FULL LENGTH PAPER

ISOPARAMETRIC FINITE ELEMENT METHOD TO GENERATE STRUCTURED GRID FOR NUMERICAL FLOW SIMULATION

R.C. Mehta
Department of Aeronautical Engineering
Noorul Islam University
Kumaracoil-629 180, India
Email : drrakhab.mehta@gmail.com

Abstract

Eight-nodded quadrilateral finite element is used to generate structured grid for a numerical simulation. The computational domain is sub-divided into number of quadrilateral regions that display the geometry of each sub-domain. The inner and the outer boundaries of the computational domain are described in terms of the surface coordinates. An algebraic homotopy procedure is used to generate grid clustering in order to resolve boundary layer. The structured grid is linked with the flow solver based on finite volume of space discretiztion scheme with multi-stage Runge-Kutta time stepping technique. Examples are illustrated to demonstrate the grid generation procedure and data processing for a forward facing aero-disc spike attached to a hemispherical blunt body at Mach 6 and over a heat shield of a satellite launch vehicle at Mach 0.8 at an angle of attack 5 deg. The present grid generation method is convenient for checking the grid independence test by varying the stretching factor. The quadrilateral grid generated by finite element, vector plot of velocity and contour plots of the computed flow field data are easily drawn with the help of MATLAB.

Introduction

Computational Fluid Dynamics (CFD) has an increasing role in aerodynamic design and flow field analysis of a satellite launch vehicle. As the computer hardware becomes affordable and numerical algorithm became computationally faster, efficient and economical for research and development of new configuration. Thus, the CFD may reduce the number of blow down of wind-tunnel testing of the aerodynamic model. The development of parallel computers has enabled aerodynamist to attempt more realistic geometrical parameters of the configuration with the Navier-Stokes flow solver. Computational fluid dynamics plays a pivotal role in the implementation with the design process of satellite launch vehicle [1, 2] and understanding the complexity of the separated flow [3]. A typical computational fluid dynamics application can be divided into three major steps such as grid generation, flow simulation algorithm, and computed flow field visualization. Numerical solution of fluid dynamics equations requires a spatial discretization of the physical domain of interest. A suitable mesh must be generated before starting the flow computation for a given configuration. Moreover, it becomes mandatory to do grid independent check [4] before applying in the engineering design. The grid generation is the most time consuming process in the numerical flow simulation.

A desirable mesh processes qualities such as sufficient resolution of the surface geometry, fine mesh resolution for viscous flow and smooth variation of the mesh spacing between fine and course grid regions. Many methods for grid generation [5, 6] have been developed in the area of the computational fluid dynamics, which are based on either solution of partial differential equations or algebraic method. These grid generations procedure compete with each other in their relative abilities to obtain improved grid quantities such as smoothness of point distribution, clustering of points of regions of physical interest, and ability to accommodate complex body geometry. An overview of different grid generation techniques in current use has been presented by Thompson [7].

The octree method [8] belong to a class of mesh generation schemes known as tree structure methods,

Paper Code : V69 N4/965-2017. Manuscript received on 03 Nov 2016. Reviewed, revised and accepted as a Full Length Contributed Paper on 03 Nov 2017

which used in solid modeling and computer graphics display methods. In the tesselartion method [9], the user gives a collection of node points and also an arbitrary starting node. The method creates the first simplex element using the neighboring nodes. Then selecting the node point that gives the least distorted element shape generates a neighboring element.

A general method for numerically generating grid is boundary fitted coordinate systems. In this approach, a geometry definition, point distributions on lines, and surface grids are achieved using non-uniform rational B splines [10, 11]. Field lines are obtained by solving a system of Poisson equations [12]. Algebraic grid generation is an efficient means of grid generation in terms of computational speed and memory requirements. The algebraic grid generation method relies on transfinite interpolation procedure to obtain a grid when the boundaries of the grid have been specified with a user described distribution. A hole-re-meshing grid generation method [13] permits efficient mesh generation about aerofoil with time dependent leading edge ice accretions.

Zienkiewicz and Philips [14] initially proposed the application of the isoparametric coordinate system for an automatic grid generation technique. Number of computer program based on the isoparametric concepts has been developed by Durocher and Gasper [15], Segerland [16] and Ho [17]. They found that the finite element mesh generation method is most efficient way to generate grid in term of computer speed and memory requirement. Suhara and Fukuda [18] and Bykat [19] has developed algorithm that can also automatically generate the grid for a two-dimensional body. Imafuku and Kodera [20] have presented a mesh generation method for quadrilateral zones. Ecer et al. [21] have developed computational grid around an aircraft using a block-structured finite element grid generation method. It has been mentioned by them that the finite element grid generation is fastest procedure to generate grid in many cases and allows explicit control of grid distribution.

