DOI: 10.21276/sjet

Scholars Journal of Engineering and Technology (SJET)
Sch. J. Eng. Tech., 2017; 5(8):387-393
©Scholars Academic and Scientific Publisher
(An International Publisher for Academic and Scientific Resources)
www.saspublisher.com

ISSN 2321-435X (Online)
ISSN 2347-9523 (Print)

Research Article

Multi-Block Structured Grid Generation Method for Computational Fluid Dynamics

R.C Mehta
Department of Aeronautical Engineering, Noolul Islam University, Kumaracoil 629180, India

***Corresponding author**
R.C Mehta
Email: drrakhab.mehta@gmail.com

Abstract: Multi-block structured finite element grid generation algorithm is developed in conjunction with a homotopy technique. The computational domain is divided into a number of quadrilateral regions that takes into consideration topology and coordinates details of each sub-domain. The inner and outer boundaries of the computational domain are described in terms of the surface coordinates. Each sub-domain is designated with an eight nodes isoparametric finite element to generate orderly structured grid and homotopy method is employed to generate grid clustering for numerical flow field simulation. The grid clustering can be controlled by selecting proper stretching factor in order to flow field features such as to resolve boundary layer, to capture shock wave and to locate high pressure gradient region. The grid generation is in two-dimensional plane that can be rotated to obtain a three-dimensional grid arrangement over axisymmetric configuration. Major features of the present grid generation technique are convenient for checking the grid independence test of flow simulation with varying stretching factor. Major attraction of the present algorithm maintains geometrical conservation of computational domain method. The grid plots can be visualized with the aid of MATLAB. The grids can easily link with user developed flow solver or it can also be coupled with commercial flow solver.

Keywords: finite element method, grid generation, computational fluid dynamics.

INTRODUCTION

Computational fluid dynamics is having significant contribution in design and development [1] of a satellite launch vehicle attributed to advancement of computer technologies, numerical schemes, graphics software and interface connecting software. Computational fluid dynamics can simulate complex flow field for example re-circulation flow region caused by shock wave boundary layer interaction or geometrically induced flow separation [2, 3]. A computational fluid dynamics application needs grid generation, flow solver and flow visualization software. Numerical simulation of fluid dynamics equations requires a spatial discretization of the physical domain of interest such as here satellite launch vehicle. A suitable grid must be generated before initiating flow solver for a given configuration. Moreover, it becomes mandatory to have grid independence [4] check before applying in aerodynamic analysis. The desirable grid should possess qualities adequate adoption of surface geometry, fine grid resolution for viscous flow and smooth variation of the grid spacing between fine and coarse grid regions.

Many methods for generating computational grids [5] have been developed in the area of computational fluid dynamics, which are based on either solution of partial differential equations or algebraic techniques. These grid generation procedures compete with each other in their relative abilities to obtain improved grid qualities such as smoothness of point distribution, clustering of grid points of regions of physical interest, maintain the conservation of computational space and ability to accommodate complex configuration.

The finite element method grid generation scheme 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 scheme is its generality and simplicity in generating grid for an irregular (complex) body with minimum restriction. In addition to that the algebraic grid generation is an efficient means of grid generation in terms of computation speed and memory requirements. The algebraic grid generation method relies on transfinite interpolation procedures to obtain grid when the boundaries of the grid have been specified with a user-described distribution. A hole-re-meshing grid generation method [6] allows efficient grid generation about aerofoil with time dependent leading edge ice accretions.

Zienkiewicz and Philips [7] initially proposed the application of the isoparametric coordinate system for an automatic grid generation method. Number of algorithm developed based on the isoparametric concepts [8]. Suhara and Fukuda [9] have developed algorithm that can also automatically generate the grid for a two-dimensional configuration. Ecer *et al.* [10] have developed computational grids around a complex aircraft configuration employing a block-structured finite element grid generation scheme. It has been shown by them that the finite element grid generation method is the fastest procedure to generate grids in many aerospace application and permits explicit control of the grid distribution. Ho [11] has presented FORTRAN program to generate grid two- and three-dimensional grid over based on finite element method [7].

