than other indicators. The wall spacing distribution with this indicator showed that a very close spacing is required in the separation region. So structured remeshing created a large number of elements in this region. On the whole, adaptive remeshing improved the aerodynamic heating prediction through successive remeshes. Though the first order accurate finite element solution did not predict the separation point very accurately, the overall trend and the peak heating rates are in agreement with the experimental results. This is mainly due to the solution algorithm being first order accurate. The wall pressure distribution agreed well with the experimental results except in the plateau region near the flow separation.

The ability of an adaptive refinement scheme to capture complex details depends strongly on the capability of the numerical scheme to solve the flow problem effectively. Since a new remesh depends on the solution from a previous mesh, the quality of the solution determines the resolution capability of the new mesh. The flow applications presented in this dissertation made use of a point-implicit, upwind, piecewise constant solution algorithm. The first order accuracy tended to give low quality solutions on some meshes, and special attention was required to develop error indicators that could effectively guide the development of a remesh.

An advantage of the remeshing method is that it is implemented in a code independent of the analysis program. Thus when higher quality, second order accurate finite element algorithms become available, the remeshing method developed in this dissertation could easily be applied. The author believes (and expects) that the remeshing method would be even more effective in association with a second order accurate flow analysis algorithm.

7.3 Recommendations for future research

The present study has shown some benefits of adaptive remeshing but further enhancement of the method is required to make it more general. The following recommendations are made for future research.

(1) Handling of curved boundary segments needs to be incorporated to make the method applicable to any two-dimensional geometry with curved boundaries.

(2) Extension of the method to three-dimensions will involve using a combination of hexahedral, tetrahedral, pyramid, and prismatic elements. Investigation of the advancing frontal technique in generating these elements is to be made. However, structured remeshing using hexahedra elements alone for three-dimensional boundary layers in a straight forward extension of the two-dimensional concepts.

(3) The remeshing method needs to be tested in association with second-order accurate flow analysis algorithms to see an improvement in the overall performance. The method can be used in association with a "h-p" hybrid algorithm too, which is still under research.

(4) Though the remeshing algorithm does not have strong vectorization properties, a local remeshing can be deviced for transient problems to reduce the total CPU time required. A local front surrounding a region of steep gradients may be preserved and remeshing can be performed in this local region for this class of problems.

(5) Different error indicators need to be tested to define the edge of the boundary layers and predict aerodynamic heating accurately for flow problems with strong viscous-inviscid interactions and for turbulent flow problems.

(6) Wall spacing distribution for the hypersonic flow over a compression corner suggests an unstructured mesh may be desirable in the flow separation region where the number of unknowns can be reduced compared to a structured mesh.

REFERENCES

1. Dejarnette, F.R., Hamilton, H.H., Weilmuenster, K.J. and Cheatwood, F.M.:"A Review of Some Approximate Methods in Aerodynamic Heating Analysis", Journal of Thermophysics and Heat Transfer, Vol 1, No.1,1987, pp.5-12.

2. Kennon, S.R.: "An Optimal Spatial Accuracy Finite Element Scheme for the Euler Equations," AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, 1988. AIAA-88-0035.

3. Oden, J.T. and Reddy, J.N.: "An Introduction to the Mathematical Theory of Finite Elements", John Wiley and Sons, New York, 1976.

4. Strang, G. and Fix, G.J.: "An Analysis of the Finite Element Method." Prentice Hall, New York, 1973.

5. Dechaumphai, P., Wieting, A.R., and Pandey, A.K.: "Finite Element Flow-Thermal-Structural Analysis of Aerdynamically Heated Leading Edges", Proceedings of the 2nd NASA/Air Force Symposium on Recent Experiences in Multi-Disciplinary Analysis and Optimization, Part 2 , pp.971-990, Hampton, Virginia, Sept., 1988.

6. Wieting, A.R.: " The Critical Role of Aerodynamic Heating Effects on the Design of Hypersonic Vehicles", Proceedings of the Seventh International Conference on Finite Element Methods in Fluids, Huntsville, Alabama, 1989, pp.143-150.

7. Thornton, E.A., Dechaumphai, P. and Vemaganti, G.R.: "A Finite Element Approach for Prediction of Aerothermal Loads", AIAA/ASME 4th Fluid Mechanics, Plasma Dynamics and Lasers Conference, Atlanta, Georgia, 1986. AIAA-86-1050.