Commercial software are available to generate grid [22] such as Gambit, Pro-grid, Patran, I-deas, Pro/Engineer, Solid Edge and many that can be linked to flow solver CFX, CFD++, Fluent, Star-CD, Flow3D etc.

The above literature survey shows that the finite element grid generation method is found easy to generate the grid and does not need numerical solution of the partial differential equations. Another attraction of the finite element grid generation method is its generality and simplicity in designing a computational grid for an irregular geometry with minimum restriction. In the present paper, the application of the block structured finite element method is implemented with an algebraic homotopic scheme for the generation of quadrilateral grid [23 - 25]. The homotopic method offers a variety of control options over the grid qualities such as smoothness and the grid clustering in the region where are needed. The method is simple, efficient and easily programmed and coupled with grid independent test and pre and post processing with the MATLAB [26].

Finite Element Grid Generation

Quadrilateral Element

The finite element grid generation procedure utilizes a multi-block structured arrangement to divide the prescribed computational domain in several sub-domains. The grid generation is demonstrated in the sub-region of the two dimensional global space (x, y) as shown in Fig.1(a). The finite element consists of curved-sided quadrilateral element. Mapping is one-to-one relationship between local coordinate (ξ, η) as depicted in Fig.1(b). The element consists of eight nodes, and all nodes are located on the boundary. The eight node quadrilateral element belongs to serendipity family of element. It is initially described by a single isoparametric finite element. The isoparametric finite elements are commonly used in finite element analysis in describing irregular geometries. The automatically grid generation procedure requires division of the node points within a region. A quadrilateral element can be written as

$$x = \sum_{l=1}^{8} N_l(\xi, \eta)\, x_l$$

$$y = \sum_{l=1}^{8} N_l(\xi, \eta)\, y_l \qquad (1)$$

where x_l and y_l are the natural coordinates of node l (l = 1, 2,..., 8) and the expression for the shape function N can be written [27] as

Corner Node

$$N_l^c = \frac{1}{4}(1+\xi_l)(1+\eta_l)(\xi_l+\eta_l-1) \qquad l = 1, 3, 5, 7$$

(1)

Middle Side Nose

$$N_l^m = \frac{1}{4}(1+\xi^2)(1-\eta) \qquad l = 2, 6 \qquad (3)$$

$$N_l^m = \frac{1}{4}(1+\xi)(1-\eta^2) \qquad l = 4, 8 \qquad (4)$$

Clustering of Grid

One of the controlling factors for the numerical simulation is the proper grid arrangement to resolve flow field features such as boundary layer, separated flow, shock wave, etc. These body-oriented grids are generated algebraically in conjunction with homotopy scheme [23]. The grid generation method is illustrated for an aero-disc spike attached with the nose-blunted body. The normal coordinate is then described by exponentially structured field points, $x_{i,j}$, $r_{i,j}$ extending outwards up to an outer computational boundary. The grid point i is considered in the axial direction and j in the normal direction. Sufficient grid points are allotted in the aero-disc region of the spike.

A computational space over the spiked body is described here as an example to demonstrate the finite element grid generation method. The normal coordinate is obtained by exponentially structured field points, $x_{i,j}$, $r_{i,j}$ extending outwards up to an outer computational boundary. The stretching of grid points in the normal direction is obtained using the following expression:

$$x_{i,j} = x_{i,0}\left[\frac{\frac{(i-1)\beta}{e^{nr-1}}-1}{e^{\beta}-1}\right] + x_{i,w}\left[1-\frac{\frac{(j-1)\beta}{e^{nr-1}}-1}{e^{\beta}-1}\right]$$

$$r_{i,j} = r_{i,0}\left[\frac{\frac{(i-1)\beta}{e^{nr-1}}-1}{e^{\beta}-1}\right] + r_{i,w}\left[1-\frac{\frac{(j-1)\beta}{e^{nr-1}}-1}{e^{\beta}-1}\right] \qquad (5)$$

where $r_{i,w}$ and $r_{i,0}$ are wall and outer surface points, respectively, β is stretching factor. n_x and n_r are total number of grid points in x and r directions, respectively. These grids are generated in an orderly manner. Grid independence tests can be carried out easily taking into consideration the effect of the computational domain, the stretching factor to control the grid intensity near the wall, and the number of grid points in the axial and normal directions.

