

Updates to Multi-Dimensional Flux Reconstruction for Hypersonic Simulations on Tetrahedral Grids

Peter A Gnoffo*

NASA Langley Research Center, Hampton, VA 23681-2199

The quality of simulated hypersonic stagnation region heating with tetrahedral meshes is investigated by using an updated three-dimensional, upwind reconstruction algorithm for the inviscid flux vector. An earlier implementation of this algorithm provided improved symmetry characteristics on tetrahedral grids compared to conventional reconstruction methods. The original formulation however displayed quantitative differences in heating and shear that were as large as 25% compared to a benchmark, structured-grid solution. The primary cause of this discrepancy is found to be an inherent inconsistency in the formulation of the flux limiter. The inconsistency is removed by employing a Green-Gauss formulation of primitive gradients at nodes to replace the previous Gram-Schmidt algorithm. Current results are now in good agreement with benchmark solutions for two challenge problems: (1) hypersonic flow over a three-dimensional cylindrical section with special attention to the uniformity of the solution in the spanwise direction and (2) hypersonic flow over a three-dimensional sphere. The tetrahedral cells used in the simulation are derived from a structured grid where cell faces are bisected across the diagonal resulting in a consistent pattern of diagonals running in a biased direction across the otherwise symmetric domain. This grid is known to accentuate problems in both shock capturing and stagnation region heating encountered with conventional, quasi-one-dimensional inviscid flux reconstruction algorithms. Therefore the test problems provide a sensitive indicator for algorithmic effects on heating. Additional simulations on a sharp, double cone and the shuttle orbiter are then presented to demonstrate the capabilities of the new algorithm on more geometrically complex flows with tetrahedral grids. These results provide the first indication that pure tetrahedral elements utilizing the updated, three-dimensional, upwind reconstruction algorithm may be used for the simulation of heating and shear in hypersonic flows in upwind, finite volume formulations.

I. Nomenclature

Bold face, lowercase variable names refer to vectors. Bold face, uppercase variable names refer to matrices.

Roman symbols

A	area
c	speed of sound
E	total energy per unit mass
$\mathbf{f}_{x'}$	flux in x' direction, $[\rho U_{x'}, \rho u U_{x'} + p n_{x,x'}, \rho v U_{x'} + p n_{y,x'}, \rho w U_{x'} + p n_{z,x'}, \rho U_{x'} H]^T$
H	total enthalpy per unit mass
$\vec{i}, \vec{j}, \vec{k}$	unit vectors in Cartesian coordinate directions (x, y, z) , respectively
$\vec{n}_{x'}$	unit vector in x' direction, $n_{x,x'} \vec{i} + n_{y,x'} \vec{j} + n_{z,x'} \vec{k}$
$\vec{n}_{A1}, \vec{n}_{A2}, \vec{n}_{A3}$	unit vectors orthogonal to faces $A1, A2, A3$
p	pressure
\mathbf{q}	conserved variables $[\rho, \rho u, \rho v, \rho w, \rho E]^T$

*Senior Research Engineer, Aerothermodynamics Branch; AIAA Fellow

Roman symbols

$\tilde{\mathbf{q}}_{x'}$	characteristic variable in direction x' , $d\tilde{\mathbf{q}}_{x'} = \mathbf{R}_{x'} d\mathbf{q}$
\mathbf{R}	eigenvector matrix for flux Jacobian, $\partial \mathbf{f}_{x'} / \partial \mathbf{q} = \mathbf{R}_{x'}^{-1} \mathbf{\Lambda}_{x'} \mathbf{R}_{x'}$
T	temperature
u, v, w	Cartesian velocity components
$U_{x'}$	velocity component in x' direction, $un_{x,x'} + vn_{y,x'} + wn_{z,x'}$
V	velocity magnitude $(u^2 + v^2 + w^2)^{1/2}$
x', y', z'	principal directions across element

Greek symbols

α	coefficient relative to face for defining derivative in an element, Eq. 33
β	coefficient relative to node for defining derivative in an element, Eq. 8
γ	ratio of specific heats
Λ	diagonal matrix of eigenvalues for flux Jacobian, $\partial \mathbf{f}_{x'} / \partial \mathbf{q} = \mathbf{R}_{x'}^{-1} \mathbf{\Lambda}_{x'} \mathbf{R}_{x'}$
ρ	density
Φ	pressure-based auxiliary limiter, Eq. 30
Ω	element volume

Subscripts

1, 2, 3	node indices in triangular element
1 - 2, 1 - 3, 2 - 3	edge identification as function of node endpoints
A	element id, centroid of element
e	value at centroid of element
GG	computed by Green-Gauss formulation of gradient at node
k	face oriented index defined by index of opposite node in the element
L	left virtual node in Option 1
LS	computed by Least Squares Gram-Schmidt formulation of gradient at node
n	node oriented index in element
R	right virtual node in Option 1
x'	function of direction x' , usually involving $U_{x'}$

II. Introduction

The simulation of hypersonic flows with fully unstructured (tetrahedral) grids has severe problems with respect to the prediction of stagnation region heating.^{1,2} The problems arise for two reasons.³

First, good shock capturing in the hypersonic regime requires alignment of the grid with the captured shock. Alignment here means that all surfaces of a control volume in contact with the shock are either parallel or orthogonal to the discontinuity. Any skewness present in a conventional upwind scheme (in which first-order reconstruction is a function of only two nodes on opposite sides of a shared face) results in a smearing of the flow, which manifests itself as a non-physical distribution of entropy from streamline to streamline crossing the shock. In the stagnation region, there is very little dissipation of these entropy gradients as they pass from the shock to the boundary-layer edge, and the surface heating is very sensitive to these non-physical gradients. Structured grids are generally easy to align with the captured bow shock, and NASA's main computational aerothermodynamic simulation codes LAURA^{4,5} and DPLR⁶ both align the structured grid with the captured bow shock to improve solution quality but accept the limitation that internal shocks (e.g., wake recompression or control surface compression) are computed with no such alignment. In fact, sharp double cone code validation test problems⁷ revealed that extremely fine structured grids were required to achieve a grid converged solution in a problem where structured grid could not be easily aligned with internally reflected shocks and shear layer over a separation bubble.^{8,9} Tetrahedral cell topologies by their very nature will always have at least one surface that is skewed to the captured shock.