8. Zienkiewicz, O.C., "The Finite Element Method", 3rd edition, McGraw-Hill Book Company, New York, 1971.

9. Cavendish, J.C.: "Automatic Triangulation of Arbitrary Planar Domains for the Finite Element Method", International Journal for Numerical Methods in Engineering, Vol 8, 1974, pp.679-697.

10. Cavendish, J.C., Field, D.A. and Frey, W.H.: "An Approach to Automatic Mesh Generation for Structural Analysis", International Journal for Numerical Methods in Engineering, Vol 21, 1985, pp.329-347.

11. Watson, D.F.: "Computing the n-dimensional Delaunay Tessalation with Application to Voronoi Polytopes", Computers Journal, Vol 21, No. 3, 1978, pp.243-245.

12. Weatherill, N.P: " The Generation of Unstructured Grids Using Dirichlet Tessalation," Princeton University Report MAE 1715, July, 1985.

13. Lee, Y.T., Pennington, A.D. , and Shaw N.K.: "Automatic Finite Element Mesh Generation from Geometric Modes- A Point Based Approach", ACM Transactions in Graphics, No. 3, 1984, pp.287-311.

14. Lo, S.H.: "A New Mesh Generation Scheme for Arbitrary Planar Domains", International Journal for Numerical Methods in Engineering, Vol 21, 1985, pp.219-249.

15. Joe, B. and Simpson, R.B.: "Triangular Meshes for Regions on Complicated Shapes", International Journal for Numerical Methods in Engineering, Vol 23, pp.751-778, 1986.

16. Yerry, M.A. and Shephard, M.S.: "Automatic Three-Dimensional Mesh Generation for Three-Dimensional Solids", Computers and Structures, Vol 20, 1985, pp.31-39.

17. Kela, A., Perucchio, R. and Voelcker, H.B.: "Toward Automatic Finite Element Analysis", Computer Methods in Mechanical Engineering, July, 1986, pp.57-71.

18. Shephard, M.S., Grice, K.R., Soechtig, C.E. and Graichen, C.M.: "Automatic , Topologically Correct, Three-Dimensional Mesh Generation by the Finite Octree Technique", Report of the Center for Interactive Computer Graphics, RPI, Troy, NY, 1987.

19. Oden, J.T.: "Notes on Grid Optimization and Adaptive Methods", TICOM report, University of Texas at Austin, Texas, 1983.

20. Lohner, R., Morgan, K. and Zienkiewicz, O.C.: "Adaptive Grid Refinement for the Euler and Compressible Navier-Stokes Equations", Proceedings of the Conference on Accuracy Estimates and Adaptive Refinements in Finite Element Computations, Lisbon, Portugal, 1984.

21. Oden, J.T., Demkowicz, L., Strouboulis, T. and Devloo, Ph.: "Adaptive Methods for Problems in Solid and Fluid Mechanics", Accuracy Estimates and Adaptive Refinements in Finite Element Computations, Ed. by Babuska et al , John Wiley & Sons, 1986.

22. Oden, J.T., Devloo, P. and Strouboulis, T.: "Adaptive Finite Element Methods for the Analysis of Inviscid Compressible Flows", TICOM report, University of Texas at Austin, Texas, 1986.

23. Ramakrishnan, R., Bey, K.S. and Thornton, E.A. "An Adaptive Quadrilateral and Triangular Finite Element Scheme for Compressible Flows", AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, 1988. AIAA-88-0033.

24. Shapiro, R.A. and Murman, E.M.: "Cartesian Grid Finite Element Solutions to the Euler Equations", AIAA 25 th Aerospace Sciences Meeting , Reno, Nevada, 1987. AIAA-87-0559.

25. Patera, A.: "A Spectral Element Method for Fluid Dynamics: Laminar Flow in a Channel Exapansion", Journal of Computational Physics, Vol 54, 1984, pp.486.

26. Shapiro, R.A. and Murman, E.M.: "Adaptive Finite Element Methods for the Euler Equations", AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, 1988. AIAA-88-0034.

27. Devloo, P., Oden, J.T. and Pattani, P.: "An h-p Adaptive Finite Element Method for the Numerical Simulation of Compressible Flow", to be published in Computer Methods in Applied Mechanics and Engineering.

28. Miller, K.: " Recent Results on Finite Element Methods with Moving Nodes", Accuracy Estimates and Adaptive Refinements in Finite Element Compuattions. Ed. by Babuska et al, Wiley, Chichester, 1986.