Pre-processing of Grid

The following MATLAB's statement can be employed to plot the computation grid in order to access the quality and accuracy of the grid points in the computational domain:

```
for    i = 1:N
          for    j = 1 : M
                    row = (i - 1) * M + j;
                    X (i,j) = f1 (row, 1);
                    Y (i,j) = f1 (row, 2);
     end
end

for    i = 2:N-1
          for    j = 2:M-1
                    row = (i - 1) * M + j;
                    xvec= [x (i-1,j), x (i,j), x (i,j+1)];
                    yvec= [y (i-1,j), y (i,j), y (i,j+1)];
     end
hold on;
     end
```

The above *.m* file can be executed employing MATLAB software in order to visualize the grid.

Examples

Geometry of Flat Disc Spike

The dimensional detail of the forward facing flat-disc spike attached to the blunt body, shown in Fig.2, is an axisymmetric design with a hemi-spherical blunt nose diameter D = 0.04 m. The spike consists of an aero disc part and a cylindrical part. The diameter of the cylinder of the spike is 0.1 *D*, The spike is having a disc of diameter 0.2 *D* attached to a stem of diameter of 0.1 *D*. The spike length is *L* = 0.5 *D*. The coordinate of the surface can be taken from CAD files.

Computational Grid

The outer boundary of the computational domain is varied from 3 to 5 times the maximum diameter *D* of the hemispherical body. The grid stretching factor β in the radial direction is varied from 1.5 to 4.0. The present numerical analysis is carried out on 132 x 52 grid points. The grid stretching factor β is selected as 3.5. The coarse grid helps in reducing the computer time. Sub-division of the computational region is depicted in Fig.3. The dotted line indicates the outer boundary of the computational

domain. The symbol *x* shows the node point on the computational boundary. Each region is having eight nodes; therefore, adjacent regions are having common node points. The dotted line indicates the outer boundary of the computational domain. There are five sub domains, which are marked by I, II, ..., V in the computational region.

The computational grid is generated using the above finite element method in conjunction with the homotopy scheme. The grid is visualized using the above mentioned *.m* file and executed using MATLAB. A close-up view of the computational grid over the spiked blunt body can be seen in Fig.4. It can be seen from the figure that the grid is well structured and clustered in the vicinity of the wall.

Axisymmetric Flow Field Solver

Axisymmetric compressible laminar Navier-Stokes equations are solved using a spatial discretization in the numerical scheme. The Navier-Stokes equation is written in integral form of a finite volume method. The spatial and temporal terms are decoupled using method of lines. The spatial computational domain is divided into a finite number of quadrilateral grids as described the above grid generation method. The cell-centered spatial desretization is augmented with artificial dissipation terms. Thus, the discretized solution to the governing equations results in a set of volume-averaged state variables for mass, momentum, and energy, which are in balance with their area-averaged fluxes (inviscid and viscous) across the cell faces [28]. The finite code constructed in this manner reduces to a central difference scheme and is second-order accurate in space provided that the mesh is smooth enough. The cell-centered spatial discretization scheme is non-dissipative; therefore, artificial dissipation terms [29] are included as a blend of a Laplacian and biharmonic operator in a manner analogous to the second and fourth difference. The artificial dissipation terms was added explicitly to prevent numerical oscillations near the shock waves to damp high-frequency modes. Temporal integration was performed using the three-stage time-stepping scheme of Jameson et al. [28] based on the Runge-Kutta method. The artificial dissipation is evaluated only on first stage and then frozen for the subsequent stages.

The convergence criterion is based on the difference in density values, ρ, at any grid point between two successive iterations, that is, $| \rho^{n+1} - \rho^{n} | \leq 10^{-5}$, where n is the iterative index. A conservative choice of the Courant-friedrichs-Lewy (CFL) number is taken as 1.4 in order to achieve stable numerical solution. The details of this flow field technique are further described in Ref.[30]. The solver uses a time-marching procedure to compute the flow. The flow is defined to be steady because the flow field is converging to a steady state. The steady options use local time stepping, which leads to a faster convergence to the steady-state flow field.

Conditions corresponding to a freestream Mach number 6.0 are given as initial conditions. All the variables were extrapolated at the outer boundary, and the no-slip wall condition was used to implement the boundary conditions. An isothermal condition was prescribed for the surface of the model, that is, a wall temperature of 300 K. The symmetric condition is applied on the centerline.