Second, tetrahedral grids are not well suited to the preservation of symmetry and the biased tetrahedral grids used in the simulations of heating herein are particularly ill-suited for this purpose. Asymmetric elements and faces skewed to the symmetry surface tend to induce non-physical cross flows. Again, this

problem is particularly evident in the stagnation region where the physical flow velocities themselves are small and the non-physical, grid-induced velocities are of similar magnitude. Surface heating rates are a sensitive indicator of these problems. The combination of random jaggedness in the shock capturing process as well as non-physical cross flow velocities cross-couple to produce unacceptable surface heating predictions.

Problems with heating can be overcome if one uses a semi-structured grid (e.g., prisms) across the boundary layer and adapts the grid to the shock. Nompelis et. al.¹⁰ show excellent heating results for the cylinder test problem¹¹ on families of grids where a prismatic grid was adapted to the shock and acceptable heating results when an unbiased (random face orientation), tetrahedral grid was adapted to the shock. In this case, random jaggedness may disguise some issues with shock capturing on tetrahedral elements. Prismatic elements are relatively easy to generate on blunt body geometries and their superior performance with respect to aeroheating predictions have led to their use as standard practice in unstructured simulations. There is no better alternative with today's algorithms than to use prismatic elements to capture the boundary layer and bow shock. Note however that application of shock fitting in the context of unstructured grids may offer improved solution quality by bypassing the difficulties associated with capturing a strong bow shock.¹²

If one is doing a hypersonic simulation without any free shear layers or internal shocks the specialized application of prismatic elements at the body and bow shock is perfectly acceptable. If such flow structures exist — as they do in almost all interesting problems — then the accuracy of any algorithm must be questioned when it ignores special topological grid requirements where the features are harder to resolve.

A challenge problem was identified in which a highly biased, tetrahedral grid is applied to the simulation of a hypersonic flow over a 3D cylinder at $V_\infty = 5.000\text{km/s}$, $\rho_\infty = 0.001\text{kg/m}^3$ and $T_\infty = 200\text{K}$ in air ($M_\infty \approx 17$).^{2,11} (In fact, other hypersonic free stream conditions are acceptable. The challenge here is to recover the proper spanwise uniformity within 1%). A successful simulation requires a constant spanwise simulation of heating, pressure, and shear and symmetric simulation of these quantities on the right and left sides of the cylinder. The grid bias accentuates any issues with the algorithm. To date there have been no successful simulations of this challenge problem on the given grid using conventional upwind formulations. A recent thesis on a related unstructured grid without bias (random orientation of tetrahedra in spanwise direction) produced excellent heating symmetry using a high-order Discontinuous Galerkin finite element method with a PDE based artificial viscosity.¹³

A flux reconstruction algorithm that is not constrained by the local orientation of the grid is thought to have greater potential for a successful resolution of the challenge problem. The finite element method noted above¹³ has an inherently three-dimensional stencil in the evaluation of flux. Other approaches using more conventional, upwind, finite-volume formulations are known as “rotated upwind schemes”. These were originally generated to address problems in capturing shocks on structured grids that were oblique to the grid.^{14,15} The underlying ideas for this class of algorithm have been adapted for use in three-dimensional reconstruction using the node-based unstructured grid solver FUN3D and are applied to the challenge problems.

III. Multi-Dimensional Reconstruction Algorithm

A. Median-Dual Control Volume

A triangular grid system is presented in Fig. 1. The two-dimensional reconstruction algorithm is developed relative to this figure. The extension of the algorithm to three dimensions is presented in Appendix A. This derivation is repeated from the original source³ for the convenience of the reader with special emphasis placed on updates to the algorithm.

Assume a node based scheme as illustrated in Fig. 1 with (x, y) coordinates of all nodes $(1, 2, 3, \dots)$ given and dependent variables \mathbf{q} at these same nodes to be determined by solution of conservation laws. The conservation laws are discretized relative to the dual control volume, represented by dashed lines in Fig. 1. The dual volume around node 3 passes through the centroids of all elements surrounding the node (A, B, C, D, E, F) and the midpoints of all edges separating these elements.

B. Reconstruction Overview - Principal (Rotated) Direction

Consider a conventional, quasi-one-dimensional, first-order reconstruction of inviscid flux across face AcB separating the dual volumes around nodes 1 and 3. This flux is defined as a function of information at nodes

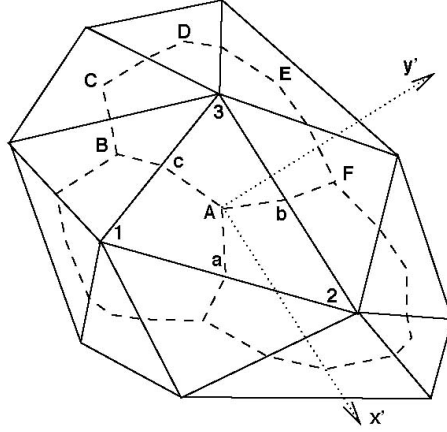


Figure 1. Schematic of node-based, unstructured grid system showing element centroids (uppercase letters), edge midpoints (lowercase letters), nodes (numerals), the dual volume (dashed lines), and the principal directions (dotted lines with arrowhead).