29. Eiseman, P.R. and Erlebacher, G.: "Grid Generation for the Solution of Partial Differential Equations", ICASE Report no.87-57, NASA, LaRC, Hampton, Virginia, 1987.

30. Erlebacher, G. and Eiseman, P.R.: "Adaptive Triangular Mesh Generation", AIAA 17th Fluid Dynamics, Plasma Dynamics and Lasers Conference, 1984. AIAA-84-1607.

31. Abolhassani, J.S.: " Topology and Grid Adaptation for High-Speed Flow Computations", Ph.D Dissertation, Old Dominion University, Norfolk, Virginia, 1987.

32. Lohner, R., Morgan, K. and Zienkiewicz, O.C.: " An Adaptive Finite Element Procedure for Compressible High Speed Flows", Computational Methods in Applied Mechanics and Engineering, Vol 51, 1985, pp.441-465.

33. Ramakrishnan, R., Thornton, E.A. and Bey, K.S.: "Finite Element Analysis of High Speed Compressible Flows Using Mesh Refinement / Movement Procedures" Proceedings of the 7th International Conference on Finite

Element Methods in Flow Problems, Huntsville, Alabama, 1989, pp.158-164

34. Babuska, I. and Dorr, M.: " Error Estimates for the Combined h and p Versions of the Finite Element Method", Numerical Mathematics, Vol 25, 1981, pp.257-277.

35. Shephard, M.S., Gallagher, R.H. and Abel, J.F.: "Synthesis of Near-Optimum Finite Element Grids with Interactive Graphics", International Journal for Numerical Methods in Engineering, Vol 15, 1980, pp.1021-1039.

36. Peraire, J., Vahdati, M., Morgan, K. and Zienkiewicz, O.C.: " Adaptive Remeshing for Compressible Flow Computations", Journal of Computational Physics, Vol 72, No. 2, 1987, pp.449-466.

37. Peraire, J., Peiro, J., Formaggia, L., Morgan, K., and Zienkiewicz, O.C.: "Finite Element Euler Computations in Three Dimensions" , AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, 1988. AIAA-88-0032.

38. Mavriplis, D.J.: " Adaptive Mesh Generation for Viscous Flows Using Delaunay Triangulation", ICASE report no. 88-47, NASA LaRC, Hampton, Virginia, 1988.

39. Baehmann, P.L., Witchen, S.L., Shephard, M.S., Grice, K.R. and Yerry, M.A.: "Robust, Geometrically Based Automatic Mesh Generation", International Journal for Numerical Methods in Engineering, Vol 24, 1987, pp.1043-1078.

40. Diaz, A.R., Kikuchi, N. and Taylor, J.E.: "A Method of Grid Optimization for Finite Element Methods", Computer Methods in Apllied Mechanics and Engineering, Vol 41, 1983, pp.29-45.

41. Demkowicz, L., Devloo, Ph. and Oden, J.T.: "On An h-Type Mesh Refinement Strategy Based on Minimization of Interpolation Errors", Computer Methods in Applied Mechanics and Engineering, Vol 53, 1985, pp.67-89.

42. Oden, J.T., and Carey, G.F.: " Finite Elements : Mathematical Aspects," Prentice-Hall, Englewood Clifs, 1981.

43. Babuska, I. and Rheinboldt, W.C.: " Analysis of Optimal Finite-Element Meshes in R^1," Mathematics of Computation, Vol 33, No. 146, pp.435-463.

44. Thareja, R.R., Stewart, J.R., Hassan, O., Morgan, K. and Peraire, J.: " A Point Implicit Unstructured Grid Solver for the Euler and Navier-Stokes Equations," AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, 1988. AIAA-88-0036.

45. Carslaw, H.S. and Jaeger J.C.: "Conduction of Heat in Solids", Second Edition, Oxford University Press, 1959, pp.170-171.

46. Roe, P.L.: "Approximate Riemann Solvers, Parameter Vectors and Difference Scheme", Journal of Computational Physics, Vol 43, 1981, pp.357-372.

47. Gnoffo, P.A.: " Application of Program Laura to Three-Dimensional AOTV Flowfields", presented at the AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, 1986. AIAA-86-0565.

48. Yee, H.T., Warming R.F., and Harten, A.: " Implicit TVD Schemes for Steady State Calculations," Journal of Computational Physics, Vol 57, 1985, pp.327-360.

