Structured and Unstructured Remeshing Method for High-Speed Flows

Gururaja R. Vemaganti*
Lockheed Engineering and Sciences Company, Hampton, Virginia 23666
Earl A. Thornton†
University of Virginia, Charlottesville, Virginia 22901
and
Allan R. Wieting‡
NASA Langley Research Center, Hampton, Virginia 23665

An adaptive remeshing method using both triangular and quadrilateral elements suitable for high-speed flows is presented. For inviscid flows, the method generates completely unstructured meshes. For viscous flows, the boundary-layer edge is identified adaptively and a structured mesh is generated in the boundary layer; an unstructured mesh is generated in the inviscid region. Examples of inviscid and viscous mesh adaptations for high-speed flows are presented. A comparison is made between first-order and higher-order finite element algorithms when used in association with the remeshing method.

Nomenclature

C_H	= coefficient of heat transfer
C_p	= coefficient of pressure
C_v	= specific heat at constant volume
h	= dimension of an element, spacing
Pr	= Prandtl number
p	= pressure
q	= heat transfer rate
Re	= Reynolds number
T	= temperature
t	= time
u	= velocity
X	= principal axis
x,y,z	= coordinates of the reference frame
α	= vector of stretching
γ	= ratio of specific heats
λ	= eigenvalue of the second derivative tensor
ρ	= density
ϕ	= dependent variable

Subscripts

res	= freestream stagnation conditions
w	= wall
∞	= freestream
1,2	= principal directions

Introduction

DESIGN, performance, and optimization studies for aerospace vehicles require techniques that can calculate aerodynamic heating rates accurately and efficiently. Aerodynamic heating has a significant effect on the thermal-structural response of high-speed flight vehicles. Finite element (FE) methods show promise in high-speed flow applications because of their unstructured mesh adaptation capabilities and their ease in handling complex geometries. In addition, FE flowfield solution methodology can be easily coupled to existing finite element structural or heat transfer analysis codes to solve multidisciplinary interaction problems.[1]

Adaptive unstructured mesh techniques have been the focus of research efforts in finite element-computational fluid dynamics (FE-CFD) in recent years. These techniques fall into two categories: adaptive refinement/derefinement methods and adaptive remeshing methods. Adaptive refinement methods include "h" methods, wherein the mesh is refined or derefined when local error indicators fall outside of preassigned upper or lower bounds; "r" methods, in which the mesh is automatically distorted to equidistribute errors; and "p" methods, where the order of the interpolation polynomial is increased or decreased depending on the local error. These methods have limitations in high-speed flows. In the neighborhood of strictly one-dimensional features like shocks and boundary layers, the "h" refinement method is not efficient, since the method increases the number of unknowns significantly from refinement to refinement and does not alter the original location of the nodal points. Though these problems do not arise with the "r" method of refinement, sometimes the "r" method produces highly distorted elements. The "p" method of refinement is more complicated because its implementation requires major changes in the analysis codes.

An alternate approach for the development of improved finite elment meshes is to use the the results on the current mesh to guide the generation of an entirely new mesh. This approach is referred to as adaptive remeshing. In adaptive remeshing, a posteriori parameters are converted into a priori parameters and used for the development of a new mesh. A major advantage of this approach is the improvement in the quality of the solution without increasing the number of unknowns significantly. The first finite element adaptive remeshing method used triangular elements.[2] The method was extended to include a combination of quadrilateral and triangular elements and applied to thermal problems in Ref. 3. The adaptive remeshing method has been successfully employed for finite element thermal analysis[3,4] and structural analysis.[4]

This paper presents an adaptive remeshing approach for finite element analysis of high-speed viscous flows, where

Received Nov. 27, 1989; presented as Paper 90-0401 at the AIAA 28th Aerospace Sciences Meeting, Reno, NV, Jan. 8-11, 1990; revision received Sept. 12, 1990; accepted for publication Sept. 17, 1990.

*Senior Engineer. Member AIAA.

†Professor, Department of Mechanical and Aerospace Engineering. Associate Fellow AIAA.

‡Head, Aerothermal Loads Branch. Member AIAA.

structured meshes consisting of only quadrilateral elements are created in boundary layers and unstructured meshes consisting of a mixture of quadrilateral and triangular elements are created in inviscid regions. The boundary-layer region is identified adaptively from a previous solution. Both the structured and unstructured meshes are generated in an adaptive manner in a single execution of the remeshing program.