1 and 3 of edge c :

$$\mathbf{f}_{AcB} = \frac{1}{2} [\mathbf{f}_3 + \mathbf{f}_1 - \mathbf{R}_c |\mathbf{\Lambda}_c| \mathbf{R}_c^{-1} (\mathbf{q}_3 - \mathbf{q}_1)] \quad (1)$$

Inviscid fluxes are then defined using a loop over edges and the reconstruction direction is fully determined by the grid.

The basic idea in multi-dimensional reconstruction is that: (1) reconstruction directions should be based on local flow characteristics and not be constrained by the grid; and (2) the flux components for any face will utilize a stencil employing all nodes of a surrounding element. Conventional, node-based schemes loop over edges when using quasi-one-dimensional reconstruction. In contrast, the new multi-dimensional reconstruction loops over elements. Consider the two-dimensional element defined by nodes (1, 2, 3) with centroid A in Fig. 1. The computation of inviscid flux in this element employs the same infrastructure as the viscous flux.

A principal direction in the element, x' , is defined by the direction of $\vec{\nabla} \rho$, computed using the same Green Gauss evaluation of the derivatives that are already used to compute the viscous terms. If density is locally constant then the principal direction aligns with the local velocity computed as the average of nodal values. Inviscid flux within the element will be computed along and orthogonal to the principal flow direction.

The flux components are defined using nodal weights based on the Green-Gauss formulation for derivatives in the component directions. Thus,

$$\frac{\partial \mathbf{f}_A}{\partial x'} = \frac{1}{\Omega_A} \sum_{k=faces} \bar{\mathbf{f}}_k A_k n_{k,x'} \quad (2)$$

where $\bar{\mathbf{f}}_k$ is the average value of \mathbf{f} across all nodes defining surface k of element A . With reference to Fig. 1 this derivative can be written

$$\frac{\partial \mathbf{f}_{x',A}}{\partial x'} = \alpha_{3,x'} (\mathbf{f}_{2,x'} + \mathbf{f}_{1,x'}) + \alpha_{1,x'} (\mathbf{f}_{3,x'} + \mathbf{f}_{2,x'}) + \alpha_{2,x'} (\mathbf{f}_{1,x'} + \mathbf{f}_{3,x'}) \quad (3)$$

where

$$\alpha_{k,x'} = \frac{A_k n_{k,x'}}{2\Omega_A} \quad (4)$$

and k is a face index identified by the number of the opposite node. The sum $\sum_{k=faces} \alpha_{k,x'} = 0$ for any closed element. It is convenient to reorder and scale the coefficients as follows.

$$\frac{\partial \mathbf{f}_{x',A}}{\partial x'} = (\alpha_{2,x'} + \alpha_{3,x'}) \mathbf{f}_1 + (\alpha_{3,x'} + \alpha_{1,x'}) \mathbf{f}_2 + (\alpha_{1,x'} + \alpha_{2,x'}) \mathbf{f}_3 \quad (5)$$

$$= \frac{1}{\Delta x'} [\beta_{1,x'} \mathbf{f}_1 + \beta_{2,x'} \mathbf{f}_2 + \beta_{3,x'} \mathbf{f}_3] \quad (6)$$

Note that $\Delta x'$ can be interpreted as a distance across the element in x' direction. It is defined

$$\frac{2}{\Delta x'} = \sum_{n=\text{nodes}} (|\alpha_{n-1,x'} + \alpha_{n+1,x'}|) \quad (7)$$

where a cyclic indexing is assumed. Note that the original formulation³ is corrected here with a factor 2 in the numerator which provides the distance between virtual nodes. The reordering and scaling yield the following relations for $\beta_{n,x'}$.

$$\beta_{n,x'} = \Delta x' (\alpha_{n-1,x'} + \alpha_{n+1,x'}) \quad (8)$$

$$\sum_n \beta_{n,x'} = 0 \quad (9)$$

$$\sum_n |\beta_{n,x'}| = 2 \quad (10)$$

Two options for computing $\mathbf{f}_{x'}$ are developed. Both mirror the baseline quasi-one-dimension reconstruction using Roe's averaging across nodes. The first option averages dependent variables at two virtual nodes used in the reconstruction. The second option utilizes actual nodes and applies a weighted average to the computed flux on surrounding edges.

C. Option 1 - Virtual Node Averaging

A right and left virtual state for computing $\mathbf{f}_{x'}$ are computed as follows.

$$\mathbf{q}_{R,x'} = \sum_{n=\text{nodes}} (|\beta_{n,x'}| + \beta_{n,x'}) \mathbf{q}_n \quad (11)$$

$$\mathbf{q}_{L,x'} = \sum_{n=\text{nodes}} (|\beta_{n,x'}| - \beta_{n,x'}) \mathbf{q}_n \quad (12)$$

$$\mathbf{f}_{x'} = \frac{1}{2} \left[\mathbf{f}(\mathbf{q})_{L,x'} + \mathbf{f}(\mathbf{q})_{R,x'} - \mathbf{R}_{x'}^{-1} |\mathbf{\Lambda}_{lim,x'}| (d\tilde{\mathbf{q}}_{x'} - d\tilde{\mathbf{q}}_{x',lim}) \right] \quad (13)$$

where the left and right states of \mathbf{f} are functions of the left and right states of \mathbf{q} , respectively. The matrices $\mathbf{R}_{x'}$ and $\mathbf{\Lambda}_{lim,x'}$ are computed as functions of the Roe's average of the right and left states. (Forming $\mathbf{q}_{R,x'}$ and $\mathbf{q}_{L,x'}$ from averages of $1/\rho_i$, u , v , w , $1/T$ have provided the most robust simulations as judged by level of temperature undershoots across strong bow shocks.)