49. Woodward, P. and Collela, P.: " The Numerical Simulation of Two-Dimensional Fluid Flows with Strong Shocks", Journal of Computational Physics, Vol 54, 1984, pp.115-143.

50. Harris, J.E. and Blanchard, K.: " Computer Program for Solving Laminar, Transitional or Turbulent Compressible Boundary Layer Equations for Two-Dimensional and Axi-symmetric Flow", NASA Technical Memorandum-83207, 1982.

51. Holden, M.S.: "A Study of Flow Separation in Regions of Shock Wave-Boundary Layer Interaction in Hypersonic Flow", AIAA 11th Fluid and Plasma Dynamics Conference, Seattle, Washington, 1978. AIAA-78-169.

52. Ramakrishnan, R., Thornton, E.A., and Wieting, A.R.: " An Adaptive Finite Element Procedure for Compressible Flows with Strong Viscous-Inviscid Interactions", AIAA Thermophysics, Plasmadynamics and Lasers Conference, San Antonio, Texas, 1988. AIAA-88-2694.

53. Bertram, M.H. and Blackstock, T.A.: "Some Simple Solutions to the Problem of Predicting Boundary Layer Self-Induced Pressure", NASA TN D-798, April, 1961.

54. "Laminar Flow Data Base", Calspan Corporation, 1987.

Appendix A

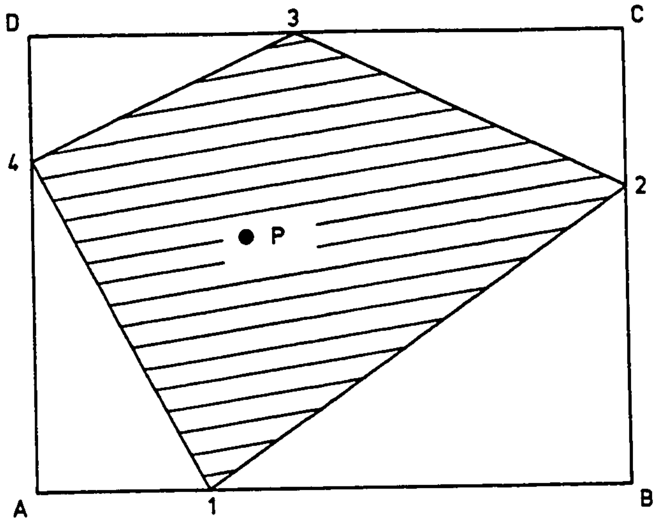
SEARCH ALGORITHM

The interpolation of mesh generation parameters in the new mesh uses a search algorithm to identify the element on the background mesh where the point of interest P lies. This search is made on a global basis since the background mesh is unstructured.

An array is created in the program that gives

XMIN	-	minimum X coordinate
XMAX	-	maximum X coordinate
YMIN	-	minimum Y coordinate
YMAX	-	maximum Y coordinate

for all the elements on the background mesh. As the search begins the coordinates of P are checked to determine if they lie within the limits of each element. This check is made one limit at a time so that a negative response avoids the remaining checks. When the response is possitive for all four checks, the point lies in the neighbourhood of an element shown as the rectangle ABCD in the following sketch.



A second set of checks are made to determine if point P lies within the element (shaded region). The second set of checks are made by computing the areas of the triangles 12p, 23p, 34p, and 41p where 1, 2, 3, and 4 are the nodes numbered in an counter-clockwise manner. If any of the areas are negative or zero (within a specified tolerance) the response is negative and the search proceeds to the next element.

Appendix B
COMPUTATION OF DERIVATIVES

For the elements used in the remeshing method the value of the dependent variable within an element is interpolated from nodal values by

$$\phi(x,y) = [N(x,y)]\{\phi\} \qquad \text{(B.1)}$$

where for a triangle the interpolation functions [N] are linear and for a quadrilateral the interpolation functions are bilinear. From equation (B.1), first derivatives can be computed, but it is not possible to compute the second derivatives that are needed for remeshing. As an alternate approach the following procedure [36] is used.