The remeshing algorithm is applied to high-speed flow problems in conjunction with a cell-centered point-implicit algorithm[5] and a streamline upwinding Petrov-Galerkin/least squares finite element algorithm.[6] Two example problems are presented in this paper to illustrate the capabilities of the remeshing procedure to capture the flow features effectively. The effects of the solution algorithm are illustrated in the first example of inviscid shock reflection by comparing solutions from the two algorithms with inviscid theory. A Navier-Stokes solution for flow over a compression corner illustrates the ability of the method to identify the boundary-layer edge and adaptively remesh the viscous and inviscid regions. Improvement in the quality of the solution through successive remeshing is demonstrated for both cases.

Adaptive Remeshing Method

Unstructured 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 nodal points. The method becomes adaptive when these parameters are used to construct a subsequent 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. This method also introduces stretching of the elements along strictly one-dimensional features like shocks (a larger dimension of the element along the shock and a smaller dimension normal to the shock), which is highly desirable for high-speed flows. Proper clustering of the elements in regions of high gradients is achieved 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 discretization change from mesh to mesh. With remeshing, the quality of the solution may sometimes be improved significantly without increasing the total number of unknowns. In some cases, solution quality increases even with a reduction in the number of unknowns.

The crude initial mesh in this method is referred to as the background mesh; the mesh must have sufficient points to capture the salient flow features. This initial mesh can be generated with fewer elements, and mesh generation parameters may be specified at the nodal points of the mesh, since a solution may not be attainable on this mesh. Based on this information on the initial mesh, 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 locates the element on the background mesh to interpolate these parameters for a new point.[3] 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. Examples of successive remeshes are illustrated in Ref. 3.

The mesh generation parameters used for the construction of a new mesh are: 1) two components of a vector α, along which an element is to be stretched, 2) a length h_1 normal to this vector, and 3) a length h_2 parallel 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, 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[7] to determine the error estimation is by computing the interpolation error. This method has the advantage of not requiring the solution of local auxiliary problems for an error indicator for each element, but it also has the disadvantages, among other things, of requiring the computation of higher order derivatives over each element. A numerical scheme to compute higher-order derivatives (described in Ref. 3) lacks mathematical rigor, but it has proven adequate for all practical purposes.

As an illustration, in one dimension, the solution error is estimated by $h^2|\partial^2\phi/\partial x^2|$. An optimum mesh is obtained when this error is equally distributed by requiring

$$h^2\left|\frac{\partial^2\phi}{\partial x^2}\right| = \text{Constant} \tag{1}$$

In an extension of this approach to two dimensions,[2] 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:

$$\lambda_1 = \left|\frac{\partial^2\phi}{\partial X_1^2}\right|, \qquad \lambda_2 = \left|\frac{\partial^2\phi}{\partial X_2^2}\right| \tag{2}$$

Now the equidistribution of error principle in one dimension is applied to each of these principal directions separately, resulting in

$$h_1^2\lambda_1 = h_2^2\lambda_2 = \text{Constant} \tag{3}$$

where h_1 and h_2 are the spacings required in X_1 and X_2, respectively. The constant on the right side of Eq. (3) is computed as $h_{\min}^2|\lambda_{\max}|$, where $h_{\min}$ is the minimum spacing specified by the user and $\lambda_{\max}$ is the maximum eigenvalue in the whole domain. Maximum allowable values of $h_{\max}$ and the aspect ratio are specified by the user to provide an upper bound for these values. In addition, the shape of the element is constrained by limits on the internal angles so that a poorly shaped element is not created. Usually these limits are between 45 to 135 deg for a quadrilateral element and 30 to 120 deg for a triangle.

The mesh generation process starts with boundary discretization. Boundary segments joining fixed boundary nodes are ordered in a counterclockwise manner for an external boundary and ordered in a clockwise manner for an internal boundary, thus defining a hole within the domain. This way the domain always exists on the left side of the boundary. Additional boundary nodes are included to satisfy the spacing requirements compatible with the background mesh. Each boundary segment is discretized in order until the entire boundary is covered.

The mesh generation process is based on an advancing front technique similar to the method proposed in Ref. 8. 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 continues, the front changes 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. Thus, the front can be defined as a chain of line segments surrounding the domain that remains to be discretized. The front changes its shape constantly during the construction process and vanishes when the mesh is complete. Element generation in this method takes place along with node creation and always proceeds from the smallest frontal segment, giving priority to regions that 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 a quadrilateral element and the second option creates a triangular element. Once the mesh is complete, a post-process combines two triangles at a time, wherever possible, to create as many well-shaped quadrilaterals as possible. These options are explained in detail in Ref. 3.

Structured Remeshing

The unstructured remeshing method discussed above is directly applicable to 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. However, the unstructured meshes generated in this manner are not 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 more 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, may require aspect ratios of the order of 1000. To accommodate these very high aspect ratios, the unstructured remeshing method generates very acute or obtuse triangles whose internal angles are not within the acceptable limits, as mentioned earlier.

