To the Graduate Council:

I am submitting herewith a thesis written by Jacob Chackasseril Varghese entitled “Sonic Boom Prediction Methods Using Feature-Based Adaptation of Unstructured Meshes.” I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computational Engineering.

___________________________
Steve L. Karman
Dr. Steve L. Karman, Jr.
Professor, Computational Engineering,
Chairperson/Principal Advisor

We have read this thesis and recommend its acceptance:

_______________________________
Timothy W. Swafford
Dr. Timothy W. Swafford
Professor and Head, Graduate School of Computational Engineering

_______________________________
Daniel G. Hyams
Dr. Daniel G. Hyams
Associate Professor, Computational Engineering

Acceptance for the Council:

___________________________
Stephanie L. Bellar
Stephanie L. Bellar
Interim Dean of the Graduate School
Sonic Boom Prediction Methods Using Feature-Based Adaptation of Unstructured Meshes

A Thesis

Presented for the

Master of Science Degree

The University of Tennessee at Chattanooga

Jacob Chackasseril Varghese

December 2009
DEDICATION

This thesis is dedicated to my parents, C. C Varghese and Thankamma Varghese, and my brothers, Jeby Kurian Varghese and Iby Mani Varghese and the rest of the family, for their unconditional love and support.
ACKNOWLEDGEMENTS

I would like to thank Dr. Steve Karman, my advisor, for his guidance throughout the process and his effort in making me familiar with the concept of grid generation. I would also like to thank the other members of my committee, Dr. Tim Swafford and Dr. Daniel Hyams, for taking the time to serve in this capacity and all my friends at the SimCenter for their help.

This research was sponsored partially by NASA Langley Research Center (Contract No. NASA NRA NNL07AA28C). This support is greatly appreciated.
ABSTRACT

This study examines the improvement of near-field sonic boom prediction of an inviscid supersonic configuration using two grid generation refinement procedures. The first method uses P_HUGG, a parallel hierarchical Cartesian mesh generation algorithm to generate a volume mesh, with the solution-based mesh adaptation capability of P_HUGG being exploited. The mesh quality was improved using P_OPT, a parallel optimization-based mesh-smoothing program. In the second method, the commercially-available software POINTWISE™ is used for volume mesh generation. Then, P_REFINE, a parallel subdivision refinement code, is used to adaptively refine the mesh. The effectiveness of capturing far field shocks was examined using TENASI, an unstructured flow solver developed at the SimCenter at the University of Tennessee at Chattanooga. The grids are adapted to high pressure gradient using SPACING, a program that computes the desired spacing at all points in the mesh. Results from both methods are compared with wind-tunnel based experimental data.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I  INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II GEOMETRY</td>
<td>4</td>
</tr>
<tr>
<td>III ANALYSIS TOOLS</td>
<td>6</td>
</tr>
<tr>
<td>III. A P_HUGG (Parallel Hierarchical Unstructured Grid Generation)</td>
<td>6</td>
</tr>
<tr>
<td>III. B P_OPT (Parallel Optimization-Based Smoothing)</td>
<td>8</td>
</tr>
<tr>
<td>III. C TENASI (Unstructured Flow Solver)</td>
<td>9</td>
</tr>
<tr>
<td>III. D SPACING (Feature-Based Adaptation)</td>
<td>11</td>
</tr>
<tr>
<td>III. E P_REFINE (Parallel Unstructured Subdivision Refinement)</td>
<td>14</td>
</tr>
<tr>
<td>IV TECHNIQUES AND STRATEGY</td>
<td>15</td>
</tr>
<tr>
<td>IV.A HYBRID GRID Method</td>
<td>16</td>
</tr>
<tr>
<td>IV.B TETRAHEDRAL GRID Method</td>
<td>21</td>
</tr>
<tr>
<td>V RESULTS</td>
<td>23</td>
</tr>
<tr>
<td>V.A Grid Generation Using HYBRID GRID</td>
<td>23</td>
</tr>
<tr>
<td>V.B Grid Generation Using TETRAHEDRAL GRID</td>
<td>24</td>
</tr>
<tr>
<td>VI SUMMARY AND CONCLUSIONS</td>
<td>36</td>
</tr>
<tr>
<td>FUTURE WORK</td>
<td>38</td>
</tr>
<tr>
<td>LIST OF REFERENCES</td>
<td>39</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>42</td>
</tr>
<tr>
<td>1. Boundary Conditions Used by the Flow Solver</td>
<td>43</td>
</tr>
<tr>
<td>2. Parameter File Used by the Flow Solver</td>
<td>44</td>
</tr>
<tr>
<td>VITA</td>
<td>46</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>A n F-18 Breaks the Sound Barrier</td>
<td>1</td>
</tr>
<tr>
<td>2.</td>
<td>Schematic of Sonic Boom Generation</td>
<td>2</td>
</tr>
<tr>
<td>3.</td>
<td>Sketch of the Wind-tunnel Apparatus</td>
<td>4</td>
</tr>
<tr>
<td>4.</td>
<td>Delta Wing-body Geometry</td>
<td>5</td>
</tr>
<tr>
<td>5.</td>
<td>Perspective view of computational domain</td>
<td>5</td>
</tr>
<tr>
<td>6.</td>
<td>Refinement Modes for Hexahedral-cells</td>
<td>6</td>
</tr>
<tr>
<td>7.</td>
<td>Symmetry-Plane Grid with Polyhedral Elements</td>
<td>7</td>
</tr>
<tr>
<td>8.</td>
<td>Symmetry-plane Grid without Polyhedral Elements</td>
<td>7</td>
</tr>
<tr>
<td>9.</td>
<td>Spacing Field with x-spacing Contours Plotted</td>
<td>13</td>
</tr>
<tr>
<td>10.</td>
<td>Spacing-rbox Setup for TETRAHEDRAL GRID Method</td>
<td>13</td>
</tr>
<tr>
<td>11.</td>
<td>Spacing-rbox Setup for HYBRID GRID Method</td>
<td>13</td>
</tr>
<tr>
<td>12.</td>
<td>Spacing Field with rbox defined</td>
<td>13</td>
</tr>
<tr>
<td>13.</td>
<td>Initial Surface Mesh of the Model with an Integrated Sting</td>
<td>15</td>
</tr>
<tr>
<td>15.</td>
<td>Surface Mesh Generated by P_HUGG using general cutting</td>
<td>17</td>
</tr>
<tr>
<td>16.</td>
<td>First Adapted Meshes Generated by P_HUGG</td>
<td>19</td>
</tr>
<tr>
<td>17.</td>
<td>Surface Mesh Generated by POINTWISE™</td>
<td>21</td>
</tr>
<tr>
<td>18.</td>
<td>Process Diagram: TETRAHEDRAL GRID Method</td>
<td>22</td>
</tr>
<tr>
<td>Page</td>
<td>Image Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
<td></td>
</tr>
<tr>
<td>19.</td>
<td>A Perspective View of Initial P_HUGG Mesh with Symmetry Plane</td>
<td></td>
</tr>
<tr>
<td>20.</td>
<td>Convergence Plots for HYBRID GRID Method</td>
<td></td>
</tr>
<tr>
<td>21.</td>
<td>Initial Mesh with Pressure Contours on Symmetry Plane</td>
<td></td>
</tr>
<tr>
<td>22.</td>
<td>Final Mesh with Pressure Contours on Symmetry Plane</td>
<td></td>
</tr>
<tr>
<td>23.</td>
<td>Spacing Field with x-Spacing Contours on Symmetry Plane</td>
<td></td>
</tr>
<tr>
<td>24.</td>
<td>Final Mesh with Crinkled Symmetry Plane</td>
<td></td>
</tr>
<tr>
<td>25.</td>
<td>Pressure Signals Plotted Against Experimental Data</td>
<td></td>
</tr>
<tr>
<td>26.</td>
<td>Pressure Adapted Grids Generated by P_HUGG with Spacing-box</td>
<td></td>
</tr>
<tr>
<td>27.</td>
<td>Perspective View of Initial TETRAHEDRAL GRID Mesh with Crinkled Symmetry Plane; 816K Cells</td>
<td></td>
</tr>
<tr>
<td>28.</td>
<td>Convergence Plot for TETRAHEDRAL GRID Method</td>
<td></td>
</tr>
<tr>
<td>29.</td>
<td>Initial Mesh with Pressure Contours on Symmetry Plane</td>
<td></td>
</tr>
<tr>
<td>30.</td>
<td>Spacing Field with x-spacing Contour on symmetry Plane</td>
<td></td>
</tr>
<tr>
<td>31.</td>
<td>Final Mesh with Crinkled Symmetry Plane</td>
<td></td>
</tr>
<tr>
<td>32.</td>
<td>Final Mesh with Pressure Contour on y=0 Plane</td>
<td></td>
</tr>
<tr>
<td>33.</td>
<td>Pressure Signals Plotted Against Experimental Data</td>
<td></td>
</tr>
<tr>
<td>34.</td>
<td>Pressure Adapted Grids Generated by P_REFINE Using Spacing-box</td>
<td></td>
</tr>
<tr>
<td>35.</td>
<td>Sketch of a Shock Passing Through a Cartesian Cell</td>
<td></td>
</tr>
<tr>
<td>36.</td>
<td>Sketch of a Shock Passing Through a Tetrahedral Cell</td>
<td></td>
</tr>
</tbody>
</table>
37. Residual Plot for No-Limiter Case ................................................................. 35
38. Pressure Distribution Comparison for No-Limiter Case ..................................... 35
I. INTRODUCTION