$$\begin{aligned} d\tilde{\mathbf{q}}_{x'} &= \mathbf{R}_{x'} (\mathbf{q}_{R,x'} - \mathbf{q}_{L,x'}) \\ &= \mathbf{R}_{x'} \sum_{n=\text{nodes}} \beta_{n,x'} \mathbf{q}_n \end{aligned} \quad (14)$$

$$d\tilde{\mathbf{q}}_{R,x'} = \Delta x' \mathbf{R}_{x'} \sum_{n=\text{nodes}} (|\beta_{n,x'}| + \beta_{n,x'}) \nabla \mathbf{q}_{n,GG} \quad (15)$$

$$d\tilde{\mathbf{q}}_{L,x'} = \Delta x' \mathbf{R}_{x'} \sum_{n=\text{nodes}} (|\beta_{n,x'}| - \beta_{n,x'}) \nabla \mathbf{q}_{n,GG} \quad (16)$$

$$d\tilde{\mathbf{q}}_{x',lim} = \Phi \text{ minmod} \left[2d\tilde{\mathbf{q}}_{R,x'}, 2d\tilde{\mathbf{q}}_{x'}, 2d\tilde{\mathbf{q}}_{L,x'}, \frac{1}{2}(d\tilde{\mathbf{q}}_{R,x'} + d\tilde{\mathbf{q}}_{L,x'}) \right] \quad (17)$$

where Φ is an auxiliary limiter ($0 \leq \Phi \leq 1$) defined in a later section.

D. Option 2 - Weighted Average of Edges to Principal Node

Select the largest of $|\beta_{n,x'}|$ to identify the principal node for construction of $f_{x'}$ and express it as a function of the remaining coefficients. Assume node 3 is the principal node. Then

$$\begin{aligned} df_{x'} &= \beta_{1,x'} f_1 + \beta_{2,x'} f_2 + \beta_{3,x'} f_3 \\ &= \beta_{1,x'} f_1 + \beta_{2,x'} f_2 - (\beta_{1,x'} + \beta_{2,x'}) f_3 \\ &= \beta_{1,x'} (f_1 - f_3) + \beta_{2,x'} (f_2 - f_3) \end{aligned} \quad (18)$$

The reconstructed flux in the direction x' , $\mathbf{f}_{x'}$, is computed as a weighted average of surrounding edges. Furthermore, it is noted that if any of the surrounding edges are parallel to x' then the weight of that edge will equal 1 and the weight of the other edges will equal zero.

$$\mathbf{f}_{x'} = |\beta_{1,x'}| \mathbf{f}_{1-3,x'} + |\beta_{2,x'}| \mathbf{f}_{2-3,x'} \quad (19)$$

$$\mathbf{f}_{1-3,x'} = \frac{1}{2} \left[\mathbf{f}_{1,x'} + \mathbf{f}_{3,x'} - \text{sign}(\beta_{1,x'}) \mathbf{R}_{1-3,x'}^{-1} |\Lambda_{lim,1-3,x'}| (d\tilde{\mathbf{q}}_{1-3,x'} - d\tilde{\mathbf{q}}_{1-3,x',lim}) \right] \quad (20)$$

$$\mathbf{f}_{2-3,x'} = \frac{1}{2} \left[\mathbf{f}_{2,x'} + \mathbf{f}_{3,x'} - \text{sign}(\beta_{2,x'}) \mathbf{R}_{2-3,x'}^{-1} |\Lambda_{lim,2-3,x'}| (d\tilde{\mathbf{q}}_{2-3,x'} - d\tilde{\mathbf{q}}_{2-3,x',lim}) \right] \quad (21)$$

where

$$d\tilde{\mathbf{q}}_{1-3,x'} = \mathbf{R}_{1-3,x'} (\mathbf{q}_1 - \mathbf{q}_3) \quad (22)$$

$$d\tilde{\mathbf{q}}_{2-3,x'} = \mathbf{R}_{2-3,x'} (\mathbf{q}_2 - \mathbf{q}_3) \quad (23)$$

$$d\tilde{\mathbf{q}}_{1-3,x',lim} = \Phi \text{ minmod} \left[2d\tilde{\mathbf{q}}_{1,x'}, 2d\tilde{\mathbf{q}}_{1-3,x'}, 2d\tilde{\mathbf{q}}_{3,x'}, \frac{1}{2} (d\tilde{\mathbf{q}}_{1,x'} + d\tilde{\mathbf{q}}_{3,x'}) \right] \quad (24)$$

$$d\tilde{\mathbf{q}}_{2-3,x',lim} = \Phi \text{ minmod} \left[2d\tilde{\mathbf{q}}_{2,x'}, 2d\tilde{\mathbf{q}}_{2-3,x'}, 2d\tilde{\mathbf{q}}_{3,x'}, \frac{1}{2} (d\tilde{\mathbf{q}}_{2,x'} + d\tilde{\mathbf{q}}_{3,x'}) \right] \quad (25)$$

$$d\tilde{\mathbf{q}}_{n,x'} = \Delta x' \mathbf{R}_{n,x'} \nabla \mathbf{q}_{n,GG} \vec{\mathbf{n}}_{x'} \quad (26)$$

Note again that Φ is the same auxiliary limiter used in Option 1.