Flow Field Visualization Over a Flat-disk Aerospike

The above mentioned numerical algorithm is applied to obtain the flow field over the flat-face disk spike attached to the hemispherical blunt body. Velocity vector and contour plots are drawn employing the MATLAB. The following statements used to create *.m* file and executed on the MATLAB to obtain velocity vector.

```
for i = 1:N
    for j = 1:M
        row = (i-1) * M+j;
        X(i,j) = f1 (row, 1);
        Y(i,j) = f1 (row, 2);
        PX(i,j) = f1 (row, 3);
        PY(i,j) = f1 (row, 4);
  end
end

quiver (X, Y, PX, PY, 0.15)
```

Figure 5 depict the velocity vector plots over the flat-disk aerospike for the length to the diameter ratio *L*/*D* of 0.5 at freestream Mach number 6. Characteristic behaviors of the flow field around the spiked blunt body at supersonic speeds are investigated with the help of velocity vector plots. The bow shock wave follows the aerospike contour and the fore-body is entirely subsonic up to the corner tangency point of the flat-faced aerospike. The reflected reattachment wave and the shear layer from the interaction are seen behind the reattachment shock wave. A large separated region is observed in front of the blunt body and the shear layer; and the boundary of the separated region is clearly observed in Fig.5. A strong recirculating flow at the spike can be observed distinctly. The bow

shock wave, the recirculating zone and the reattachment shock are distinctly visible in the vector plot. A large separated region is observed in front of the blunt body and the shear layer; the boundary of the separated region is clearly observed in the velocity vector plots. The following .m file is used to obtain the contour plots:

```
for i = 1:N
    for j = 1:M
        row = (i-1) * M+j;
        X(i,j) = f1 (row, 1);
        Y(i,j) = f1 (row, 2);
        Z(i,j) = f1 (row, 7);
   end
end
[cs,h] = contour (X, Y, Z, 25);
colorbar ('vert')
```

The density contours and corresponding vector plots for the flat-face disk are shown for Mach 6.0. In Fig.6, the interaction between the bow shock wave starting from the aero-disc of the spike and the reattachment shock wave of the blunt body is observed. The reflected reattachment wave and shear layer from the interaction are shown behind the reattachment shock wave. The numerical simulation gives the effects of the subsonic region over the spike. The flow field is characterized by the formation of the bow shock wave in front of the spike, a separated flow region and interaction between the bow shock wave and the reattachment shock wave. The body is completely enveloped within the recirculation region, which is separated from the inviscid flow within the bow shock wave by a separation shock. The bow shock wave interacts with the reattachment shock generated by the blunt body. The cause of the drag reduction is due to increase of the separation region over the aerodisk. The normal shock wave in front of the aerodisk will reduce the drag. In the fore region of the aerodisk, the fluid decelerates through the bow shock wave. At the shoulder of the aerodisk, the flow turns and expands rapidly, the boundary layer detaches, forming a free shear layer that separates the inner recirculating flow region behind the base from the outer flow field. The corner expansion over aerodisk process is a modified Prandtl-Mayer pattern distorted by the presence of the approaching boundary layer.

Computational Grid Over the Heat Shield of Satellite Launch Vehicle

The outer boundary of the computational domain is varied from 3 to 5 times the maximum diameter *D* of the hemispherical body. The grid stretching factor in the radial direction is varied from 1.5 to 4.0. The present numerical analysis is carried out on 132 x 52 grid points. The grid stretching factor is selected as 3.5. The coarse grid helps in reducing the computer time. Sub-division of the computational region is depicted in Fig.7. The dotted line indicates the outer boundary of the computational domain. The symbol *x* shows the node point on the computational boundary. Each region is having eight nodes; therefore, adjacent regions are having common node points. The dotted line indicates the outer boundary of the computational domain. There are five sub-domains in the heat shield as depicted in the figure, which are marked by I, II, ..., V in the computational region.