Man has been able to beat the speed of sound and break the sound barrier thus enabling supersonic travel. The incredible thing about breaking the sound barrier is that not only can it produce an audible effect but sometimes a visible one (see Figure 1). Because of sonic boom intensity, the Federal Aviation Administration prohibits supersonic flight over land, except in special military flight corridors. Numerous efforts are underway to identify and mature technologies to reduce the sonic boom over-pressure [1], which could eventually enable unrestricted supersonic flight over land by future military and business aircraft. The Figure 1 shows an F/A-18 Hornet cruising at transonic speed. As the aircraft accelerates to just break the sound barrier, however, an unusual cloud might form. The origin of this cloud is still debated. A leading theory is that a drop in air pressure at the plane occurs so that moist air condenses there to form water droplets. The flow inside the cloud is locally supersonic.

A typical airplane traveling at a Mach number greater than one generates two main shock waves; one at the nose and one off the tail (see Figure 2). Shock waves coming off the canopy, wing leading edges, engine nacelles, etc. tend to merge with the main shocks some distance from the airplane. The resulting pressure pulse appears to be N-shaped. To an observer on the ground, this pulse is felt as an abrupt compression above atmospheric pressure followed by a rapid decompression below atmospheric pressure and a final recompression to atmospheric pressure. The strongest sonic boom is felt directly beneath the airplane and decreases to nothing on either side of the flight path.

Figure 1. An F-18 Brakes the Sound Barrier

Source: NASA
The prediction of sonic boom intensity is often accomplished by first obtaining a near-to-mid field (<10 body lengths from aircraft) pressure distribution, either by measuring it in a wind tunnel or through the use of computational fluid dynamics (CFD). From this near-field signature, the sonic boom at the ground can be rapidly extrapolated using propagation methods [2] that include various atmospheric models.

A problem that arises when using CFD method is that the shocks that causes sonic boom are “smeared” or diffused proportional to the distance from the aircraft because of coarser grids in these regions. The approach to address this problem is through the use of grid adaptation. Some of the grid adaptation methods include the use of Adjoint-based adaptation procedures [3], in which the Cartesian grid is rotated to align mesh cells with the free stream Mach-wave angle, and the use of feature-based adaptation methods on unstructured meshes [4]. In some cases the initial grid is clustered and oriented based on general flow characteristics that are known in advance. Typically, a grid adaptation procedure is used to reduce the computational cost while maintaining a given level of accuracy for the solution. Feature based adaptation methods try to resolve local features by increasing the node density in regions of high gradients at the expense of regions with comparatively low gradients.

This work examines the use of grid adaptation to capture near field shocks using two grid generation approaches. The first method automatically generates a Cartesian aligned hybrid mesh by using a user-defined spacing or/and by using spacing field information from a feature-based
adaptation process. In the second method, a fully tetrahedral mesh is generated using POINTWISE™ and this mesh is refined using spacing field information from a feature-based adaptation process. To compare the results of the study, a supersonic wind-tunnel configuration that includes a delta wing-body model was selected due to the availability of experimental data.

The thesis begins with an overview of the wind tunnel experimental setup and the wing-body geometry under consideration. Thereafter, a description is given of the grid generation and feature-based adaptation algorithms, namely P_HUGG, P_OPT, SPACING and P_REFINE. These algorithms are then used to obtain meshes better suited for the computations of flows with shocks using a numerical flow solver, TENASI. All of the above mentioned grid generation algorithms and the flow solver have been developed by researchers at the SimCenter: National Center for Computational Engineering at the University of Tennessee at Chattanooga. Finally, the two adaptation procedures are described and the results from the study are tabulated.