An algorithm for generating 2-dimensional Delaunay triangular unstructured has been presented by Sloan et al. [12] with a computer program. Peraire *et al.* [13] have developed grid generation using advancing front method and generated mesh over aircraft. These unstructured grids show some difficulties to resolve flow field in vicinity of wall. Cartesian grids are used for generation by Haselbacher [14] over airfoil. The grid properties and qualities for structured and unstructured are discussed by Hirsch [15] and Versteeg and Malalasekra [16]. Commercial software are available to generate grid [17] 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.

In this paper, multi-block-structured finite element method is employed with the homotopic technique to generate grid for flow simulations over satellite launch vehicle. The flow field quality and the grid independence check can easily implemented using the stretching factor. The grid generation software is simple to program and output of the program can link with user software or can import to commercial software. The structured grid can adopt geometry of configuration, and also satisfy requirement on smoothness and continuity of cell sizes. The grid can be visualized using MATLAB software in order to assess the quality of the grid.

FINITE ELEMENT GRID GENERATION METHOD

The finite element grid procedure utilizes a multiple-block structured method to divide the computational domain in number of zones. The grid generation is carried out in two steps. Each of these blocks is considered as a super element, which is initially described by a single isoparametric finite element [7]. The position of a plane inside an element can be described in terms of eight nodes quadrilateral as

$$\begin{bmatrix} x \\ y \end{bmatrix} = N_i(\xi,\eta)\begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{1}$$

where N_i is a shape function of isoparametric element [7] associated with each node, *i* is defined in terms of a curvilinear coordinate system ξ, η which have values ranging from 1 to −1 on opposite sides. The *x*- and *y*-coordinates of a typical point is related to the eight pairs of nodal coordinates as

$$x = \sum_{i=1}^{8} N_i(\xi,\eta)x_i \tag{2}$$

$$y = \sum_{i=1}^{8} N_i(\xi,\eta)y_i \tag{3}$$

The isoparametric or sub-region is defined by 8-sets of nodal coordinates, and then relatively simple programming logic is used to generate a grid. The shape function for a quadrilateral super element has been tabulated by Zienkiewicz *et al.* [7]. Thus, if the nodal coordinates (x_i, y_i) are known as the Cartesian coordinates of any specified (ξ, η) can be easily evaluated from Eq. (1).

HOMOTOPY METHOD

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. the body oriented grids can be easily generated algebraically in conjunction with homotopy scheme. The grid generation method is illustrated for heal shield of satellite launch vehicle. The normal coordinate is then described by exponential stretched field points (x_{ij}, r_{ij}) extending outward 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 heat shield region of the satellite launch vehicle.

The typical computational space over the heat shield is defined by a number of grid points in cylindrical coordinate system. Using these surface structured field points as the reference nodes, the normal coordinate is the described. The clustering of the grid inside the individual block can be carried out using following expression

$$x_{i,j} = x_{i,0}\left[\frac{e^{\frac{(i-1)\beta}{nr-1}}-1}{e^{\beta}-1}\right] + x_{i,w}\left[1-\frac{e^{\frac{(j-1)\beta}{nr-1}}-1}{e^{\beta}-1}\right] \qquad (4a)$$

$$r_{i,j} = r_{i,0}\left[\frac{e^{\frac{(i-1)\beta}{nr-1}}-1}{e^{\beta}-1}\right] + r_{i,w}\left[1-\frac{e^{\frac{(j-1)\beta}{nr-1}}-1}{e^{\beta}-1}\right] \qquad (4b)$$