Reference 5 used a structured mesh generating algorithm to create elements of large aspect ratio 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 priori determination of the edge of the boundary layer is difficult in several problems, for example, in strong shock boundary-layer interactions. Moreover, if the structured mesh is not created in an adaptive manner, there is a possibility of overprescribing 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. Thus, a method of implicitly determining the boundary-layer region is desirable.

A 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 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 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 unstructured remeshing during the boundary layer meshing, so that at the end of the process the front defines the edge of the inviscid flowfield. Generation of elements in a simple boundary layer and the final front defining the inviscid region are shown in Fig. 1. 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 different indicators can be used for remeshing in both regions.

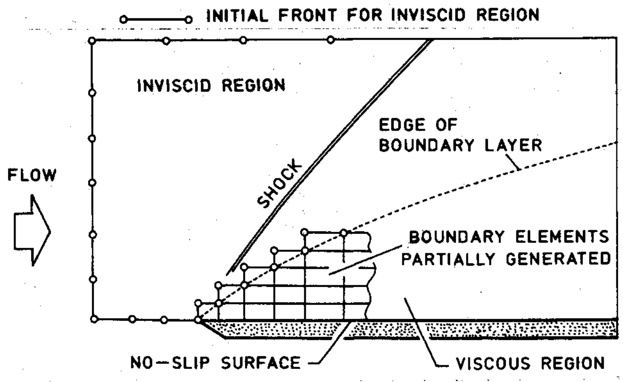


Fig. 1 Structured and unstructured remeshing for a simple viscous flow.

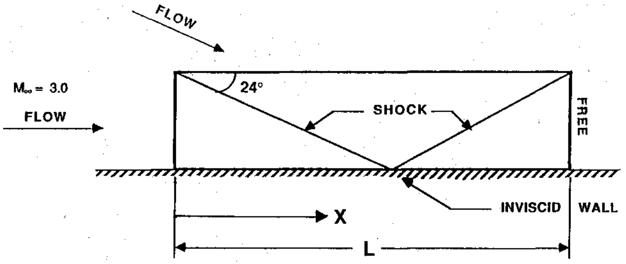
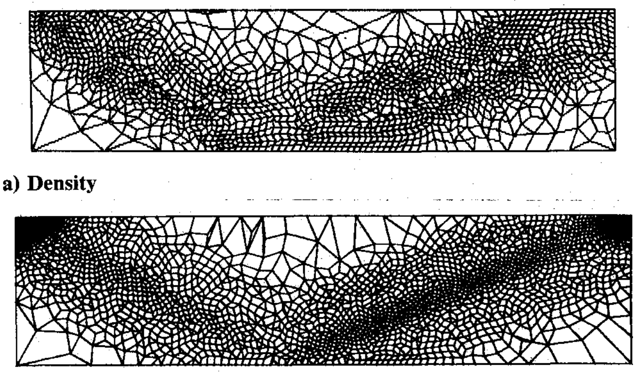


Fig. 2 Problem statement for the inviscid shock reflection.



a) Density

b) Density plus magnitude of gradient of density

Fig. 3 Influence of the error indicators on remeshing for inviscid shock reflection.

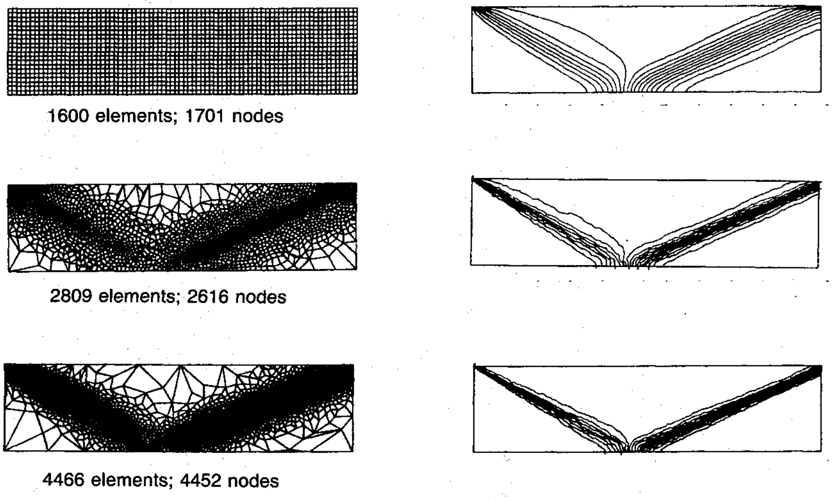
In developing the adaptive structured remeshing method for boundary layers, the following assumptions are made: 1) The maximum gradient of the flow variable, used as an indicator for remeshing, occurs at the no-slip surface. Principal directions of the second derivative tensor of the indicator coincide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