It should be noted that similar adaptive refinement studies were conducted by Beasley [5] using 2-dimensional geometries. In his work, HUGG2D, a 2-dimensional, serial version of P_HUGG was employed to generate an adapted unstructured mesh. This thesis extends his work to three dimensions using the latest meshing tools of the SimCenter.
II. GEOMETRY

A wind tunnel model designed to study sonic boom was used in a 1973 wind tunnel experiment [6]. The experiment was conducted in a 9-by 7-foot supersonic wind tunnel at the NASA Ames Research Center. A schematic of the wind tunnel apparatus is shown in Figure 3.

Two pressure probes fixed on the wind tunnel wall were used to measure the near-field signature. Four pressure orifices were located around the circumferences of each probe. The orifices of the reference probe were in a plane normal to the free stream while those for the overpressure probe were oriented in a plane parallel to the Mach angle. This configuration produces a sharper signature [6].

Near field pressure signature data were measured on the vertical plane of symmetry, at a distance 3.6 times the body length above the geometry, with zero lift condition and at Mach number 1.68. Dashed line in Figure 3 represents the computational domain selected for this study.

Figure 3. Sketch of the wind-tunnel Apparatus

Source: Reference 6
The present study considers the “Model 4” geometry described in reference 6. This model consists of a cylindrical shaped fuselage with 69-degree leading edge sweep delta wing. Detailed model measurements are shown in Figure 4. The dimensions are in centimeters. The airfoil has a symmetric, 5% thick diamond section.

While modeling the geometry there is an approximate sting attached to the tail end. This attachment is necessary because it was present in the experiment. However, the sting geometry used herein is an approximation to that used in [6]. Fortunately, this approximation should not impact the pressure-field where the measurements were made due to the Mach number and wave angle signal propagation. The model is symmetric to both y-plane and z-plane.

A perspective view of the computational domain is as shown in Figure 5. The wing-body is placed such that the nose of the geometry is at the origin. A cutting plane (light grey) which is also the y-symmetry plane is also shown in the figure. The top boundary (not shown in the figure) is at a distance ratio of 3.6 times the body length above the geometry, where the wind-tunnel experimental data is measured. This study computes the pressure distribution along the intersection (black line on green surface in Figure 5) of y=0 symmetry plane and bottom boundary, which is at 3.6 times the body length below the model. Since the geometry is symmetric and the angle of attack used is zero degree, both top and bottom boundary should give same pressure distributions.
III. ANALYSIS TOOLS

III.A. P_HUGG (Parallel Hierarchical Unstructured Grid Generation)

![Refinement Modes for Hexahedral cells](Figure 6. Refinement Modes for Hexahedral cells)

Source: Reference 8

P_HUGG is a robust parallel mesh generation procedure [7] that uses subdivision refinement of a Cartesian aligned mesh with general cutting to produce an inviscid volume mesh. The mesh is automatically generated around any complicated geometry with minimal user interventions.

Initially a hexahedral-shaped root cell is created around the triangulated surface input-geometry. This cell is recursively refined [Figure 6] using a hierarchical Octree data structure [8] until pre-defined spacing conditions, computed from the surface geometry, user input or a feature-based adaptation criterion are achieved. Also, non-unity aspect ratios can be employed to achieve both isotropic and anisotropic refinement, although the latter was not used in this study. After the subdivision refinements are completed, the cells that intersect the geometry are cut at the boundary level, yielding a surface mesh that approximates the original geometry shape at an appropriate level of refinement.

P_HUGG has the capability to generate pure Cartesian meshes using polyhedral elements (elements with an arbitrary number of faces). Such meshes contain hanging nodes [see Figure 7]. Some flow solvers cannot handle these types of meshes. In order to solve this problem P_HUGG uses transitional elements such as tetrahedrons, prisms and pyramids, thus generating a hybrid mesh [see Figure 8]. The transitional cells also help to improve the volume variations in the mesh [9].
Figure 7. Symmetry Plane Grid with Polyhedral Elements

Figure 8. Symmetry Plane Grid without Polyhedral Elements
III.B. P_OPT (Parallel Optimization-based Smoothing)

The general cutting process involved in mesh generation using the P_HUGG algorithm can create poorly shaped elements (elements with large aspect ratio) or inverted elements (elements with a negative volume) near the boundary. The purpose of smoothing is to improve the quality of a mesh by adjusting the node location while maintaining element connectivity. P_OPT perturbs each interior node and boundary node of the inviscid mesh to improve the quality of surrounding elements by improving a cost function [10]. The cost function is computed based on Jacobian and condition number of an element (defined below).

As indicated, P_OPT is capable of moving the boundary nodes while still maintaining the integrity of the surface mesh. Boundary nodes are projected back on to the original geometry surface mesh and then perturbed to improve the element quality. Special care is taken with critical points, which occur at distinct boundary intersections and sharp edges. P_OPT can handle both hybrid and fully tetrahedral meshes.

A cost function for an element \( C_e \) is computed using the condition number of the element (CN), i.e.

\[
C_e = 1 - \frac{1}{CN}, \quad \text{where} \quad CN = \frac{|J| |U^{-1}|}{3}
\]  

where, \( J \) is the Jacobian matrix of the element and it is the scalar triple product of normalized edge vectors emanating from a vertex of an element. A positive value of Jacobian indicates a valid element and a value below zero indicates an inverted element. A condition number of one represents an ideal element and increases to infinity for a degenerate element. Further, the cost function is zero for an ideal element and approaches one as the element collapses. Inverted elements have a cost function value greater than one.
Using Equation 2, a cost at each node \( (C_n) \) is computed, which is a combination of the average cost \( (C_{avg}) \) and maximum cost \( (C_{max}) \) of all the elements surrounding the node-

\[
C_n = f C_{\text{max}} + (1 - f) C_{\text{avg}} \quad \text{where} \quad f = \frac{C_{\text{avg}}}{C_{\text{max}}}
\]  

(2)

Only nodes that exceed a user specified cost function threshold value are allowed to move. This considerably reduces the overall computational cost of the mesh optimization process. Nodes are perturbed in Cartesian directions, along the edge to neighboring nodes and in the normal direction of the opposite face of the element. Also the user can control the movement of the boundary nodes and the number of interior layers of nodes, which are allowed to move near the boundary. This smoothing process can be performed in parallel using MPI protocols [11]. P\_OPT can also work with meshes generated by POINTWISE™. However normally POINTWISE™ produces a high quality, valid mesh that does not require any smoothing process.