E. Arguments to the Limiter

Simulations of heating and shear using multi-dimensional reconstruction presented in the original paper³ showed excellent symmetry but poor quantitative agreement with the baseline reference solutions on structured grids. At the time it was thought that small offsets between the centroid of the element and the midpoint of virtual nodes may have reduced order-of-accuracy. This line of inquiry was pursued by trying various corrective measures — none of which produced any significant improvement. A detailed examination of the limiting process revealed an inconsistency of magnitude and sign among arguments to the *minmod* limiter in a relatively large number of calls. Specifically, the arguments computed from the nodal values in Eqs. 15 and 16 that originally used $\nabla \mathbf{q}_{n,LS}$ were consistent with each other but too often inconsistent with the argument computed from Eq. 14. With attention now focused on the limiter arguments an algebraic error was noted in the original formulation of the distance between virtual nodes in Eq. 7. Even with this correction, the simulated heating and shear showed little improvement and the arguments to the *minmod* limiter still appeared to be inconsistent even in smooth regions of the flow. Attention thus moved to the formulation of gradients at the nodes.

The arguments to the minmod limiter are derived from gradients of primitive variables that must be constructed at each node by using information from surrounding nodes. The baseline algorithm used for this purpose in FUN3D is the Least Squares Gram-Schmidt process. This algorithm allows the gradients $\nabla \mathbf{q}_{n,LS}$ at each node to be calculated by looping over the edges in the mesh and distributing the contribution of each edge to each of the nodes. This edge-based formulation of gradients at a node is consistent with the edge-based flux reconstruction normally used in FUN3D.

The new multi-dimensional reconstruction is element-based rather than edge-based. Principal directions and virtual nodes within the element are derived from a Green-Gauss formulation of gradients within the element. A consistent formulation of gradients $\nabla \mathbf{q}_{n,GG}$ at the nodes can be defined using Green-Gauss by integrating $q \cdot \vec{\mathbf{n}} dA$ over the surface of the dual control volume and then dividing by the volume. This simple change was the final key to producing good heating and shear on tetrahedral grids in hypersonic flow. Arguments to the limiter are now consistently defined in smooth regions of the flow.

F. Component Assembly on Faces

Both option 1 (Eq. 13) and Option 2 (Eq. 19) present algorithms to define the flux, $\mathbf{f}_{x'}$, in the principal direction x' . The same logic is applied to construct $\mathbf{f}_{y'}$. Within each two-dimensional, triangular element there are three partial faces. The partial faces Ac and Ab can be combined into an equivalent face $A3$

separating node 3 from the other nodes in the element. In like manner Ac and Aa form $A1$ and Aa and Ab form $A2$. The flux across each equivalent face is constructed from the component flux in the principal direction and orthogonal to the principal direction. Thus, for element with centroid A as shown in Fig. 1 the flux through each equivalent face is

$$\mathbf{f}_{An} = n_{An,x'} \mathbf{f}_{x'} + n_{An,y'} \mathbf{f}_{y'} \quad (27)$$

where An refers to face separating node n from other nodes in the element.

G. Eigenvalue Limiter

An eigenvalue limiter (entropy fix) is required in Roe's method to suppress formation of a carbuncle in hypersonic flows over blunt bodies. In extreme cases, the carbuncle manifests itself as counterrotating flows displacing the bow shock at the symmetry plane. By putting a lower limit on the magnitude of the eigenvalues Λ_c in Eq. 1 the inviscid flux Jacobian is better conditioned, the carbuncle is suppressed, and overall stability of the algorithm is enhanced. The limiting value, λ_{min} , is set as $0 < \lambda_{min}/(V + c) \leq 1$. Each eigenvalue of Λ_c is then limited by

$$\lambda_{lim} = \begin{cases} \frac{\lambda^2}{4\lambda_{min}} + \lambda_{min} & \lambda < 2\lambda_{min} \\ \lambda & \lambda \geq 2\lambda_{min} \end{cases} \quad (28)$$

The limiting value must be small across the boundary layer or heating and shear may be over-predicted by as much as 20%. (The effect of the eigenvalue limiting across a free shear layer has not been quantified.) In the case of a structured grid the computational coordinate emanating from the body is easily defined and λ_{min} is a function of the computational coordinate direction. For example in structured-grid code LAURA λ_{min} is typically set to $0.3(V + c)$ for directions parallel to a solid boundary in computational space and it is set to $0.0001c$ (a small, positive definite value) for directions perpendicular to a solid boundary in computational space. In the unstructured-grid code FUN3D there is no intrinsic property of the grid alone that informs the algorithm about the orientation of an edge relative to a boundary layer. Consequently, the scaling must key on some local property of the flow. In the standard, edge-based reconstruction λ_{min} is kept small everywhere except in the vicinity of a shock. A pressure-based auxiliary limiter Φ (defined in the next section) is used to set $\lambda_{min} = 0.5(1 - \Phi)(V + c) + 0.0001c$. With 3D reconstruction the eigenvalue limiter is scaled down in the principal direction $\lambda_{min} = 0.0001c$ and is set as $\lambda_{min} = 0.5(V + c)$ orthogonal to the principal direction. As will be seen in the STS-2 results section, concern remains that the algorithm for defining λ_{min} may still require some adjustment to avoid excessive dissipation in the upper edge of the boundary layer if the principal direction rapidly changes at this location.