Influence of the Indicators

The selection of the dependent variable as an indicator for remeshing is a vital aspect for remeshing. Temperature is a logical choice as an indicator for thermal problems.[3,4] Similarly, density has been used successfully as an indicator in solving several inviscid problems. But for problems where different physical phenomenon occur in separate regions of the flow domain, multiple indicators may be required to produce effective meshes. An example is a compressible viscous flow, where different indicators may be required for resolving shocks, free shear layers, and boundary layers. Some of the choices for such a problem are discussed in this paper. But further research is required to establish a robust set of indicators for a general viscous problem. In many instances, choice of indicators is problem dependent and may be dependent on the solution algorithm. These ideas will be illustrated later in the numerical examples.

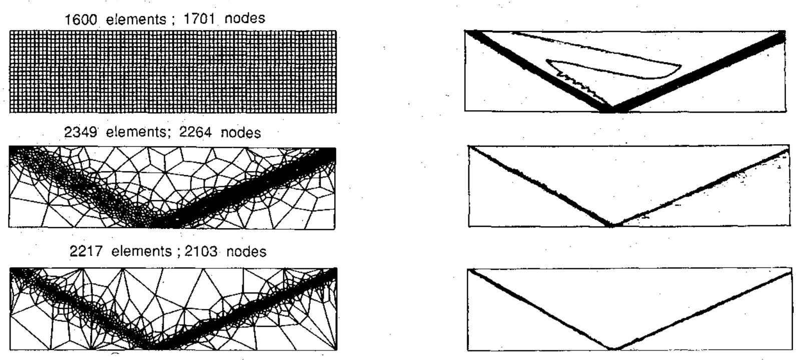
Solution Algorithms

The ability of an adaptive refinement scheme to capture complex flow details depends on the capability of the numeri-


1600 elements; 1701 nodes
2809 elements; 2616 nodes
4466 elements; 4452 nodes

a) Meshes b) Density contours

Fig. 4 Successive adaptive remeshing and solution sequence for the inviscid shock reflection with the point-implicit algorithm.


1600 elements ; 1701 nodes
2349 elements; 2264 nodes
2217 elements ; 2103 nodes

a) Meshes b) Density contours

Fig. 5 Successive adaptive remeshing and solution sequence for the inviscid shock reflection with the SUPG/LS algorithm.

cal scheme to solve the governing equations, i.e., the continuity, momentum, and energy equations, effectively. Finite element algorithms are in an early stage of development for high-speed flows. Explicit Taylor-Galerkin algorithms[9] and Runge-Kutta-Galerkin algorithms[10] are two schemes that have been used to analyze inviscid and viscous flows. Although these schemes are second-order accurate, the convergence rates are excessively slow for viscous flows because of the time step limitation due to their explicit time marching technique.

In this paper two solution algorithms are used: 1) a cell-centered point-implicit algorithm,[5] and 2) a streamline upwinding Petrov-Galerkin/least squares (SUPG/LS) finite element algorithm.[6] The point-implicit algorithm was originally developed as a finite-volume scheme for inviscid flows[3] that was extended for unstructured meshes and the viscous effects in a finite element context.[5] This algorithm is basically first-order accurate, but higher order effects have been incorporated[5] in this algorithm following the concept of flux corrected transport (FCT).[12]

The SUPG/LS formulation for the Navier-Stokes equations is expressed in terms of entropy variables and yields a symmetric advective-diffusion system. A mathematical theory for algorithm convergence for compressible Navier-Stokes equations is described in Ref. 13. The method employs an element by element preconditioned "generalized minimal residual" algorithm as solver.[6]

Applications

Inviscid Shock Reflection

This inviscid flow problem is solved to demonstrate the adaptation of the remeshing method to shocks. The problem statement is given in Fig. 2. An oblique shock at an angle in a Mach 3.0 freestream flow reflects over an inviscid wall. The computational domain is a rectangle with dimensions of 4 × 1. The boundary conditions are shown in the problem statement. The problem was analyzed with both algorithms to investigate the interaction between the algorithms, the error indicators, and the adaptive remeshes.

Starting from the solution on a uniform mesh of 1600 elements, adaptive remeshing was performed using the point-implicit algorithm. Initially, density was used as an error indicator. The solution smeared the shocks over a large band so that the largest second derivatives computed in the error indicators occurred at some distance to either side of the shock. Near the core of this shock, the density varied almost linearly with a small second derivative. From this solution, when the remesh was created small elements were generated on either side of the shock, and large elements were generated at the core of the shock. Figure 3a shows the schematic of a new mesh. Equidistribution of error was employed in computing the parameters for the remesh, and relatively larger elements resulted in the core of the shock, as shown in Fig. 3a. This mesh in turn produced a more diffused solution, and subsequent adaptation failed to resolve the shocks.