III. C TENASI: Numerical Flow Solver

The flow solver used herein is a node-centered, finite volume, implicit scheme that can be applied to general unstructured grids [12]. The flow variables are stored at the vertices, and surface integrals are evaluated on the median dual surrounding each of these vertices. The non-overlapping control volumes formed by the median dual completely cover the domain and form a mesh that is dual to the elemental grid. Thus a one-to-one mapping exists between the edges of the original grid and the faces of the control volume.

TENASI solves the Reynolds-Averaged, Navier-Stokes (RANS) equations using a finite-volume formulation on unstructured meshes. The RANS equations in conservative, non-dimensional form without body forces are given as:

\[
\frac{\partial Q}{\partial t} + \int_{\partial n} \vec{F} \cdot \hat{n} \, dA = \frac{1}{Re} \int_{\partial n} \vec{G} \cdot \hat{n} \, dA, 
\]

(3)
where \( \hat{n} \) is the outward pointing unit normal for the control volume and \( Q \) is the conserved variable vector given by

\[
Q = \begin{bmatrix}
\rho \\
\rho u \\
\rho v \\
\rho w \\
\rho e_t
\end{bmatrix}
\]

Furthermore, the vectors \( \vec{F} \) and \( \vec{G} \) are

\[
\vec{F} = \begin{bmatrix}
\rho (u - x_t) \\
\rho u (u - x_t) \\
\rho v (u - x_t) \\
\rho w (u - x_t) \\
\rho h_t (u - x_t) + (\gamma - 1) M_t^2 p x_t
\end{bmatrix}
\]

\[
\vec{G} = \begin{bmatrix}
0 \\
\tau_{x x} \\
\tau_{y y} \\
\tau_{zz} \\
\tau_{x x} x + \tau_{y y} y + \tau_{zz} z - q_x
\end{bmatrix}
\]

where \( \rho \) is density; \( u, v, w \) are the cartesian components of velocity; \( x_t, y_t, z_t \) are cartesian components of mesh speed; \( \tau \) is the shear stress; \( q_x, q_y, q_z \) are components of conduction heat transfer; \( p \) is pressure; \( e_t \) is total energy; and \( h_t \) is the total enthalpy.

The solution algorithm employs iterative solutions of the implicit time-marching scheme. Even though the present study addresses only inviscid flows, the unstructured flow solver is capable of addressing viscous flows as well. The code uses MPI messages passing for inter processor communication. This study makes use of roe fluxes and Barth limiter.
III.D. SPACING

The SPACING algorithm makes use of a Riemannian metric tensor field [13] to specify a desired spacing for each point in the mesh and the spacing information is stored in a spacing field file. This spacing is computed based on gradients of the selected functions, such as velocity magnitude, pressure, Mach number, etc.

A solution plot file from the flow solver contains both the mesh and the converged solutions for conservative variables such as density, momentums and total energy. Other parameters are derived from these conservative variables. SPACING computes gradients of user-selected adaptation function/functions at each node in the mesh. The SPACING then computes an adaptation function \( AF \) for each edge using Equation 4, i.e.

\[
AF = \left( \frac{\| \nabla f_0 \cdot \hat{e} \| + \| \nabla f_1 \cdot \hat{e} \|}{2} \right)^p
\]

where, \( \nabla f_0, \nabla f_1 \) are the gradients at nodes of the edge; \( l \) is the length of the edge; \( p \) is a user specified constant, usually greater than or equal to one. The value of \( p \) is chosen to provide more influence in adapting the mesh in regions where there is a larger grid spacing as opposed to regions of small spacing in the presence of discontinuities, such as shocks. The mean and standard deviation \( (SD) \) of \( AF \) is computed and used to set a refinement threshold \( (AF_T) \), using Equation 5,

\[
AF_T = mean + c_R \cdot SD
\]

where, \( c_R \) is another user specified constant. This threshold or a user specified value is then used to define the local desired grid spacing \( (d) \) at each edge or node in the mesh using the equation,

\[
d = \left( \frac{AF_T}{\| \nabla f \cdot \hat{e} \|} \right)^p
\]

The vector \( \hat{e} \) can be the unit edge vector, producing the desired mesh spacing in the direction of the edge, or it can be the Cartesian coordinate directions, producing the desired X, Y and Z.
spacing values [14]. If the desired distance computed in the Cartesian coordinate directions are about the same or within a certain range then they are stored as a scalar value, else these spacing values are stored as a tensor (described next). Controls are set to prevent the desired size from exceeding a maximum global spacing parameter. The minimum spacing value is not restricted. Figure 9 show a spacing field using x-spacing contours.

The desired spacing and orientation can be defined by a symmetric, positive definite matrix $M$, as product of a rotation matrix $R$ and a scaling matrix $\lambda$ given in Equation 7. The columns of $R$ are eigenvectors of $M$ and corresponding to the principle directions. $\lambda$ is a diagonal matrix eigenvalues of $M$, those are specified as the inverse square of the desired distances along the principle directions.

$$M = [R][\lambda][R]^{-1} = [\vec{n}_1 \ \vec{n}_2 \ \vec{n}_3] \begin{bmatrix} h_1^{-2} & 0 & 0 \\ 0 & h_2^{-2} & 0 \\ 0 & 0 & h_3^{-2} \end{bmatrix} \begin{bmatrix} \vec{n}_1^{-T} \\ \vec{n}_2^{-T} \\ \vec{n}_3^{-T} \end{bmatrix}$$  \hspace{1cm} (7)

where, $\vec{n}_1$, $\vec{n}_2$, $\vec{n}_3$ are the edge vectors in the principle directions and $h_1$, $h_2$, $h_3$ are the desired spacing in those three directions.

The P_HUGG and the P_REFINE algorithms use a SPACING_FIELD library to generate an adapted mesh. The library stores the spacing field information in an Octree data structure for efficient searches. The P_HUGG and the P_REFINE algorithms make a call to the library by passing coordinates of an edge. The library retrieves the tensors near the edge and computes the maximum metric length, based on desired spacing values stored at points near that edge.