From equation (B.1) element first derivatives are computed by direct differentiation. For example,

$$\frac{\partial \phi^e}{\partial x} = \left[\frac{\partial N}{\partial x}\right]\{\phi\} \qquad \text{(B.2)}$$

values of the first derivatives are computed at nodal points by assembling system equations from element contributions of the form,

$$\int_A \{N\}[N]\, dA \left\{\frac{\partial \phi}{\partial x}\right\} = \int_A \{N\}\, dA \,\frac{\partial \phi^e}{\partial x} \qquad \text{(B.3)}$$

The coefficient matrix on the left-hand side of (B.3) is diagonalized to yield an explicit set of equations that are solved for the nodal values of the derivatives. The procedure given by equation (B.3) may be interpreted as computing the nodal derivative as a weighted average of the derivatives from the elements surrounding the node. Element area factors serve as the weighting factors.

The computation of the second derivative follows the same steps. Element second derivatives are computed from nodal derivatives by

$$\frac{\partial^2 \phi^e}{\partial x^2} = \left[\frac{\partial N}{\partial x}\right] \left\{\frac{\partial \phi}{\partial x}\right\} \qquad (B.4)$$

and then second derivatives are computed from system equations assembled from,

$$\int_A \{N\}\,[N]\;dA \left\{\frac{\partial^2 \phi}{\partial x^2}\right\} = \int_A \{N\}\;dA\,\frac{\partial^2 \phi^e}{\partial x^2} \qquad (B.5)$$

The procedure described by equations (B.1) - (B.5) lacks mathematical rigor, but it has proven adequate for computing the second derivatives needed for remeshing. The second derivatives computed on the boundaries typically are less accurate than those computed at interior nodes.

Appendix C

ELEMENTS OF THE MATRIX [As]

The matrices $[R]^{-1}$, $[\Lambda]$, and $[R]$ that define the matrix $[A_S]$ in equation (5.23) of Chapter 5 are given below.

$$[R]^{-1} = \begin{bmatrix} \frac{-1}{C^2} & 0 & \frac{1}{2C^2} & \frac{1}{2C^2} \\ \frac{-u}{C^2} & -\eta_y & \frac{u + C_x}{2C^2} & \frac{u - C_x}{2C^2} \\ \frac{-v}{C^2} & -\eta_x & \frac{v + C_y}{2C^2} & \frac{v - C_y}{2C^2} \\ \frac{-\alpha}{C^2} & \gamma & \frac{\alpha + U_c}{2C^2} + \frac{1}{2\beta} & \frac{\alpha - U_c}{2C^2} + \frac{1}{2\beta} \end{bmatrix} \qquad (C.1)$$

$$[|\Lambda|] = \begin{bmatrix} |U| & 0 & 0 & 0 \\ 0 & |U| & 0 & 0 \\ 0 & 0 & |U + C| & 0 \\ 0 & 0 & 0 & |U - C| \end{bmatrix} \qquad (C.2)$$

$$[R] = \begin{bmatrix} \alpha\beta - C^2 & -\beta u & -\beta v & \beta \\ \gamma & -\eta_y & \eta_x & 0 \\ \alpha\beta - UC & C_x - \beta u & C_y - \beta v & \beta \\ \alpha\beta + UC & -C_x - \beta u & -C_y - \beta v & \beta \end{bmatrix} \qquad (C.3)$$

where

$$u = U\,n_x + V\,l_x = U\,n_x - V\,n_y$$

$$v = U\,n_y + V\,l_y = U\,n_y + V\,n_x \qquad (C.4)$$

$$\begin{aligned} u &= u\,n_x + v\,n_y \\ v &= u\,l_x + v\,l_y \\ &= -u\,n_y + v\,n_x \end{aligned} \qquad (C.5)$$

$$C_x = C\,n_x \;;\; C_y = C\,n_y$$

$$l_x = -\,n_y \;;\; l_y = n_x \qquad (C.6)$$

$$\beta = \gamma - 1 \;;\; \alpha = \frac{\left(u^2 + v^2\right)}{2} \;;\; C^2 = \frac{\gamma p}{\rho}$$

$$p = (\gamma - 1)\left[\rho\varepsilon - \frac{1}{2}\rho\left(u^2 + v^2\right)\right] \qquad (C.7)$$

$$\varepsilon = e + \frac{1}{2}\left(u^2 + v^2\right) \;;\quad e = \frac{p}{(\gamma - 1)\,\rho}$$

$$h = \gamma e = \gamma\left[\varepsilon - \frac{1}{2}\left(u^2 + v^2\right)\right]$$

$$\text{Total enthalpy } H = h + \frac{1}{2}\left(u^2 + v^2\right) = \gamma\varepsilon - \frac{(\gamma - 1)}{2}\left(u^2 + v^2\right) \qquad (C.8)$$