To address this problem, the use of multiple indicators was studied. The mesh generation parameters were computed independently, based on two or more variables. On any node on the background mesh, the set of mesh generation parameters with the smallest h_2 value was taken for adaptation. Combinations of density and Mach number, as well as density and temperature, were attempted in generating meshes. These combinations resulted in meshes similar to the mesh shown in Fig. 3a, since 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 (Fig. 3b) which produced smaller elements at the core of the shock. This result was 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 446 elements, respectively, consisting predominantly of quadrilaterals. The adaptive remeshes and the corresponding density contours are shown in Fig. 4.

Before the SUPG/LS algorithm was used to solve the same problem with adaptive remeshing, a convergence study on uniformly refined meshes was performed. The convergence rate study for both algorithms used three uniform meshes consisting of 400, 1600, and 6400 elements, respectively. In these meshes, the element size is uniformly reduced from 0.1 to 0.025. A global error was computed based on the exact solution and the numerical solutions on these meshes. The results of this study indicate that the SUPG/LS algorithm has a faster convergence rate and, hence, a higher order of accuracy than the point-implicit algorithm. Slopes of the straight lines passing through the points corresponding to the global error show that the order of accuracy is close to 1(0.94) for the point-implicit scheme and to 2(1.82) for the SUPG/LS algorithm.

The SUPG/LS algorithm typically requires more CPU time than the point implicit algorithm. The CPU time for the finite element algorithm is proportional to the number of nodes, and the CPU time for the point-implicit algorithm is proportional to the number of element sides. For a large number of elements, the number of nodes and element sides are roughly equal. The SUPG/LS scheme requires about 0.9E-03 CPU s/node/time step on a CRAY-YMP computer, and the point implicit scheme takes about 0.3E-03 CPU s/element side/time step. Although the CPU time spent by the SUPG/LS algorithm is three times more than that spent by the point-implicit algorithm on the same mesh, the SUPG/LS algorithm required four times less number of nodes to obtain the same level of accuracy (level or error); hence, for this problem, it is more efficient.

To conclude study of this problem, solutions were obtained with the SUPG/LS algorithm and adaptive remeshing, starting from the solution on the same initial mesh consisting of 1600 elements; two successive adaptive remeshes were created with 2349 elements and 2217 elements, respectively. Successive remeshes and the corresponding density contours are shown in Fig. 5. Note that the number of elements and nodes in the second remesh is less than that of the first remesh and still the solution quality was improved.

This example illustrates the important interaction between the solution algorithm and the adaptive approach. When the shock was smeared, the adaptive remeshing program required multiple indicators to refine the elements appropriately, but when a crisp shock was predicted, a single indicator was sufficient for the remeshing method to refine the mesh correctly.

Hypersonic Flow over a Compression Corner

The ability of the remeshing method to identify the boundary-layer edge and adaptively construct a structured quadrilateral mesh in the viscous region and an unstructured mesh in the inviscid region 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 deg compression corner are shown in Fig. 6. This problem has been experimentally investigated in Ref. 14. 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.

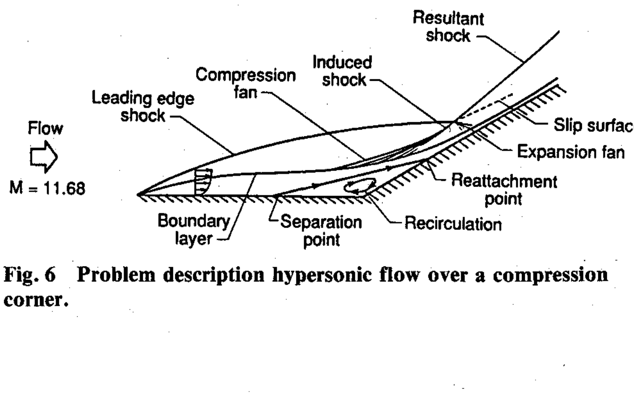


Fig. 6 Problem description hypersonic flow over a compression corner.

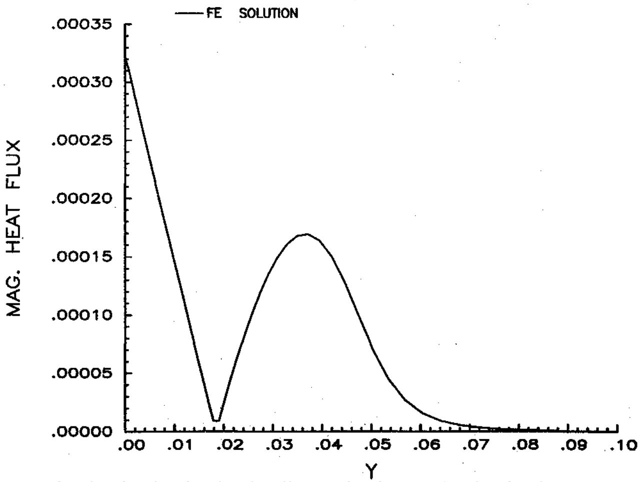


Fig. 7 Magnitude of the heat flux distribution normal to the wall for the compression corner.