In SPACING, the user has the option to specify spacing-rboxes. These rboxes are defined to limit areas of refinement to regions of influence. This feature reduces the size of the problem considerably, if the user has prior knowledge about the regions where the variables would change rapidly. Multiple r-boxes can be used such that, when P_HUGG or P_REFINE use the spacing field, only edges that exist inside the rbox region get refined. This study uses two rboxes for each methods (see Figures 10 and 11). Figure 12 shows the spacing field with spacing-rbox applied.
Figure 9. Spacing Field with x-Spacing Contours Plotted

Figure 10. Spacing-rbox Setup for TETRAHEDRAL GRID Method

Figure 11. Spacing-rbox Setup for HYBRID GRID Method

Figure 12. Spacing Field with rbox defined
III.E. P_REFINE

P_REFINE performs the refinement of a hybrid or all-tetrahedral mesh through cell subdivision [14]. The procedure refines a volume mesh by adapting it to spacing field file created by SPACING. P_REFINE visits each edge of the volume mesh and computes a metric length based on the desired spacing information from the spacing field. The formula to compute metric length of an edge $AB$ is given in Equation 8.

$$d_{AB} = \sqrt{\overline{AB}^T [M] \overline{AB}}$$

where

$$\overline{AB} = \begin{bmatrix} x_b - x_a \\ y_b - y_a \\ z_b - z_a \end{bmatrix}$$

A metric length of unity means the edge $AB$ is at the target size. If the metric value is greater than a user defined threshold value (greater than 1, typically 1.25), that edge is marked for subdivision and a mid-edged node is created.

Subdivision of a face of an element should produce a triangle or a quadrilateral. Volume element subdivision must result in tetrahedra, pyramids, prisms or hexahedra. When P_REFINE is used to refine an all-tetrahedral mesh, in order to ensure that an all tetrahedral mesh is produced, the faces of the tetrahedra are forced to refine either one or all three edges. If only two edges of a face were marked, the algorithm just adds the third edge for refinement. And this requirement is applicable to each faces of an element.

A single subdivision of an edge need not always meet the metric spacing for that edge. So, multiple passes of P_REFINE often must be performed to achieve the desired metric spacing in the mesh. In order to control possible over-refinement of the mesh, the user can specify the number of new nodes added for each pass. Normally a very large number is used to prevent under-refinement of the mesh and to achieve a fully adapted mesh in rapid fashion.
IV. TECHNIQUES AND STRATEGY

This section explains the computational methods used to generate the unstructured grids used in the study, perform the flow analysis at supersonic test conditions, and modify the grids to obtain more accurate solutions.

The mesh generation process begins with the creation of far-field boundaries covering the domain of interest. The wing-body is placed such that the nose of the geometry is at the origin. Since there were limited data given in reference 6, the sting is roughly approximated as a simple body of revolution meeting the base of the model and extending roughly twice the body length behind the geometry. The out-flow boundary is located 150 cm behind the model from the origin. Since there is little pressure variation expected in front of the origin, the in-flow boundary is kept at 20 cm from the origin. Other far-field boundaries are at a distance ratio of 3.6 times the body length from the geometry, which is also the distance to the measurement plane for the wind-tunnel experimental data.

The surface mesh generated using POINTWISE™ (see Figure 13) is used to define the shape to create input geometry to use with P_HUGG. This mesh will not be part of the final P_HUGG mesh.

![Initial Surface Mesh of the Model with an Integrated Sting](image)

Figure 13. Initial Surface Mesh of the Model with an Integrated Sting
IV. A HYBRID GRID Method

A brief description of the process involved in the HYBRID GRID method is shown in Figure 14. This method generates a new volume mesh for each adaptation cycle.

The initial input geometry file consists of unstructured triangulated surfaces of the delta wing-body-sting configuration and the outer boundaries. \texttt{P\_HUGG} creates a Cartesian aligned unstructured hybrid volume mesh around the geometry. Spacing information based on user input and geometric parameter is used. Although \texttt{P\_HUGG} can use spacing information from a spacing field file, such information is not available at this phase because this requires a flow field solution file. The subdivision process retains a constant cell aspect ratio (unity for this case) in all three directions. A global maximum and minimum cell size is always enforced to control the number of elements in the mesh and to prevent indefinite cell refinement. After refinement, the cells near the boundary are made body-conforming through general cutting, and a standalone mesh that has preserved the essential shape of the original geometry is output.

![Figure 14. Process Diagram: HYBRID GRID Method](image-url)
Figure 15 shows the surface boundary generated by the general cutting process in P_HUGG. The grid generation process can introduce skewness into the mesh or sometimes can result in the generation of boundary cells with negative volumes producing a poor quality mesh. The quality of elements in this mesh is improved using P_OPT, where nodes are moved to produce a lower nodal cost function. A user specified threshold value of 0.01 is used to get high quality mesh. The threshold value (range is 0-1) can be increased to reduce the overall computational cost of the optimization process by compromising the quality of the mesh elements. Since the cutting process produces poor quality elements only near the boundary surface, process speedup in is achieved by allowing only three layers (more layers or all the interior nodes may be used) of nodes that are located near the boundary, along with the boundary nodes to move.

TENASI requires two input files to obtain a flow field solution: (1) an input file defines the boundary conditions being used, and (2) another file defines the flow parameters (example files are shown in the Appendix). Input parameters are used in a previous study [14]. For this compressible, supersonic, inviscid, steady-state configuration a Mach number of 1.68 and a zero degree angle of attack are used. A CFL (Courant-Friedrich-Lewy) number of two is used to obtain a flow solution in the HYBRID GRID method. The solution is advanced in time and the solution residuals are monitored for convergence. Once the solution is converged to an acceptable level, a pressure distribution along the symmetry plane (y=0) at the bottom far-field is plotted and compared with the available experimental data.
Using this converged solution plot file, the SPACING code computes spacing fields file, also known as an adaptation file. Pressure was used as the adaptation function in the present study. To set optimal values for $p$ and $C_R$, a parametric study was done (see Figure 16). It was found visually that, $p = 2$ and $C_R = 1$ gives an acceptable refinement by preventing aggressive refinement before the flow field can adequately develop (see the case in Figure 16 for, $p = 2$ and $C_R = 1$ ).
Figure 16. First Adapted Meshes Generated by P_HUGG

Used spacing field data Computed by Changing Parameter Values. (Symmetry Plane (y=0) Shown)
Figure 16. Continued
Also the user has the option to set a ‘spacing-rbox’ to reduce the computational cost. Once the regions of influence are defined, the SPACING code computes the desired spacing value only at this region. For this supersonic study the focus is on the shocks present in the mesh. Therefore, rbox is defined as region around the body plus the region between the body and the bottom boundary (see Figure 11).

With this new spacing field file plus user-defined spacing information, P_HUGG creates a new volume mesh from the same initial input geometry file. TENASI is then rerun to compute the flow field solutions. This process continues until a mesh with adequate adaptation is achieved.