where subscripts *o* and *w* are outer and wall grid points, respectively. β is stretching factor and *nr* is total number of grid points in *r* direction. The grids are generated in an orderly manner. Grid independent tests can be carried out taking into consideration the effect of the computational domain, the stretching factor to control the grid concentration in the vicinity of wall, and the number of grid points in the axial and normal directions. 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
```

GRID GENERATION PROCEDURE

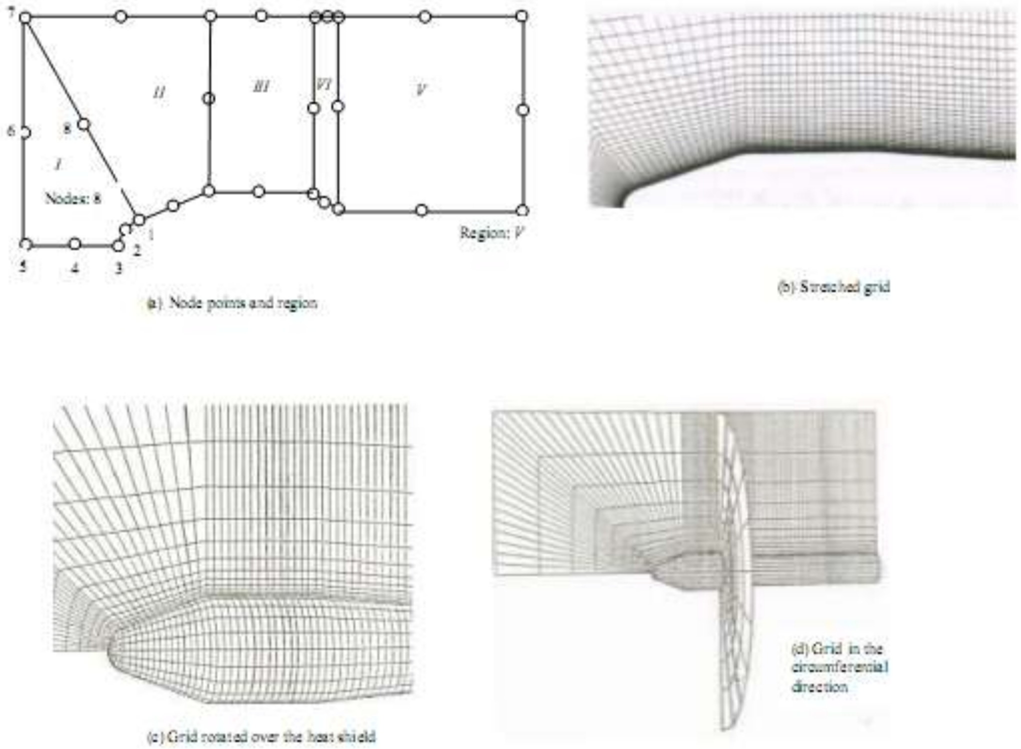
The above mentioned grid generation method is applied to obtain quasi-three dimensional grid over a heat shield of a typical satellite launch vehicle. The computational domain is divided into a number of blocks. The stretching of the grid in the vicinity of the heat shield of the launch vehicle is required to resolve the viscous flow [2]. Grid concentration is also needed to capture flow discontinuity. The finite element grid generation is used to obtain coordinates along the inner and outer boundaries of the computational domain. It is important to say here that the present grid generation procedure is very rapid for grid generation with stretching factor. The blocks are aimed to provide separate grid depending upon the flow gradients.

EXAMPLES

The grid generation algorithm needs following steps: define a computational domain, divide the computational domain into number of zones. Input to grid generation program is coordinates of 8-node quadrilateral finite element and coordinate of each zone. Number of examples such as heat shield without and with forward facing spike, and high expansion rocket nozzle is presented here in order to describe steps: computational domains, number of blocks, inner and outer boundaries and eight node isoparametric super element.

Heat shield of a satellite launch vehicle

Figure 1 (a) depicts typical heat shield of a satellite launch vehicle. The heat shield of a satellite launch vehicle is having axisymmetric configuration. The heat shield is divided into 5 zones as shown in Fig. 1 (a). Each zone is having eight nodes. Small circle on the isoparameric element represents the node points and *x* and *y* coordinates. They are input to the generate grid program.



(a) Node points and region

(b) Stretched grid