In Ref. 15, the problem was solved by dividing the entire flow domain into three regions and using the Runge-Kutta-Galerkin algorithm. The three regions are the sharp leading edge section, the flat plate section, and the ramp section. The results were in good agreement with the strong interaction theory[16] in the first region and with the experimental results of Ref. 14 in the third region. The third region is of interest because 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, which starts at about the midpoint of the flat plate section and goes to the location of shock-shock interaction on the ramp, requires the solution of the full Navier-Stokes equations and has been chosen for the adaptive analysis. The inlet profiles for this region are obtained from the results of Ref. 15. This inlet section is chosen sufficiently far upstream such that the flow separation effects do not influence the inflow. In the current work, results using the point-implicit scheme only are presented.

An initial mesh (not shown) was constructed with 5283 and 5142 elements. A structured mesh is constructed to a nondimensional height of 0.4 from the wall and the remainder of the region is discretized in an unstructured manner. Specified values on a crude background mesh for the mesh generation parameters are used to generate the entire mesh. A solution was obtained after 5000 iterations when the L_2-norm of the residuals of all the conservation variables reduced at least three orders of magnitude.

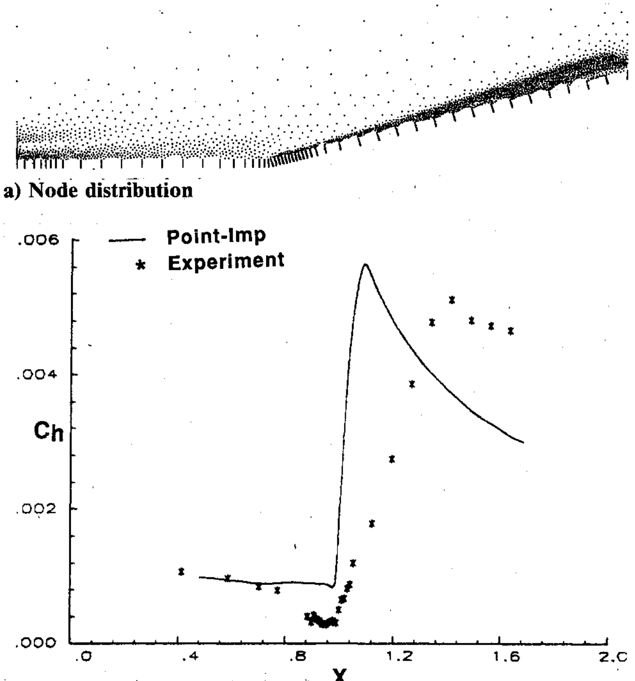
Details of Adaptation

The two variables predominant in the boundary-layer region are the absolute velocity and heat flux. Outside of the boundary layer, neither variable showed the shocks because the absolute velocity change across the shock was not as significant as it was in the boundary layer and the flow is adiabatic across the shock. On the initial mesh, two indicators were employed, where the mesh generation parameters are computed based on the second derivatives of absolute velocity in the boundary layer and on second derivatives of density in the inviscid region. During the remeshing process, a structured mesh is created in the region where the heat flux is predominant, and the unstructured mesh is generated in the inviscid region. The distribution of the absolute value of the heat flux normal to the wall is shown in Fig. 7, which shows a maximum value occurring at the wall that gradually reduces to zero away from the wall. This resulted in the nodal distribution, as shown in Fig. 8a. For the sake of clarity, only the nodal distribution is shown; from the nodal distribution, the boundary layer region and the shock region are identifiable. This first remesh consisted of 3823 elements and 3470 nodes. Initial conditions for the next solution on this mesh were interpolated linearly from the solution on the initial mesh. The solution was iterated until the L_2-norm on the residuals of the conservation variables reduced three orders of magnitude. The wall heat flux distribution is shown in Figs. 8b. In these figures, C_H is the heat transfer coefficient, given by

$$C_H = \frac{q_w}{\rho_\infty u_\infty \gamma c_v [T_{res} - T_w]} \tag{4}$$

This solution does not contain a separation region and, hence, predicted higher heating rates near the corner as well as on the ramp section.