Quasi-Three-Dimensional Grid and Flow Solver

The above mentioned grid generation method is applied for generating a quasi-three dimensional finite element grid over a typical heat shield of a satellite launch vehicle. The high speed flow around the heat shield is required to study the capability of the present grid generation method. The computational domain is divided into five sub-domains to accommodate the computational zone. The stretching of the mesh near the surface of the heat shield requires resolving the flow features. The finite element grid generation is used to get coordinates along the inner and the outer boundaries of the computational domain. It is important to mention here that the present grid generation is very rapid for generating the grid for different grid stretching factor, far-field boundary location and number of grid points as needed for the validation of grid independent check for the flow solver. The design of the structured block is aimed to provide separate grids for critical flow regions and automate the process of designing or modifying the grid on a block basic. Fig.8 depicts the grid over the heat shield. The grid is drawn using in-house plotting software. The present grid is employed to calculate inviscid surface pressure over the heat shield [1, 31]. Grid convergence has been checked with respect to the vehicle aerodynamics coefficient, the inviscid solution was obtained on 30 x 18 x 45 -cell grids, counting stream wise, circumferentially, and normal to the heat shield. Axial-force coefficient is within 1% on the above grid arrangement.

The numerical simulation is carried over heat shield of satellite launch vehicle by solving the compressible form of three-dimensional Euler equations. The governing fluid dynamic equations are closed by two assumptions of thermodynamics of gas. First, the gas is considered thermally

perfect following the ideal gas law. Second, the gas is calorically perfect. The compressible inviscid equations are numerically simulated with a finite volume solver developed in-house and described [31]. The code solves the compressible fluid dynamics equations in a conservative form on a boundary-fitted multi-block grid. The convective fluxes are solved with a low dissipation upwind scheme and three-stage Runge-Kutta scheme employed for time marching. The details on the implementation of the local artificial damping [29], and the boundary conditions are discussed in detail [1]. At the wall, slip conditions are imposed and at the out flow boundary the two-tangential velocity components are extrapolated from the interior, while at the inflow boundary they are specified at having field values.

Transonic Flow Field Over the Heat Shield

An example of grid generation in conjunction with CFD result is presented for typical heat shield of satellite launch vehicle. Fig.9 depicts the density contour plots at Mach 0.8 and angle of attack 5 deg. it can be visualized from the density contour plots a terminal shock is well captured in leeward and windward side of the flow field. The grid generation can be easily used to generate grid for complex configuration such as protuberance on the satellite launch vehicle [32].

Conclusion

The grid generation algorithm associates with the finite element method and the homotopy scheme has been presented and demonstrated by utilizing a practical configuration. The grid generation procedure needs following three steps:

- Each block of the computational region is defined as eight-nodded quadrilateral isoparametric single element.
- Appropriate grids are defined in each sub-division of the computational domain.
- An algebraic homotopy scheme is employed to cluster the grid in the high gradient region of the flow.

The grid generation method is very simple, flexible and robust. The grid generation procedure is very rapid therefore can be coupled with the flow solver for the grid independency check, influence of the computational region, and stretching of the grid. The present grid generation method maintains geometrical conservation of computational domain. The grid, velocity vector and contour plots can be easily visualized with the help of the MATLAB software.

References

1. Mehta, R. C., "Aerodynamic Design of Payload Fairing of Satellite Launch Vehicle", International Review of Aerospace Engineering, Vol.8, No.3, 2015, pp.167-173.

2. Mehta, R. C., "High Speed Flow Field Analysis for Satellite Launch Vehicle and Reentry Capsule", Journal of Magneto-hydrodynamics, Plasma and Space Research, Vol.15, No.1, 2010, pp.1-49.

3. Mehta, R. C., "Effect of Geometrical Parameters of Reentry Capsule Over Flow Field at High Speed Flow", Advances in Aircraft and Spacecraft Science, Vol.4, No.4, 2017, pp.453-467.

4. Freitas, C. J., "Editorial Policy Statement on the Control of Numerical Accuracy", Journal of Fluid Engineering, Vol.115, 1993, pp.339-340.

5. Thomoson, J. F., Warsi, Z. U. A. and Mastin, C. W., "Numerical Grid Generation - Foundation and Applications", North-Holland, Amsterdam, 1985.

6. Ushakova, O. V., (Editor), "Advances in Grid Generation", Nova Science Publishers, New York, USA, 2007.

7. Thomson, J. F., "Grid Generation Technique in Computational Fluid Dynamics", AIAA Journal, Vol.22, No.11, 1984, pp.1505-1523.

8. Armstrong, C. G., Special issue: "Automatic mesh generation", Advances in Engineering Software, Vol.13, 1991, pp.217-337.

9. George, P. L., "Automatic Generation of Meshes", Wiley, New York, 1991.

10. Piegel, L. G., "On NURBS: A Survey", IEEE Computer Graphics and Applications, Vol.11, No.1, 1991, pp.57-71.

11. Farin, G.L., "NURBS Curves and Surfaces from Projective Geometry to Practical Use", First Edition, A. K. Peters Ltd., 1995.