(c) Grid rotated over the heat shield

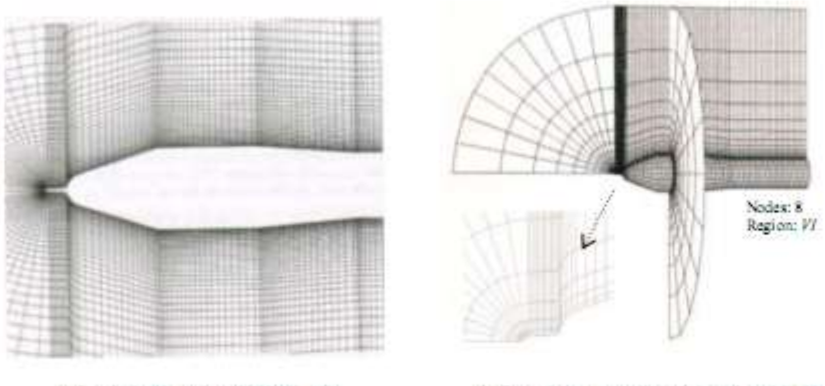
(d) Grid in the circumferential direction

Fig-1: Grid generation over heat shield of satellite launch vehicle

In each block one can select the mesh points depending on prior assumption of flow gradient. If the prior assumption is not available then initial grid is generated and subsequently revised based on the results of flow field simulation. The number of grid points in each zone can be prescribed and their summations lead to total number of mesh in *x*-direction, that is, *nx*. Equation (4) is used to stretch the mesh by choosing appropriate stretching factor β (may base on Reynolds number). The grids are stretched in the radial direction. The number grid points in the entire zone are kept *nr*. Figure 1 (b) displays axisymmetric grid over the heat shield. The three-dimensional grid can be obtained by revolving the grid along the centre line of the model. Figure 1(c) and (d) depicts three-dimensional mesh over the heat shield of typical satellite launch vehicle generated by revolving the mesh in the circumferential direction.

A forward facing spike attached to a Heat shield

Figure 2 (a) display grid generation for a spike attached to heat shield of a satellite launch vehicle. In this case, there are 6 blocks and each block is having eight node points. Figure 2 (b) depicts the three-dimensional view of grid over the spike attached to the heat shield. An enlarge view is also depicted to show the grids in the vicinity of spike. The grids are structured well and having continuity in the radial direction.



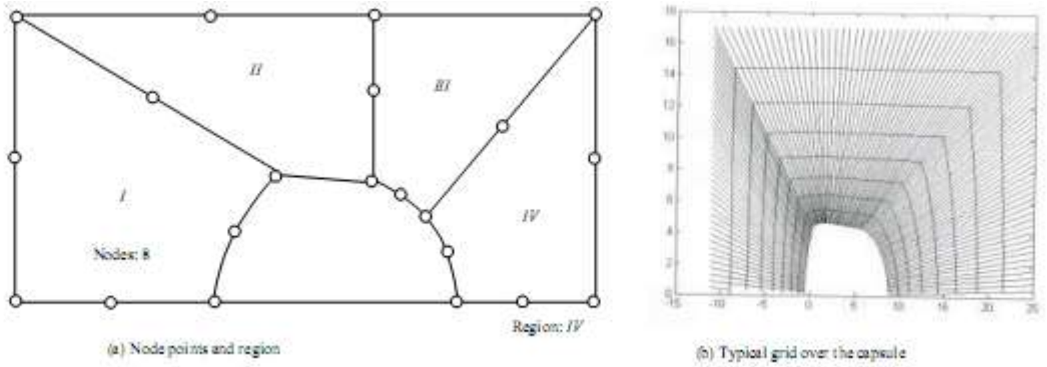
(a) Stretched grid over heat shield with spike

(b) Zoomed view over spike and circumferential grid

Fig-2: Grid generation over forward facing spike attached to heat shield

Reentry capsule

We have taken here a reentry capsule identical to a Soyuz. The reentry computation domain is divided in four zones and each zone is having 8 node points for isoparametric element as shown in Fig. 3(a). The grid arrangement and is obtain orderly manner and well structured. Figure 3(b) depicts the mesh in the computational domain of the typical capsule. The grids are exhibited with the help of MATLAB software.



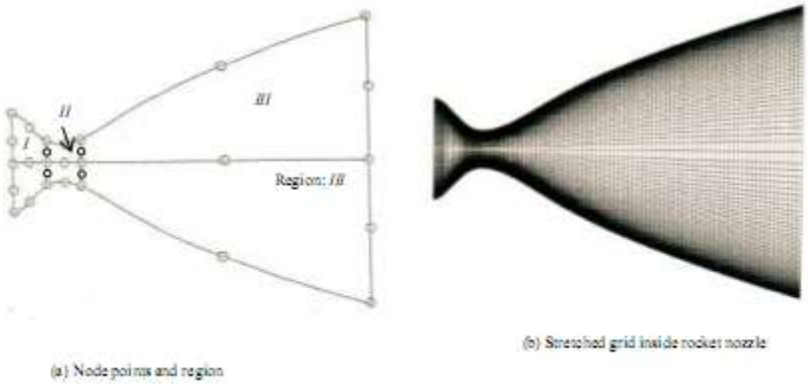
(a) Node points and region

(b) Typical grid over the capsule