Appendix D

ELEMENTS OF THE MATRIX [B_S]

The components of the matrix [B_S] related to the viscous fluxes in equation (5.46) (shown as equation D.1 below)

$$\left[I + \frac{\Delta t}{2\Omega_E} \sum_{s=1}^{NSIDE} \delta_S \left([|A_s|]^* + 2[B_S] \right) \right] \{\Delta U_E\} =$$

$$- \frac{\Delta t}{2\Omega_E} \sum_{s=1}^{NSIDE} \delta_S \left[\{F_{nR}\}^* + \{F_{nE}\}^m - [|A_s|]^* \left(\{U_R\}^* - \{U_E\}^* \right) + 2 \{G_{nS}\}^m \right] \tag{D.1}$$

are as follows,

$$B_{11} = B_{12} = B_{13} = B_{14} = 0$$

$$B_{21} = \frac{\left(\emptyset_1 u_e + \emptyset_2 v_e\right)}{\rho_e} \; ; \; B_{22} = \frac{-\emptyset_1}{\rho_e} \; ; \; B_{23} = \frac{-\emptyset_2}{\rho_e} \; ; \; B_{24} = 0$$

$$B_{31} = \frac{\left(\emptyset_3 u_e + \emptyset_4 v_e\right)}{\rho_e} \; ; \; B_{32} = \frac{-\emptyset_3}{\rho_e} \; ; \; B_{33} = \frac{-\emptyset_4}{\rho_e} \; ; \; B_{34} = 0$$

$$B_{41} = \frac{\left(\emptyset_5 u_e + \emptyset_6 v_e\right)}{\rho_e} - \frac{\emptyset_7 \left(u^2 + v^2 - \varepsilon\right)}{\rho_e C_V}$$

$$B_{42} = \frac{\left(-\emptyset_5 + \emptyset_7 u_e\right)}{\rho_e} \; ; \; B_{43} = \frac{\left(-\emptyset_6 + \emptyset_7 v_e\right)}{\rho_e C_V} \; ; \; B_{44} = \frac{\emptyset_7}{\rho_e}$$

$$\varnothing_1 = \mu\, \delta_S \left(\frac{4}{3} n_x b_{S_e} + n_y c_{S_e} \right) ; \quad \varnothing_2 = \mu\, \delta_S \left(-\frac{2}{3} n_x c_{S_e} + n_y b_{S_e} \right)$$

$$\varnothing_3 = \mu\, \delta_S \left(n_x c_{S_e} - \frac{2}{3} n_y b_{S_e} \right) ; \quad \varnothing_4 = \mu\, \delta_S \left(- n_x b_{S_e} + \frac{4}{3} n_y b_{S_e} \right)$$

$$\varnothing_5 = u_e \varnothing_1 + v_e \varnothing_3 ; \quad \varnothing_6 = u_e \varnothing_2 + v_e \varnothing_4$$

$$\varnothing_7 = \frac{\mu\, \delta_S\, C_P \left(n_x b_{S_e} + n_y c_{S_e} \right)}{Pr}$$

$$b_{S_e} = \frac{1}{2} \left(\widetilde{b_{M_e}} + \widetilde{b_{N_e}} \right) ; \quad \widetilde{b_{M_e}} = \frac{1}{(M_L)_M} \int_{\Omega_e} \frac{\partial N_M}{\partial x} \, d\Omega$$

$$c_{S_e} = \frac{1}{2} \left(\widetilde{c_{M_e}} + \widetilde{c_{N_e}} \right) ; \quad \widetilde{c_{M_e}} = \frac{1}{(M_L)_M} \int_{\Omega_e} \frac{\partial N_M}{\partial y} \, d\Omega$$

Appendix E

GRID REFINEMENT STUDY FOR A VISCOUS PROBLEM

To determine the order of accuracy of the point-implicit upwind algorithm used in the current work for the compressible flow analysis, a uniform grid refinement study was performed for a simple viscous problem. The problem statement is shown in Figure E.1. The inlet profile for the problem is taken from a boundary layer solution.

A uniform mesh consisting of 800 elements and 861 nodes (41 nodes in x direction and 21 nodes in the y direction) was chosen as an initial mesh (this mesh looks similar to that shown in Figure 25 in the main text). Two uniform refinements in the y direction were performed on the initial mesh to give a mesh consisting of 1600 elements and 1681 nodes (41X41) and a mesh with 3200 elements and 3321 nodes(41X81). Refinement was performed in the y direction alone, since change in the flow variables was predominant in this direction. The element dimension in the y direction "h", in these meshes was 0.012in, 0.006in, 0.003in respectively.