12. Thomas, P. D. and Middlecoff, J. F., "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations", AIAA Journal, Vol.18, No.6, 1980, pp.652-656.

13. Caruso, S.S., "Development of an Unstructured mesh/Navier-Stokes Method for Aerodynamics of Aircraft with Ice Accretions", AIAA 90-0758, 1990.

14. Zienkiewicz, O. C. and Philips, D. C., "An automatic Mesh Generation Scheme for Plane and Curved Surfaces by Isoparametric Coordinates", International Journal of Numerical Methods in Engineering, Vol.3, 1971, pp.519-528.

15. Durocher, L. L. and Gasper, A. A., "A Versatile Two-dimensional Mesh Generator with Automatic Bandwidth Reduction", Computers and Structures, Vol.10, 1979, pp.561-575.

16. Segerland, L. J., Applied Finite Element Analysis, First Edition, John Wiley and Sons, 1976.

17. Ho, P. T. S., "MSHGEN - A FORTRAN Program on Automatic Two-dimensional and Three-dimensional Triangular Mesh Generation", Advances in Engineering Software, Vol.14, 1992, pp.61-75.

18. Suhara, J. and Fukuda, J., "Automatic Mesh Generation for Finite Element Analysis", Advances in Computational Methods in Structural Mechanics and Design, University of Alabama Press, 1974, pp.607-624.

19. Bykat, A., "Automatic Generation of Triangular Grids, I - Subdivision of a General Polygon into Convex Subregions, II - Triangulation of Convex Polygon", International Journal of Numerical Methods in Engineering, Vol.10, 1976, pp.1329-1342.

20. Imafuku, I. and Kodera, Y., "A Generalized Automatic Mesh Generation Scheme for Finite Element Methods", International Journal of Numerical Methods in Engineering, Vol.15, 1980, pp.713-731.

21. Ecer, A., Spyropoulos, J. and Maul, J. D., "A Three-dimensional, Block-structured Finite Element Grid Generation Schemes", AIAA Journal, Vol.23, 1985, pp.1483-1490.

22. Tu, J., Yeoh, G. H. and Liu, C., "Computational Fluid Dynamics: A Practical Approach", Butterworth-Heinemann, New Delhi, India, 2008.

23. Wilson, H. B., Turcotte, L. H. and Halpern, D., "Advanced Mathematics and Mechanics Applications Using MATLAB", Third Edition, Chapman and Hall/CRC Press, USA, 2003.

24. Shang J. S., "Numerical Simulation of Wing-fuselage Aerodynamic Interference", AIAA Journal, Vol.22, No.10, 1984, pp.1345-1353.

25. Eisemen, P. R. and Smith, R. E., "Mesh Generation Using Algebraic Techniques", NASA CP 2166, 1980, pp.73-120.

26. Moitra, A., "An Algebraic Homotopic Method for Generating Quasi-three-dimensional Grids for High Speed Configurations", NASA CR 4242, 1989.

27. Zienkiewicz, O. C. and Morgan, K., "Finite Elements and Approximation", John Wiley and Sons, Inc., New York, 1982.

28. Peyret, R. and Vivind, H., "Computational Methods for Fluid Flows", Springer-Verlag, Berlin, 1993, pp.109-111.

29. Jameson, A., Schmidt, W. and Turkel, E., "Numerical Simulation of Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes", AIAA Paper 81-1259, 1981.

30. Mehta, R. C., "Pressure Oscillations Over a Spiked Blunt Body at Hypersonic Mach Number", Computational Fluid Dynamics Journal, Vol.9, No.2, 2000, pp.88-95.

31. Mehta, R. C., "Flow Direction Estimation Based on Computed and Flight Measured Surface Pressure", Computational Fluid Dynamics Journal, Vol.12, No.3, 2003, pp.555-561.

32. Mehta, R. C., "Computation of Aerodynamic load on Protuberance Over Satellite Launch Vehicle at Supersonic Speed", Sch. J. of Eng. and Technology, Vol.4, No.7, 2016, pp.301-307.