Fig-3: Grid generation over re-entry capsule

High expansion rocket nozzle

The grid is generated for interflow numerical flow simulation. The computation domain is shown in Fig. 4 (a) and is showing 3 regions and each region is contained 8 nodes. Figure 4 (b) displays the highly stretched grid in the vicinity of nozzle wall. The grid is having capability to generate suitable mesh for high expansion rocket nozzle.



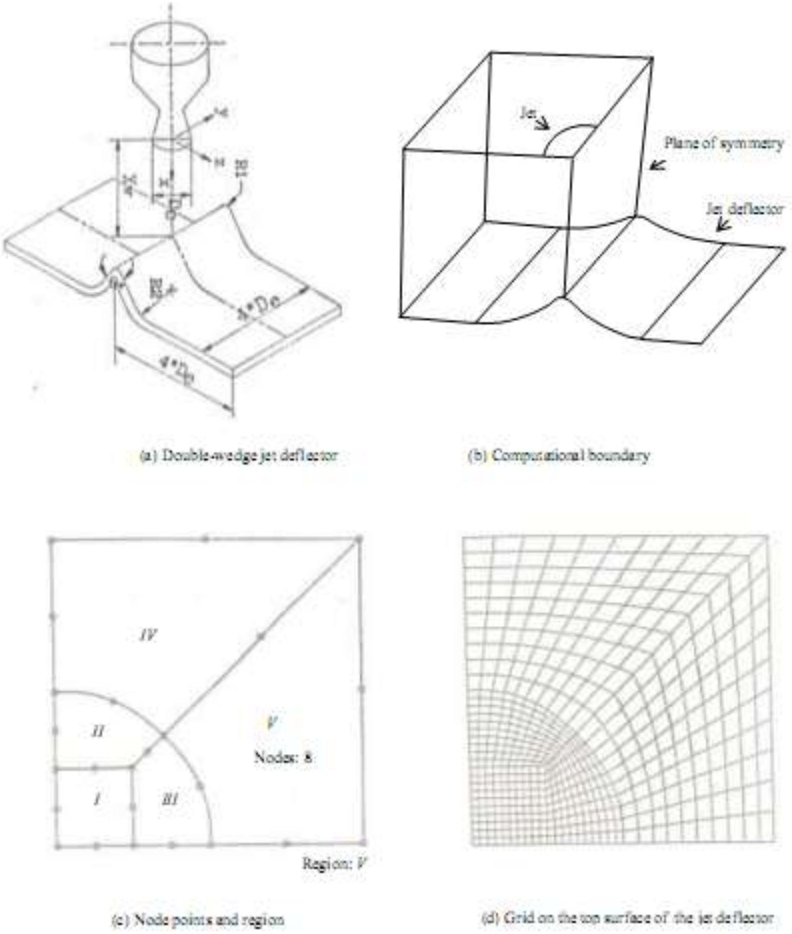
(a) Node points and region

(b) Stretched grid inside rocket nozzle

Fig-4:Grid generation in bell shape rocket nozzle

Double-wedge jet deflector

The grid generation program is applied to generate mesh over a double-wedge jet deflector employed to divert the supersonic exhaust of a satellite launch vehicle as depicted in Fig. 5 (a) and (b) with a line diagram to illustrate the plane of symmetry. Figure 5 (c) shows node isoparametric eight nodes on element points and having five zones. A structured grid over the top plane of the jet deflector is shown in Fig. (d) and can stack to obtain three-dimensional grid on the supersonic jet deflector.



(a) Double-wedge jet deflector

(b) Computational boundary

(c) Node points and region

(d) Grid on the top surface of the jet deflector

Fig-5: Double-wedge jet deflector

CONCLUSIONS

The finite element grid generation in conjunction with homotopy method is computationally efficient, reasonably simple to implement and to possess a good degree of generality. As it is always desirable to produce computer plots to grids such an adjustment if confined to a few points is easily made. MATLAB can be used to visualize the computed mesh. The programme has been extended to allow for the multi-connected zones so that complex geometries such as blunt body attached with a forward facing spike and high expansion rocket nozzle and mesh over wedge supersonic jet deflector within its compass. The scheme can easily be extended to three-dimensional grid generation. Number of examples is worked out to demonstrate the capability of the present algorithm. The computational mesh can easily link with commercially available flow solver.

REFERENCES

1. Raj P. Requirements for effective use of CFD in aerospace design, surface modeling, grid generation, and related issues in computational fluid dynamics solutions, NASA CP 3291, May 1995.