H. Auxiliary Limiter

Both options 1 and 2 are observed to sometimes admit temperature undershoots in capturing strong bow shocks. This undershoot may occur for strong shocks even in the case of a first-order reconstruction where $d\tilde{\mathbf{q}}_{x',lim} = 0$. Though every component of the reconstructed flux uses a baseline Total Variation Diminishing (TVD) algorithm the net formulation is not TVD. A floor on temperature is specified, but this practice causes the convergence to stall as temperatures are reset.

The problem is suppressed and solution robustness is enhanced by creating a shock sensor to use as an auxiliary limiter. (The shock sensor keys off the maximum pressure ratio across any two nodes in a tetrahedral element. Within any tetrahedral element $p_{max} = \max(p_1, p_2, p_3, p_4)$ and $p_{min} = \min(p_1, p_2, p_3, p_4)$ and $\phi = p_{max}/p_{min}$. The maximum pressure ratio across the element is used to define a C_1 continuous transition function Φ that goes from 0 to 1 at user defined set points ϕ_{max} and ϕ_{min} with zero slope at the set points.

$$\hat{\phi} = \min \left[1, \max \left(0, \frac{\phi_{max} - \phi}{\phi_{max} - \phi_{min}} \right) \right] \quad (29)$$

$$\Phi = \frac{1 - \cos(\hat{\phi}\pi)}{2} \quad (30)$$

This auxiliary limiter, plotted in Fig. 2, is a pre-multiplier to the minmod function in Eq. 17 in Option 1 and Eqs. 24 and 25 in Option 2. If the pressure ratio across an element is less than ϕ_{min} then the auxiliary limiter is not engaged. If the pressure ratio is greater than ϕ_{max} then the inviscid flux reconstruction is

first-order for all faces in the element. Recommended values are $\phi_{min} = 2$ and $\phi_{max} = 3$. Note that the limiter is potentially engaged in severe expansions that are under-resolved.

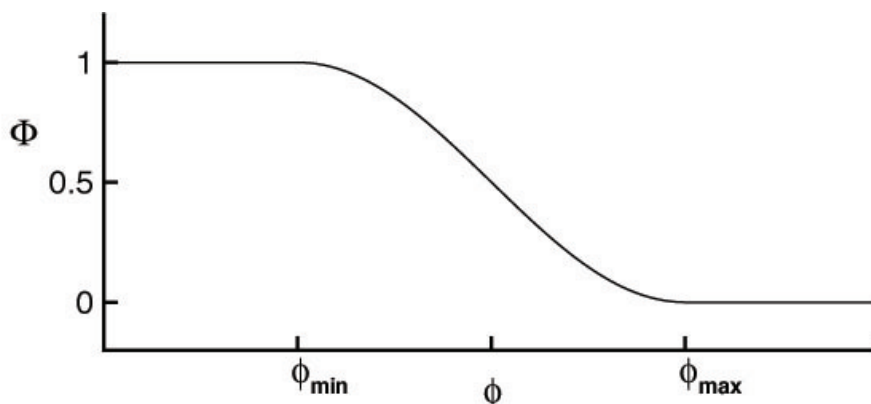


Figure 2. Auxiliary limiter as function of pressure ratio across element.

IV. Numerical Results

A. FUN3D

The three-dimensional reconstruction scheme is tested within FUN3D, which is a node based, fully unstructured, finite-volume solver of the Euler and Navier-Stokes equations.¹⁶ The baseline method uses Least Squares (LS) gradient information for second-order accurate inviscid flux reconstruction.¹⁷ However, as noted previously, a Green-Gauss formulation of gradients at the nodes is used for multi-dimensional reconstruction. Viscous gradients are also computed from a Green-Gauss formulation. A suite of modules includes all of the gas physics models in LAURA and VULCAN¹⁸ for thermodynamics, transport properties, chemical kinetics, and thermal relaxation. The baseline inviscid flux reconstruction algorithms utilize quasi-one-dimensional (edge-based) reconstruction using Roe’s averaging with Yee’s symmetric total variation diminishing algorithm adapted for unstructured grids.

B. Challenge Problem - Biased, Tetrahedral Grid on Cylinder with Spanwise Resolution

Accurate simulation of stagnation region heating in hypersonic flows is a key requirement for acceptance of any algorithm proposed for aerothermodynamic analyses. A structured-grid solution generated with LAURA is used as both a benchmark and to generate initial grids for use in FUN3D.² The 1m radius cylindrical cross-section test problem^{11,19} uses $V_\infty = 5.000\text{km/s}$, $\rho_\infty = 0.001\text{kg/m}^3$, and $T_\infty = 200\text{K}$. Sutherland’s law for air is used to define transport properties in all perfect-gas cases. As noted previously, this simple problem provides insight into the ability of a scheme to cleanly capture the bow shock, smoothly resolve the post-shock stagnation region flow and predict a smoothly varying stagnation region heating distribution. These flowfield characteristics are particularly sensitive to the inviscid flux reconstruction algorithm and problems that are not evident in well aligned, structured (hexahedral or prismatic) grids are exposed in the unstructured (tetrahedral) environment.¹⁹

The structured grid from LAURA, adapted to the shock and boundary layer, is converted from hexahedral elements to tetrahedral elements by adding diagonal edges consistently from minimum index to maximum

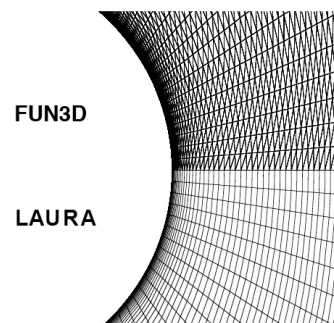


Figure 3. Structured grid (LAURA, bottom) and biased, unstructured grid (FUN3D, top) in plane orthogonal to cylinder surface.