IV.B TETRAHEDRAL GRID Method

A brief description of the process involved using the TETRAHEDRAL GRID method is shown in Figure 18. This method uses an initial volume mesh and adapts it with an updated solution for each adaptation cycle.

An all-tetrahedral volume mesh is generated using POINTWISE™. Once the surface element aspect ratio limit was resolved, the volume mesh generation is usually automatic. Volume mesh spacing is based on the spacing at the boundaries. A decay factor is used to control the gradation of the mesh spacing away from the boundaries. Meshes created by this method usually have high quality cells, and it is ready for use a flow solver.
Although the solver setup and input files are similar to the HYBRID GRID method, this method used a CFL of 5. As before, the residuals were monitored for convergence. A spacing field was then computed with ‘spacing-rbox’ using a converged solution plot file. The rbox defines the region between the geometry and bottom far-field and the region around geometry. In this case, only a part of the sting was considered in the rbox to reduce the computational cost even more.

Figure 10 shows the rboxes used for this method. These rboxes cover smaller regions compared to that of the HYBRID GRID method, and were reduced in size. This will help to reduce the total mesh-size. The criteria $p = 2$ and $C_r = 0.5$ were used in an attempt to obtain optimum refinement.

P_REFINE adapts the volume mesh using this newly obtained spacing field information. A user defined threshold (normally 1.2) was used to force the algorithm to work on edges with larger metric length. For each refinement pass the adaptation loop (see Figure 18) was repeated multiple times to reduce the maximum metric length in the total mesh. First solution is usually large and spread out. The spacing field computed from this file will cover a large area. As new solutions are calculated the shock gets refined. So the element refined before are not in the large gradient area now. So P_REFINE always starts from the initial mesh, this result to get a narrow refinement and this process carries through each time.
V. RESULTS

A delta wing-body configuration has been analyzed using the HYBRID GRID method and the TETRAHEDRAL GRID methods at supersonic cruise conditions and results are compared with experimental data. The near-field boom signatures are shown as the difference of the local and the free stream static pressure divided by the free stream pressure ($\frac{\Delta P}{P_\infty}$) versus the stream wise location ($x$) at a distance 3.6 times the body length below the vehicle on the symmetry plane.

V.I. Grid Generation Using HYBRID GRID

A perspective view of the initial mesh generated using P_HUGG is show in Figure 19, along with the symmetry plane mesh. The bottom boundary is intentionally made finer as the flow field variables to be compared with the experimental data are computed at that boundary on the symmetry plane. A closer view of the symmetry plane mesh was shown in Figure 8. This volume mesh contains a total of 273,548 nodes and 758,441 elements.

P_OPT is then executed to produces higher quality elements. Typical values of the cost function obtained are in the range 0.5 to 0.6, which are considered good values, taking into account the presence of a sharp cone shaped nose and sting in the model.
FIELDVIEW™ [15] is used to view the solution plot files obtained from TENASI. A convergence plot is shown in Figure 20 (red line). The solution algorithm was made to switch from first-order to higher-order flux approximations after the first 1000 iterations causing the discontinuous behavior seen at step 1000.

A variable extrapolation limiter is used in TENASI to prevent extrapolation of negative density and pressure. This causes the convergence to reach a limit cycle around 3500 time steps. An initial solution plot file with pressure contours is shown in Figure 21. From this solution a spacing field file is created, adapting to pressure gradients present in the mesh.

*Figure 20. Convergence Plots for HYBRID GRID Method*
Now \texttt{P\_HUGG} uses the spacing field file with the initial surface geometry and creates a new volume mesh. \texttt{P\_OPT} is then used to smooth the mesh and \texttt{TENASI} is rerun to compute new solutions.

By repeating this process, six new adapted \texttt{P\_HUGG} meshes are created (see Figure 26). The final mesh with crinkled symmetry plane is shown in Figure 24, which has a total of 28,923,819 nodes with 51,976,561 elements. In FieldView, when a location is defined in the coordinate direction, it creates a cutting plane through that point. When this cutting plane intersects an element, the entire element is displayed to get a better visualization, which is termed as a crinkled surface. Figure 22 shows the solution plot file with pressure contours on the symmetry plane obtained using the sixth adapted mesh. Even though \texttt{P\_HUGG} is a parallel algorithm, for this study the code
was executed serially, as the parallel version was not ready then for use, and further adaptations were not possible due to the lack of memory resources.

Upon getting a solution plot file from TENASI, a graph is plotted to compare the computed near-field pressure distribution with experimental data. For this study, the bow shock originates at the nose of the model, defined herein as the origin. Since there was little information in [6] about how to relate location to a known reference location, a trial and error method is employed to match surface grid position with corresponding position given in the experiment [6]. The best match was found when the computed data are moved -84.2 cm in the x-direction on the symmetry plane.

Figure 25 shows computed and measured pressure distributions. The computations shown include the result of six consecutive grid adaptations. It is evident from the plot that each adaptation produce computed values that are in better agreement with the experimental value.

In this study the pressure signals are calculated at 3.6 body lengths away from the model along the bottom boundary, using a second-order accurate scheme with variable extrapolation limiter. Although the solution scheme is second-order use of limiters requires that the scheme switch to first-order near the shocks.

If the meshes created after each adaptations (see Figure 26) are investigated, it can be seen that, as the refinement progresses, the algorithm starts to detect a higher gradients in front of the bow shock region. This is assumed to be inaccuracies in computing the gradient values in SPACING code. These “hot” spots occur at nodes connected to transition elements.
Figure 25. Pressure Signals Plotted Against Experimental Data
Figure 26. Pressure Adapted Grids Generated by P_HUGG with Spacing-rbox
V.II Grid Generation Using TETRAHEDRAL GRID

Figure 27 shows the perspective view of initial mesh created using POINTWISE™. This all tetrahedral volume mesh has 148,022 nodes with 816,280 cells. P_OPT can be used to improve the quality of this mesh, but normally POINTWISE™ generates high quality meshes.

This mesh was used by TENASI to compute flow field solutions. The flow parameters and boundary conditions were similar to the HYBRID GRID method. The solution was advanced in time and residuals were monitored for convergence. A convergence plot is shown in Figure 28 (red line). Since this method used CFL=5 (larger time step), the solution shows faster convergence than the previous case which was executed using CFL=2.