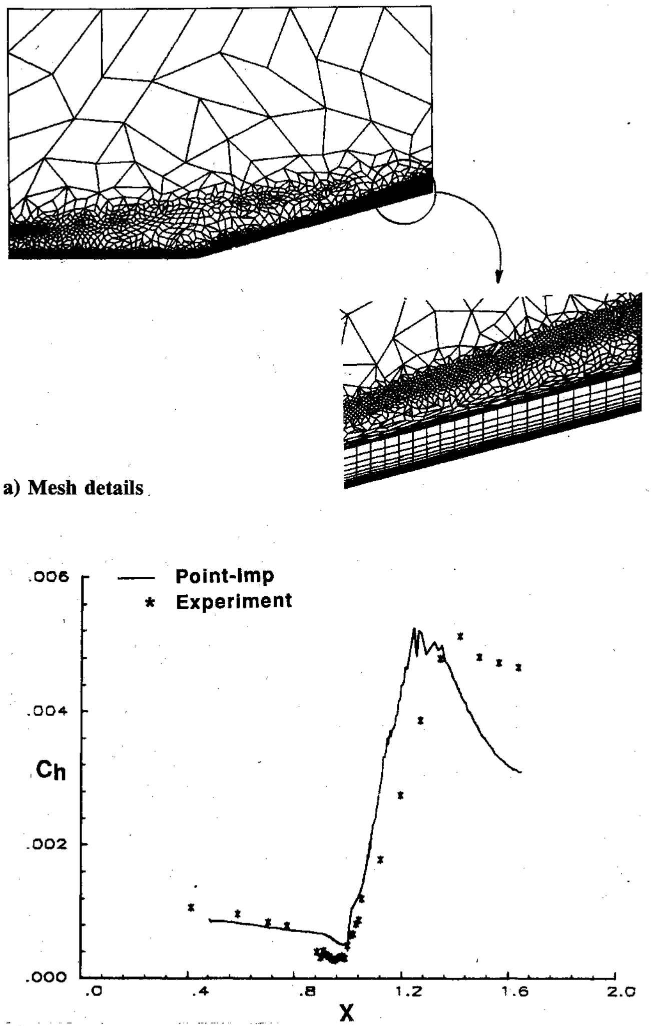
At this point, it was obvious that the mesh was inadequate to capture the flow separation. From the results, it was observed that the tangential velocity would be a better indicator than the absolute velocity in refining the boundary-layer region, the reason being that the gradients of tangential velocity are much stronger than that of absolute velocity in the separation region. After two successive mesh adaptations, 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 14,626 nodes and 14,663 elements and is shown in Fig. 9a. Here, mesh elements were clustered near separation and reattachment points instead of at the corner, as done in the previous meshes.



a) Node distribution

b) Wall heat flux distribution

Fig. 8 First remesh for the hypersonic flow over a compression corner.



a) Mesh details

b) Wall heat flux distribution

Fig. 9 Third remesh for the hypersonic flow over a compression corner.

Converged results obtained on this mesh are presented after 10,000 iterations. The error norm on the residuals of the conservation variables reduced at least four orders of magnitude. The wall heat flux distribution is shown in Fig. 9b. Although separation is predicted further downstream than the experimental value, the heat flux distribution is closer to the experimental data compared to that of earlier meshes. The peak heating rate is within 5% of the peak experimental value but at the wrong location. Out of approximately 9000 elements in the boundary layer, nearly 6000 elements are placed in the separation region because the wall spacing value was two orders of magnitude less than that on the flat plate portion of the domain. Although there is a significant improvement in the overall prediction of the heat flux distribution through successive remeshing, further refinement may be required to improve the heat flux predictions. But further refinement would increase the already large problem size to an imprudent level. The experience with the point implicit algorithm on the preceding inviscid problem suggests that the performance of the algorithm, rather than the mesh, is the reason that better agreement with the data is not obtained.

The problem demonstrates that the viscous remeshing method is working effectively. If the tangential velocity is used as the indicator of viscous dominated regions, structured elements of variable aspect ratio are generated along the wall. These elements effectively define unseparated and separated viscous regions of the flow. Once the structured viscous region of the mesh is generated, the unstructured mesh for the inviscid region is easily generated, using density as the error indicator.

Conclusions

An adaptive unstructured/structured remeshing method suitable for high-speed flows is presented. The method generates structured meshes consisting of quadrilateral elements in boundary layers and unstructured meshes consisting of a combination of quadrilateral and triangular elements in the inviscid region. The edge of the boundary layer that interfaces the viscous and the inviscid regions is determined adaptively, using the absolute value of heat flux. Nodes and elements are created simultaneously following an advancing front technique.