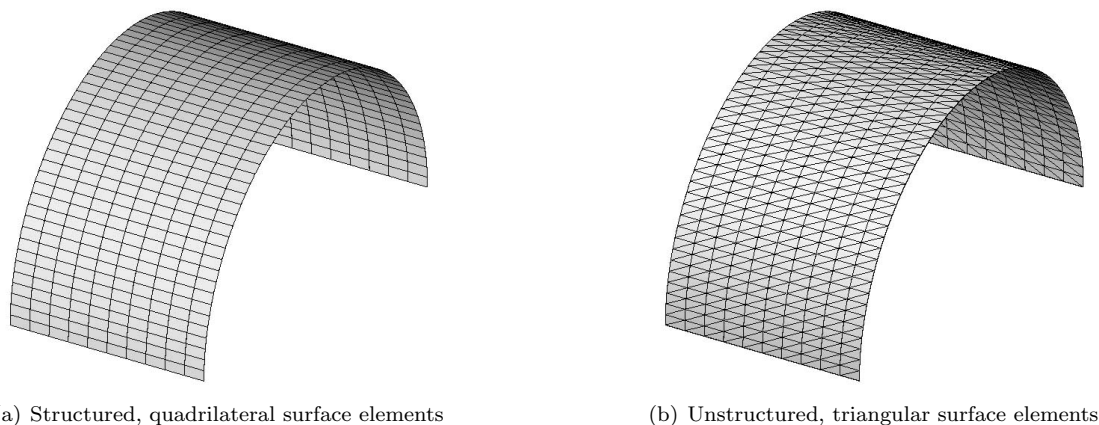


Figure 4. Surface mesh on cylinder with ten rows of spanwise cells.

index corners. A comparison of the grids in a plane of nodes perpendicular to the cylinder axis is presented in Fig. 3. The structured grid has 65 nodes from the body to the inflow boundary ahead of the bow shock and 61 nodes from the left to right outflow boundaries. Node placement is identical between the two grids. The placement of additional edges in the unstructured grid is the only difference. The strong biasing of diagonals in the grid is an intentional characteristic to expose algorithm deficiencies that may otherwise be averaged out.

A key element of this test problem is the addition of ten spanwise cells, shown vertically in Fig. 4, across the cylinder, providing additional degrees of freedom in the simulation to allow asymmetries to develop. Earlier tests¹⁹ (when the code was referred to as High Energy Flow Solver Synthesis — HEFSS) have shown that the single spanwise cell grids show good agreement with the structured code results in heating (Fig. 5). The spanwise degrees of freedom enable non-physical cross-flow velocities to develop and irregular shock capture in the spanwise direction that combine to corrupt the predicted heating distribution.

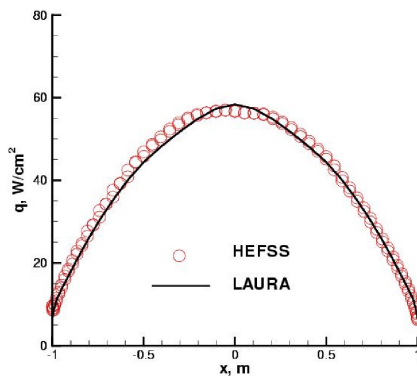
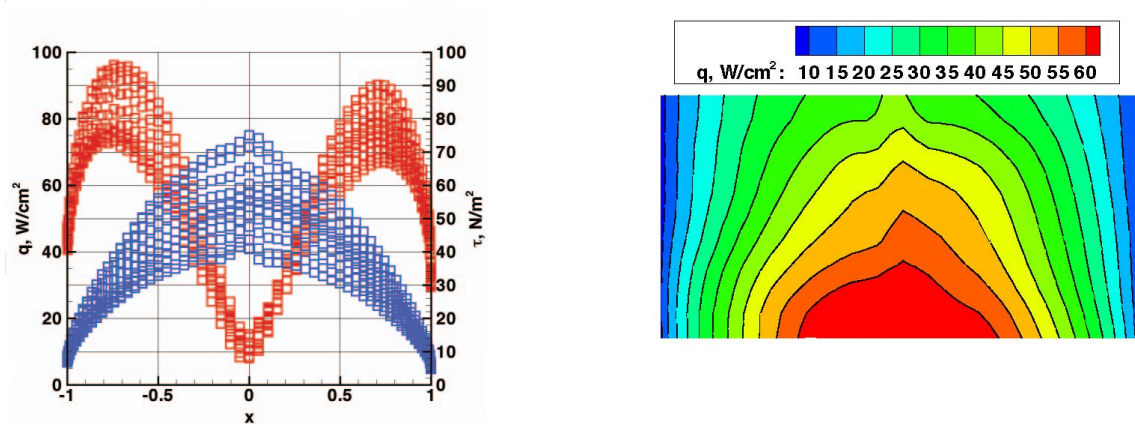


Figure 5. Surface heating over cylinder at standard test conditions with 5-species reacting air model and fully catalytic wall.