Like in the HYBRID GRID method, the flow solution algorithm switches from first order to higher order flux approximations after the first 1000 iterations, again causing the discontinuity in residual seen at step 1000. Also as before a limiter was used in TENASI to prevent extrapolation of negative densities and pressures. As seen in Figure 28, this causes the convergence to level off past about 1800 time-steps. Figure 29 shows a converged solution in the form of pressure contours. A pressure distribution plot was then made on the symmetry plane where it intersects the bottom far-field plane and compared with the experimental data.

Next SPACING was used to compute a spacing field file, which defines the desired spacing of each point in the mesh, based on the information from the above solution plot file. As before a
spacing-rbox was defined to restrict the refinement region. Figure 30 shows the x-spacing of the spacing_field on the symmetry plane.

Using the new spacing field file, P_REFINE was used to generate an adapted mesh. Once again TENASI is executed to compute a refined solution. Each time the residual is monitored for convergence. This process continues until a desired refinement is obtained as shown in Figure 31. Five adaptive refinement passes were done. Figure 28 shows a plot of residual for each adaptation cycle including that for the initial solution (line in red). As stated earlier, for initial case residual leveled off past 1800 time steps. However residuals for the next four adaptation cycles are seen to be converging, which is a little unusual in the case of a supersonic configuration where limiters are used. This could be the effect of refinement producing a mesh that is better aligned with the shock and thus avoids the limiter having to switch on and off very often, or at all.

Even though it was not necessary, each refinement pass was begun with the initial original volume mesh from POINTWISE™. This approach is used because, for the initial flow solution the gradients are large and spread out, and that causes the refinement algorithm to refine larger regions. As new spacing field files are computed using new flow solution, the grid refinement will confined to regions containing the shock. The initial mesh is refined multiple times to achieve an edge close to the metric length. For this study the refinement was repeated 3 times for each pass, and an all-tetrahedral mesh was enforced to be produced for each pass.

Figure 31 shows the final adapted mesh which contains 84M cells. Figure 32 shows the final solution file with pressure contours. Also the pressure distribution plots from consecutive solution plot file are given in Figure 34. Similar to the HYBRID GRID method, a trial and error was used to align the grid with corresponding known axial locations in the experiment.
Figure 28. Convergence Plot for TETRAHEDRAL GRID Method

Figure 29. Initial Mesh with Pressure Contours on Symmetry Plane

Figure 30. Spacing Field with x-spacing Contours on Symmetry Plane

Figure 31. Final Mesh with Crinkled Symmetry Plane

Figure 32. Final Mesh with Pressure Contours on y=0 Plane
Figure 3. Pressure Signals Plotted Against Experimental data
Figure 34. Pressure Adapted Grids Generated by \texttt{P\_REFINE} Using Spacing-rbox
Comparison between Figures 20 and 28 shows that the \texttt{P\_REFINE} meshes gave better flow solution convergence compare to those using \texttt{P\_HUGG} meshes, which could be the effect of refinement producing a mesh that is better aligned with the shock. Suppose a shock is going through a triangle (considering 2D for simplicity) as shown in Figure 36. \texttt{SPACING} will detect the gradient in the direction perpendicular to the shock and mark that edge for refinement. Since other edges are almost parallel to the shock and may not detect the presence of a gradient. So the elements near the shocks are getting aligned with the shock as the refinement progresses.

In the Cartesian aligned refinement approach (Figure 35), the square elements will never follow a diagonal shock.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure35.png}
\caption{Sketch of a Shock Passing Through a Cartesian Cell (3 Refinements Passes)}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure36.png}
\caption{Sketch of a Shock Passing Through a Tetrahedral Cell (3 Refinement Passes)}
\end{figure}
In order to address the effect of the use of limiter in computing the flow field solution, one last experiment was conducted. The adapted final mesh from the HYBRID GRID Method was used to compute a flow solution with no limiter turned on. The Solution was converged (see Figure 37). The computed pressure distribution was plotted against experimental data (see Figure 38) and found that no better results produced.

**Figure 37. Residual Plot for No-Limiter case**

**Figure 38. Pressure Distribution Comparison for No-Limiter Case.**
VI SUMMARY AND CONCLUSIONS

The objective of this study was to develop a mesh adaptation process that enable improved predictions of near field pressure distributions associated with shock waves generated by aircraft travelling at supersonic speeds.

Two grid generation procedures were demonstrated, each uses feature-based adaptation techniques to generate volume grids. An initial mesh is used with an unstructured flow solver and the resulting solution is then used to adapt the mesh. The new mesh is used to find an improved solution and this process can be repeated for better results.

A spacing field file, that contains the desired spacing at each mesh point based on flow field solution, was employed to adapt the mesh to selected functions. The HYBRID GRID method uses a robust grid generation algorithm to create hybrid volume meshes adapting to spacing field information. The TETRAHEDRAL GRID method generates an all-tetrahedral volume mesh and refines the mesh using cell subdivision while adapting to the mesh using information contained in the spacing field file.

The mesh size is one of the factors that define the computational resources required to conduct any CFD analysis. Although generic refinement can generate enormous numbers of cells and cause the computational cost to rise rapidly. The methods used herein restrict the region in which refinement occur. In the HYBRID GRID approach with a spacing field file the mesh will be refined to the finest sizing indicated by the spacing field up to the minimum cell size specified by the user. This minimum cell spacing along with maximum cell spacing, also defined by the user, controls the cell count. For this case initial mesh generated using P_HUGG had 758K cells and the sixth adapted mesh had 51.9M cells. This is very small number comparing to the number of cells that would have obtained (approximately 19B cells) by using uniform refinement of the domain. In the TETRAHEDRAL GRID approach, P_REFINE has to be executed multiple times to reach the required mesh size. P_REFINE does not have controls on its minimum cell size, but
**P_REFINE** allows the user to define the maximum number of edges that can be refined in each pass. Also the user can set a threshold, such that no edge that has a metric length below the pre-defined threshold is refined. These two parameters can be manipulated to control the mesh size. For the present case the initial volume mesh created by POINTWISE™ had 816K cells and after 5 refinements the mesh had 84M cells.

The effectiveness of the adaptation methods were examined using experimental data. It is evident that each adaptation produced improved the pressure predictions. After the final adaptation, the computed result matched the experimental result at many points. However some lack of agreement with the experimental data still exists. It is believed that this difference could be reduced with additional refinements could be done. Also the discrepancy between experimental and computed values for the aft expansion could be due to an inconsistency in the geometry definition of the sting-fuselage intersection.