2. Mehta RC. “High speed flow field analysis for satellite launch vehicle and re-entry capsule. Journal of Magneto-hydrodynamics, Plasma and Space Research. 2008;15(1): 51-99.
3. Mehta RC. “Numerical simulation of supersonic flow past reentry capsules. Shock Waves2006; 15(1): 31-41.
4. Freitas CJ. Editorial policy statement on the control of numerical accuracy. ASME. Journal of Fluid Engineering. 1993; 115: 339-340.
5. Thomson JF. Grid generation techniques in computational fluid dynamics. AIAA Journal. 1984; 22(11): 1505-1523.
6. Caruso S. Development of an unstructured mesh/ Navier-Stokes method for aerodynamics of aircraft with ice accretion, AIAA 90-0758, Jan 1990.
7. Zienkiewicz OC, Philips DC. An automatic mesh generation scheme for plane and curved surfaces by isoparametric coordinates. International Journal of Numerical Method in Engineering. 1971; 3: 519-528.
8. Segerland JJ. Applied Finite Element Analysis. First Edition, John Wiley & Sons, 1976.
9. Suhara J, Fukuda J. Automatic mesh generation for finite element analysis. Advances in Computational

Methods in Structural Mechanics and Design, University of Alabama Press, 1974; 607-624.

10. Ecer A, Spyropoulos J, Maul JD. A three dimensional block structured finite element grid generation schemes. AIAA Journal. 1985;23: 1483-1490.
11. Ho PTS. MSHGEN- a FORTRAN program on automatic two-dimensional and three-dimensional grid generation. Advances Engineering Software. 1992;14(1): 61-75.
12. Sloan SW. An implementation of Watson's algorithm for computing 2-dimensional Delaunay triangulations. Advances Engineering Software 1984; 6(4): 192-198.
13. Peraire J, Morgan K, Peiro. Unstructured finite element mesh generation and adaptive procedures for CFD, paper at the AGARD FDP: Specialist's Meeting 24-15 May 1989, Leon, Norway.
14. Haselbacher AC. A grid-transparent numerical method for compressible viscous flows on mixed unstructured grids, doctoral thesis, Loughborough University of Technology, UK, 1999.
15. Hirsch C. Numerical Computation of Internal & External Flows, Butterworth-Heinemann, New Delhi, India, 2nd Edition, 2012.
16. Versteeg H, Malalasekra W. An Introduction to Computational Fluid Dynamics, Pearson Education Ltd, New Delhi, India, 2007.
17. Tu J, Yeoh GH, Liu C. Computational Fluid Dynamics: A Practical Approach, Butterworth-Heinemann, New Delhi, India, 2008.