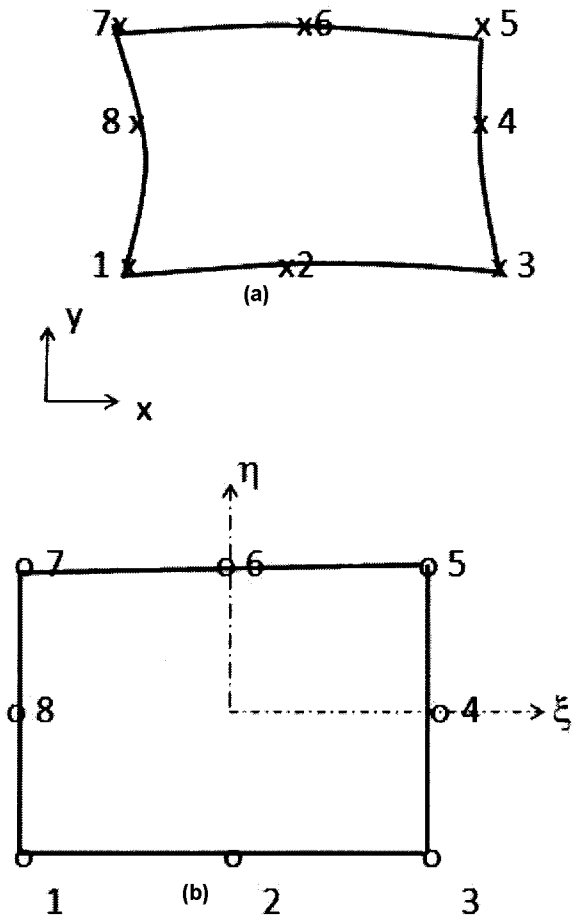

Fig.1 Quadrilateral Element (a) Global Coordinate (b) Natural Coordinate

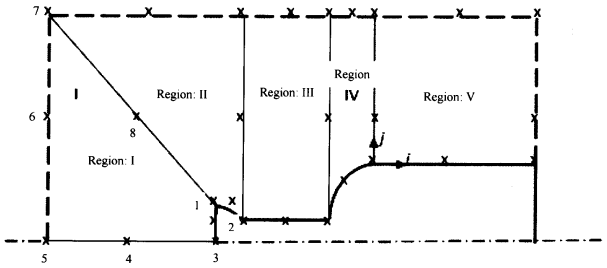

Fig.3 Subdivision of Computational Domain

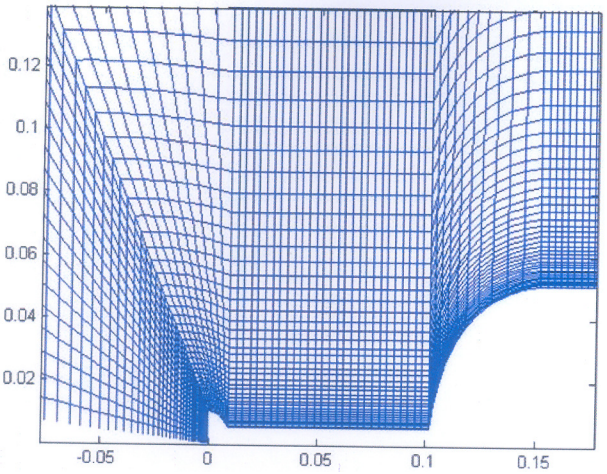

Fig.4 Enlarged View of Grid Over the Spiked Blunt Body

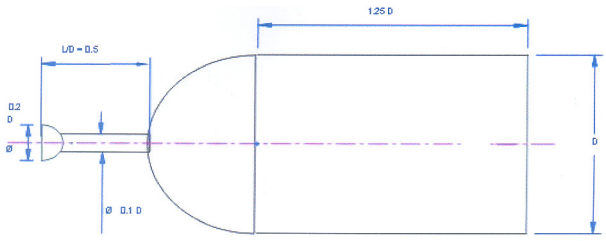

Fig.2 Dimensions of the Spiked Blunt Body

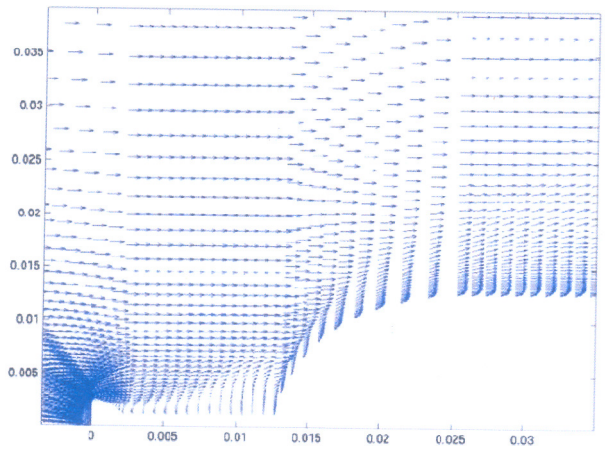