The remeshing method is used in association with a first-order accurate cell-centered point-implicit algorithm and a second-order accurate streamline upwinding Petrov Galerkin/least square (SUPG/LS) algorithm to solve high-speed flow problems. An inviscid and a viscous problem are both presented to show the adaptation capability of the remeshing method. For the problem of inviscid shock reflection, the first-order scheme required multiple indicators for remeshing to completely capture shock discontinuities, and the shock was smeared over several points. The SUPG/LS algorithm on the other hand required fewer number of nodal points to obtain comparable or even better accuracy, but it required more CPU time/node/time step. With either of the solution algorithms, the remeshing method proved to be more efficient and required fewer number of nodal points to get a converged solution than a uniform refinement.

Similarly, the remeshing method improved the quality in the finite element solution for the hypersonic flow over a compression corner. Although successive remeshing starting from a crude initial mesh significantly improved the solution quality, final agreement with the experimental data is only fair. The results suggest that the first-order accurate point implicit algorithm was not capable of predicting the flow features effectively. Absolute velocity seems to be a good indicator for remeshing attached boundary layers, but inadequate for remeshing separated boundary layers. Tangential velocity appears to construct appropriate meshes in the separation region.

The experience of using the structured and unstructured remeshing method for these examples shows that the method can automatically generate high-quality meshes for complex flows. There is a strong interaction between the adaptive remeshing method and the solution algorithm. In conjunction with an effective solution algorithm, the adaptive remeshing methods offers significant potential for producing robust reliable solutions for high-speed viscous flows.

Acknowledgments

The majority of this work was done as a part of the first author's Ph.D. dissertation research supported by NASA Research Grant NSG-1321. The authors would like to thank Pramote Dechaumphai of NASA Langley Research Center for his technical discussions.

References

[1] Thornton, E. A., and Dechaumphai, P., "Coupled Flow, Thermal, and Structural Analysis of Aerodynamically Heated Panels," *Journal of Aircraft*, Vol. 25, Nov. 1988, pp. 1052–1059.

[2] 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.

[3] Thornton, E. A., and Vemaganti, G. R., "Adaptive Remeshing Method for Finite Element Thermal Analysis," *Journal of Thermophysics and Heat Transfer,* Vol. 4, No. 2, 1990, pp. 212–220.

[4] Dechaumphai, P., "Evaluation of an Adaptive Remeshing Technique for Integrated Fluid-Thermal-Structural Analysis," AIAA Paper 90-0555, Jan. 1990.

[5] 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 Paper 88-0036, Jan. 1988.

[6] Hughes, T. J. R., "Recent Progress in the Development and Understanding of SUPG Methods with Special Reference to the Compressible Euler and Navier-Stokes Equations," *International Journal for Numerical Methods in Fluids*, Vol. 7, No. 11, 1987, pp. 1261–1275.

[7] Demkowicz, L., Devloo, P., 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, No. 1, 1985, pp. 67–89.

[8] Lo, S. H., "A New Mesh Generation Scheme for Arbitrary Planar Domains," *International Journal for Numerical Methods in Engineering*, Vol. 21, No. 8, 1985, pp. 1403–1426.

[9] Thornton, E. A., Dechaumphai, P., and Vemaganti, G. R., "A Finite Element Approach for Prediction of Aerothermal Loads," AIAA Paper 86-1050, June 1986.

[10] Ramakrishnan, R., Bey, K. S., and Thornton, E. A., "An Adaptive Quadrilateral and Triangular Finite Element Scheme for Compressible Flows," *AIAA Journal*, Vol. 28, No. 1, 1990, pp. 51–59.

[11] Gnoffo, P. A., "Application of Program Laura to Three-Dimensional AOTV Flowfields," AIAA Paper 86-0565, Jan. 1986.

[12] Boris, J. P., and Book, D. L., "Flux Corrected Transport Algorithm that Works," *Journal of Computational Physics*, Vol. 11, No. 1, 1973, pp. 38–69.

[13] Hughes, T. J. R., Franca, L. P., and Mallet, M., "A New Finite Element Formulation for Computational Fluid Dynamics: VI. Convergence Analysis of the Generalized SUPG Formulation for Linear Time-Dependent Multidimensional Advective-Diffusive Systems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 63, No. 1, 1987, pp. 97–112.

[14] Holden, M. S., "A Study of Flow Separation in Regions of Shock Wave-Boundary-Layer Interaction in Hypersonic Flow," AIAA Paper 78-1169, July 1978.

[15] Ramakrishnan, R., Thornton, E. A., and Wieting, A. R., "An Adaptive Finite Element Procedure for Compressible Flows with Strong Viscous-Inviscid Interactions," AIAA Paper 88-2694, June 1988; and *Journal of Thermophysics and Heat Transfer* (to be published).

[16] 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.

Walter B. Sturek
Associate Editor