Finite element solutions were obtained on these meshes using the point-implicit flow algorithm. One more refinement is performed on the third mesh to give a fourth mesh consisting of 6400 elements and 6601 nodes (41X161) and the finite element solution on this mesh was treated as the exact solution for

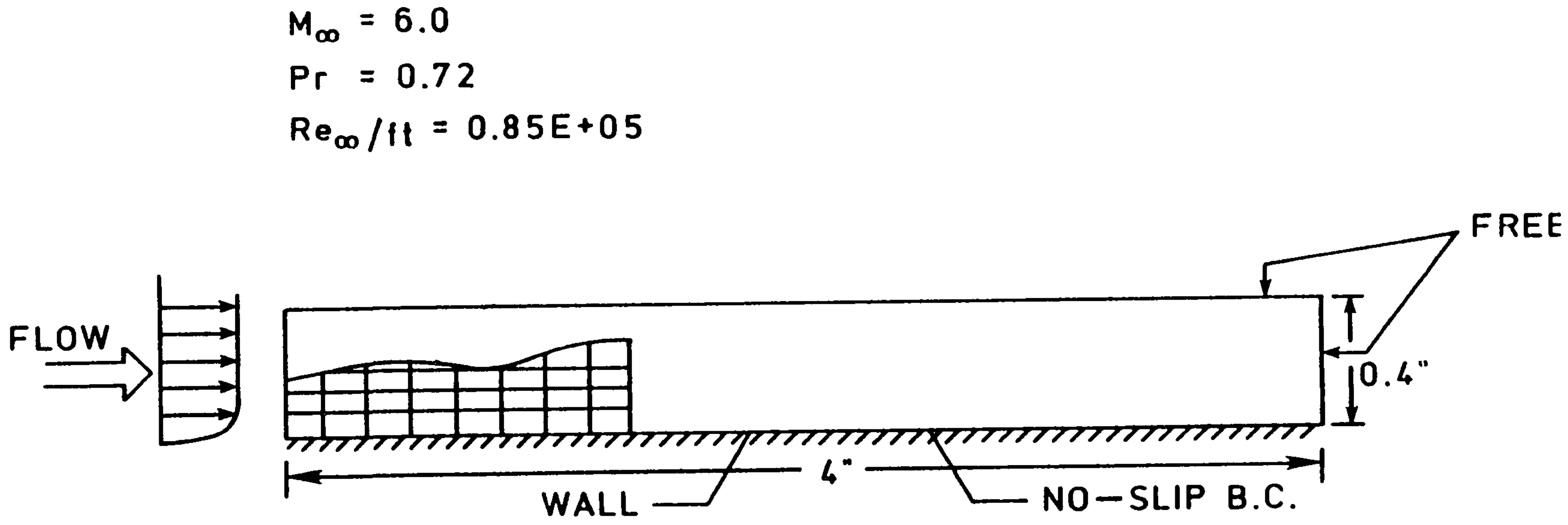


Fig. E.1 Problem statement for hypersonic boundary layer flow

comparison since it was the best solution available. For each of the first three meshes, L_2-norm of the error was computed based on the finite element solution and the exact solution (finite element solution on the most refined mesh) for two conservation variables, density (ρ) and density*u-velocity (ρu) respectively. Results are shown in Figures E.2 and E.3. The gradient of the linear curve relating the logorithm of the error with the logorithm of h was found out to be 1.49 for density and 1.51 for density*u-velocity. It can be concluded from these results that the order of accuracy of the point-implicit, cell-centered upwind finite element algorithm for a viscous problem with no strong inviscid effects is close to 1.5.