Fig.5 Velocity Vector Plot Over the Spiked Blunt Body

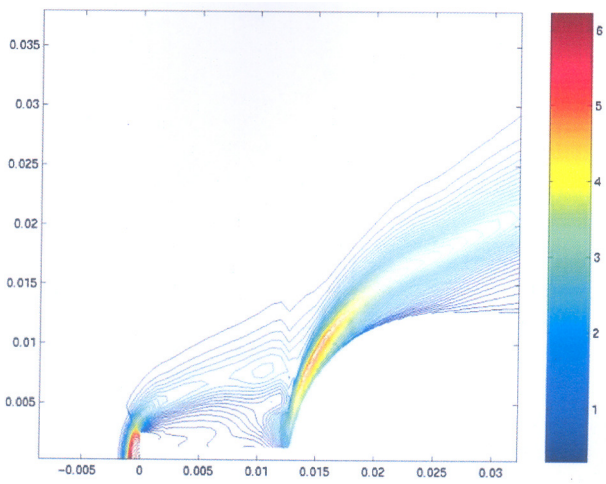

Fig.6 Mach Contour Over the Spiked Blunt Body

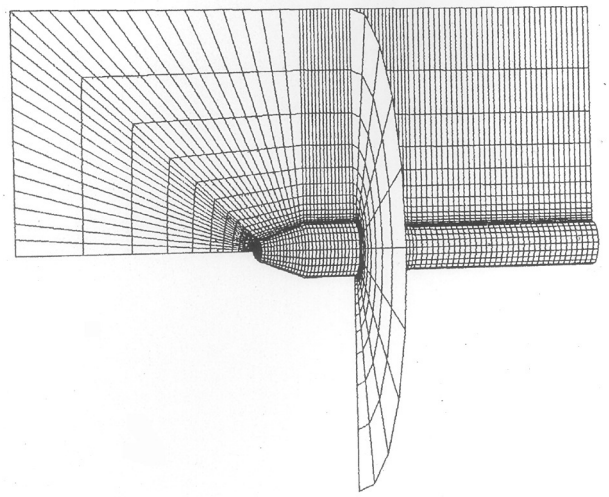

Fig.8 Grid Over the Heat Shield of Satellite Launch Vehicle

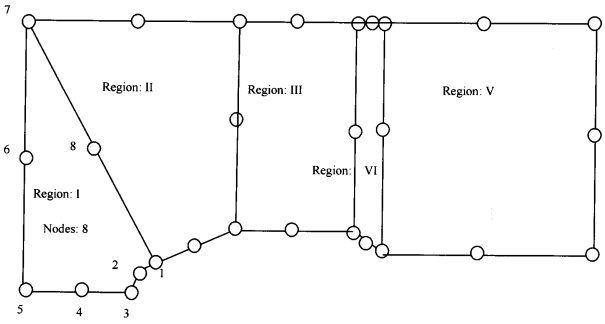

Fig.7 Grid Over the Heat Shield of Satellite Launch Vehicle

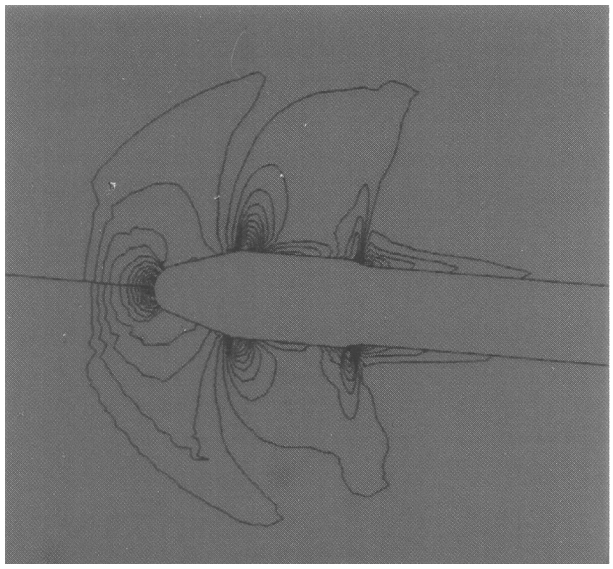

Fig.9 Mach Contour Over the Heat Shield at Mach = 0.8 and Angle of Attack 5 deg