The analysis shows that each method has advantages and drawbacks. The code **P_HUGG** creates a new hybrid mesh each time while adapting to the updated solution and produces higher quality Cartesian aligned meshes using an isotropic subdivision procedure. In the **P_REFINE** approach, the existing mesh is adaptively refined multiple times using the updated solution. The non-uniform subdivision procedure involved in the **P_REFINE** process refines a high aspect ratio element and could possibly generate poor quality elements by connecting a single node to multiple nodes and this will degrade the quality of the mesh. This will not happen with the **P_HUGG** approach. The **P_HUGG** algorithm generates poor quality or invalid elements near a boundary as a result of a general cutting process involved in the method which the **P_OPT** is used to obtain higher quality elements.

The HYBRID GRID method mostly produces high quality elements, however larger number of nodes associated with the method. TETRAHEDRAL GRID method involves less number of points; the elements may align with the shock and refinement process degrades the quality of elements.
FUTURE WORK

The feature-based adaptive refinement methods discussed in this study shows that each adaptation produces improved pressure predictions. However some lack of agreement with the experimental data still exists. More research in the following areas could improve the efficiency of the methods.

More refinement passes using updated parallel version of \texttt{P\_HUGG} algorithm will be conducted. The discrepancy between experimental and computed values for the aft expansion could be researched by modeling the sting- fuselage intersection differently. The effect of using higher order accurate (order three or above scheme to compute the flow field solution should be explored. The changes in the final result should be explored using a different adaptation function or combination of functions. The efficiency of the methods would be tested and compared with the other Mach numbers used in the wind tunnel experiment. Since the model is symmetric to \(y\) and \(z\) planes use of symmetry plane while solving the problem can reduce the problem size, which will be explored. The possibility of incorporating a tet-mesher to re-mesh the grid instead subdivision will be investigated, which is expected to produce high quality meshes.
LIST OF REFERENCES


1. Boundary Conditions Used by the Flow Solver

surface 1 [name=Far_left]: farfield;
surface 2 [name=Far_bottom]: farfield;
surface 3 [name=Far_outflow]: farfield;
surface 4 [name=Far_top]: farfield;
surface 5 [name=Far_inflow]: farfield;
surface 6 [name=Far_right]: farfield;
surface 7 [name=LU2]: inviscid;
surface 8 [name=LU1]: inviscid;
surface 9 [name=LL2]: inviscid;
surface 10 [name=LL1]: inviscid;
surface 11 [name=RU1]: inviscid;
surface 12 [name=RU2]: inviscid;
surface 13 [name=RL1]: inviscid;
surface 14 [name=RL2]: inviscid;
surface 15 [name=Body_UL]: inviscid;
surface 16 [name=Body_UR]: inviscid;
surface 17 [name=Body_LL]: inviscid;
surface 18 [name=Body_LR]: inviscid;
surface 19 [name=Bevel]: inviscid;
surface 20 [name=Cone_4]: inviscid;
surface 21 [name=Cone_2]: inviscid;
surface 22 [name=Cone_1]: inviscid;
surface 23 [name=Cone_3]: inviscid;
surface 24 [name=Sting]: inviscid;
2. Parameter File Used by the Flow Solver

```plaintext
# current run control
MULTIGRID = n
MULTIGRID-LEVELS = 2
MULTIGRID-AGGLOM-METHOD = 2
RESTART = n
NSTEPS = 5000
POSTPROCESS-FREQUENCY = 1000
POSTPROCESS-EXTRACT-FREQUENCY = 50000
POSTPROCESS-TAGWITHSTEP = 0

# control of time stepping
LOCAL-TIME-STEPPING = yes
TIME-STEP = 2.0e-3
CFL-RAMP = 200
CFL-START = 0.1
CFL = 2.0

# flow conditions
GAMMA = 1.4
REFERENCE-MACH = 1.68
SCALE-FACTOR = 1.0
REYNOLDS-NUMBER = 0.4139485e06
ARTIFICIAL-COMPRESSIBILITY = 5.0
FLOW-DIRECTION = ( 1.000000 0.0 0.000000 )
LIFT-DIRECTION = ( 0.000000 0.0 1.000000 )
USER-INITIALIZATION = 0
UINF-UREF = 1.0

# governing equations
FLOW-REGIME = compressible
UNSTEADY = 0
VISCOUS = n
2D = n #[experimental]
```

# turbulence model parameters
TURBULENCE-MODEL = 0
USE-LES = n
TURB-SUBITS = 10

# solution algorithm parameters
USE-ENTROPY-FIX = 2
IMPLICIT = 1
LIMITER = 1
USE-EDGE-LIMITING = 1
UNIFORM-LIMITING = 0
DENSITY-MIN-RELAXATION = 0.0
DENSITY-MAX-RELAXATION = 0.5
ENERGY-MIN-RELAXATION = 0.0
ENERGY-MAX-RELAXATION = 0.5
UVEL-MIN-RELAXATION = 0.5
UVEL-MAX-RELAXATION = 0.5
VVEL-MIN-RELAXATION = 0.5
VVEL-MAX-RELAXATION = 0.5
WVEL-MIN-RELAXATION = 0.5
WVEL-MAX-RELAXATION = 0.5
RECONSTRUCTION-ORDER = 1
SUBITERATIONS = 10
FIRST-ORDER-STEPS = 1000
JACOBIAN-METHOD = 3
HIGHER-ORDER-JAC = no
JUPDATE = 1
RECONSTRUCTION-GRAD-METHOD = 0
VISCOUS-GRAD-METHOD = 1
NEWTON-ITERATIONS = 1
VISCMETHOD = 0       # currently not used?
INVISCID-BC-RAMP = -1
USE-SUBEDGE = n      # [experimental]
SET-FARFIELD-PB = n

# reference frame parameters
RELATIVE-FRAME = n
RELATIVE-FRAME-OMEGA = ( 1.0 0.0 0.0 )
RELATIVE-FRAME-ORIGIN = ( 0.0 0.0 0.0 )

# misc parameters
ADJOINT = n
USE-IO-THREAD = n
VERTEX-REORDERING = y
ELEMENT-REORDERING = n
VITA

Jacob Chackasseril Varghese was born in Kottayam, a city in the state of Kerala, India on December 30, 1980. He was raised in Kottayam and went to high school at Mar Thoma Seminary High School in Kottayam. Also he completed Pre-Degree Course at Mahatma Gandhi University, Kottayam in 1998. Then, he went to the Visveswaraya Technological University, Bangalore and received B.Engg in Mechanical in 2002. Then he worked as a Production Engineer for the next 3 years. In January 2006, he moved to United States of America to pursue higher studies. And in August 2006, he enrolled in M.S program at University of Tennessee SimCenter: National Center for Computational Engineering, Chattanooga, Tennessee.