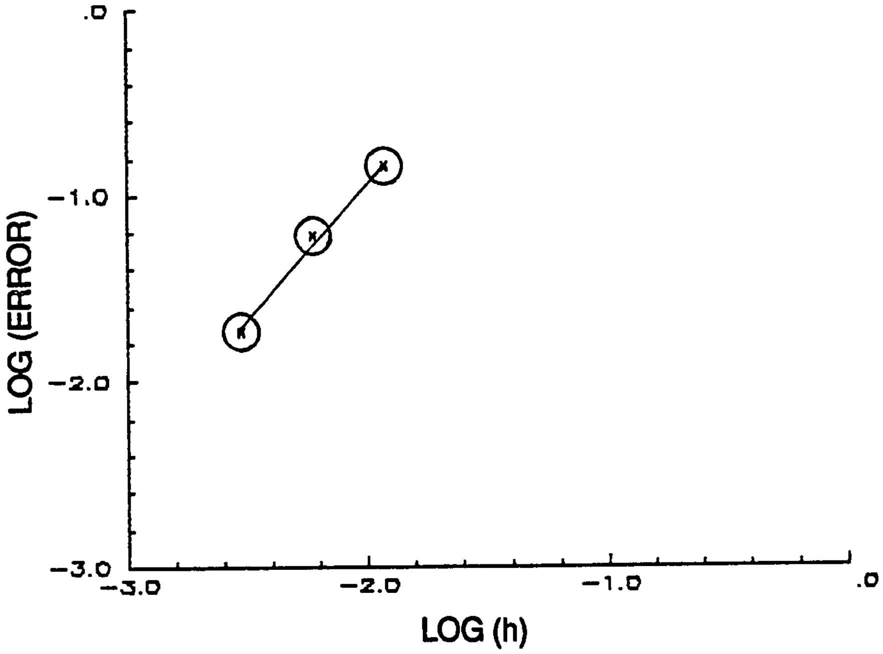


Fig. E.2 Density error norm versus element dimension for viscous algorithm convergence study

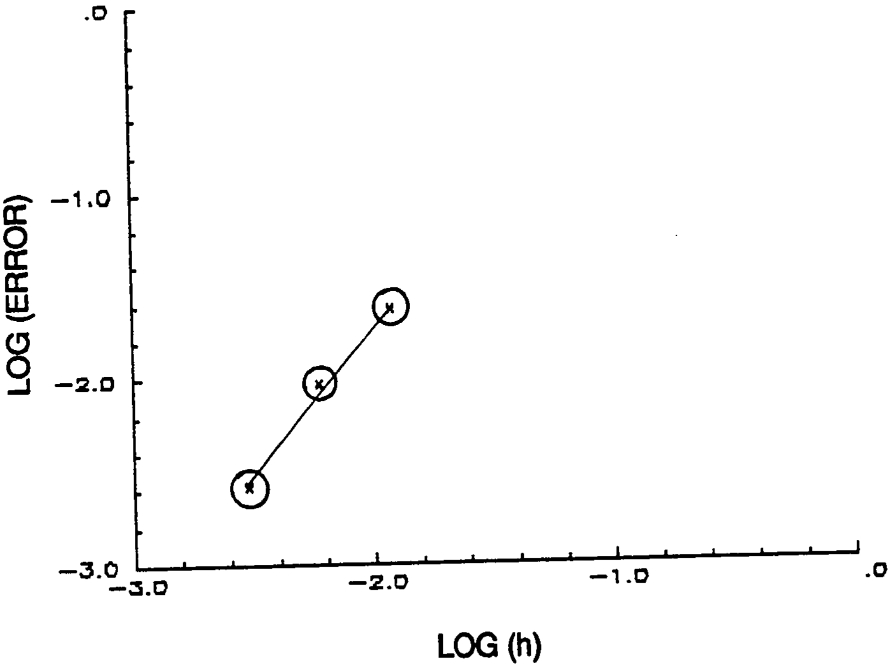


Fig. E.3 (Density*u-velocity) error norm versus element dimension for viscous algorithm convergence study

BIOGRAPHY

Gururaja Vemaganti was born in southern part of India in July, 1957. After formal schooling, he graduated from Osmania University, Hyderabad, India in 1979 majoring in mechanical engineering. Because of the interest he developed in thermal and fluid sciences during his undergraduate education, he pursued his higher studies in the Indian Institute of Technology at Madras specializing in heat transfer and thermal power. He obtained his M.Tech degree in this institute in summer of 1981. After teaching for one year in India he went to Canada in 1982 for higher education. He obtained his M.Engg degree in aeronautical engineering from Carlton university in 1984. His masters thesis was on the finite element method applied to incompressible flows. He came to the United States in 1984 to pursue his Ph.D in mechanical engineering at the Old Dominion University. His interest in the application of the finite element method to the flow problems brought him close to Aerothermal Loads Branch at NASA LaRC which is committed for research in high speed flows.