1. Conventional Upwind Results on Biased, Tetrahedral Grid

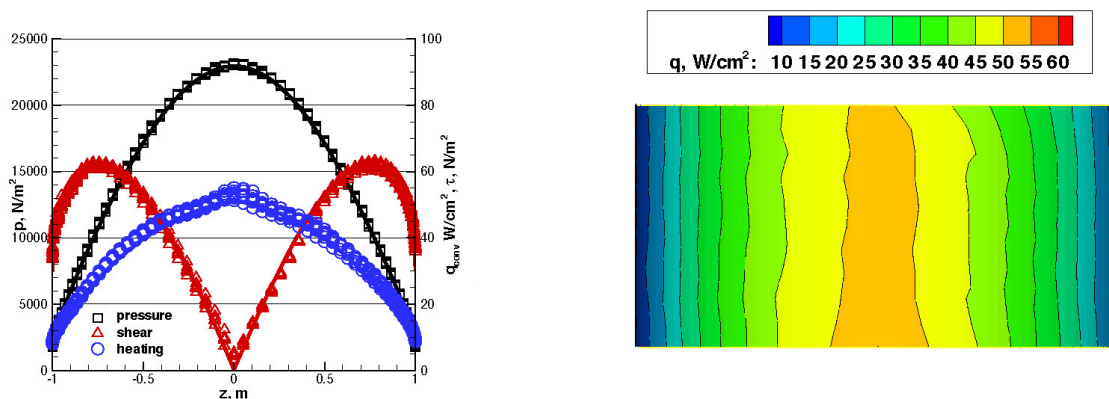
The challenge problem requires simulation of a perfect gas with $\gamma = 7/5$, $V_\infty = 5.0\text{km/s}$, $\rho_\infty = 0.001\text{kg/m}^3$, $T_\infty = 200\text{K}$ and $T_w = 500\text{K}$. A large spanwise variation in heating and shear for this simulation is evident in Fig. 6(a) with $\pm 30\%$ about the mean — an unacceptable result for aerothermodynamic analyses. Recall that the symbols at every x location should overplot and the peak value of shear on the right should equal the peak value of shear on the left. The LAURA result in this case is 52 W/cm^2 which is in good agreement with the spanwise mean of the heating at the stagnation point. The surface heating contours in Fig. 6(b) indicate that a slight drift in the velocity — probably induced by grid bias — produces a higher heating toward the front ($y = 0$) plane.



(a) Heating (blue squares) and shear (red squares) distribution for each of ten rows of spanwise cells.

(b) Surface heating contours.

Figure 6. Simulation results for challenge problem using conventional reconstruction on the tetrahedral grid with ten spanwise cells.



(a) Values of heating - blue circles, shear - red triangles, and pressure - black squares at nodes on surface. Figure shows results for all 11 surface nodes at a given x location.

(b) Heating contours on surface showing significant improvement in spanwise symmetry. Some edge effects at spanwise boundaries still persist.

Figure 7. Simulation results using Option 2 reconstruction on the tetrahedral grid with ten spanwise cells.

2. Results on Biased, Tetrahedral Grid

Option 2 results produce smoother contours in the vicinity of the captured shock than Option 1. Both options produce good quantitative and qualitative distributions of heating and shear. Research will continue on defining better averages of primitive variable at virtual nodes since Option 1 requires fewer operations.

Results for the challenge problem conditions are shown in Fig. 7. The Option 2 results are presented as symbols and a benchmark, grid-converged, structured grid solution (LAURA) for the same case are presented as solid lines. The heating contours show excellent spanwise uniformity in Fig. 7(b) in the interior. Some end effects are evident at the top and bottom boundaries. The heating and shear distributions are in good agreement with the benchmarks.

The element-based reconstruction algorithm requires three reconstructions per element whereas the baseline 1D algorithm requires one reconstruction per edge. The cylinder challenge problem with 10 spanwise

cells required 14.8 seconds per relaxation step as compared to 6.8 seconds per step for the baseline 1D algorithm. Some improvement is expected when the inviscid and viscous terms are computed in a single loop over elements. This refactoring will eliminate some redundancies occurring in both the inviscid and viscous formulations.

C. Challenge Problem - Biased, Tetrahedral Grid on 3D Sphere

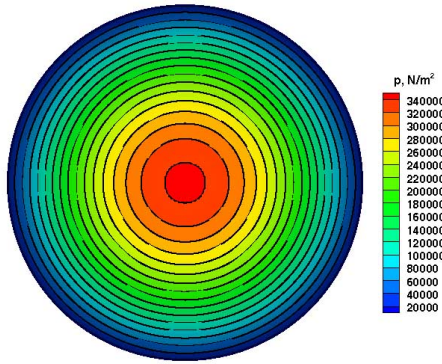
The 1m radius sphere test problem¹ uses $V_\infty = 4.167$ km/s, $\rho_\infty = 0.0216$ kg/m³, $T_\infty = 300$ K, and $T_{wall} = 800$ K. Three sets of figures are presented to illustrate solution quality with the baseline, quasi-one-dimensional reconstruction on hexahedral and tetrahedral grids and with the multi-dimensional reconstruction (Option 1) on tetrahedral grids. The three sets include surface pressure in Fig. 8, surface heating in Fig. 9, and surface shear in Fig. 10. A close-up view of the hexahedral surface grid and tetrahedral surface grid are presented in Fig. 11.

The three sub-figures on the left of Fig. 8 from top to bottom show that surface pressure contours are consistently predicted on both hexahedral and tetrahedral grid systems with either reconstruction algorithm. The axisymmetry is well maintained with all contour lines appearing as concentric circles. On the right side of the figure an axisymmetric solution generated with the structured grid code LAURA is used as a benchmark shown as a solid black line. The symbols indicate the 3D unstructured solution cuts at 0 degrees (along the x-axis) and at 45 degrees so that slices cut the surface grid at two different angles. The cut lines repeat across the axis. In the case of perfect symmetry, all symbols lie on top of each other. In the case of imperfect symmetry two sets of blue squares and two sets of red squares may be observed at any radial location. The three sub-figures on the right indicate good agreement with the structured grid benchmark and nearly perfect overplotting indicating excellent symmetry. Surface pressure is well predicted with every permutation of grid and reconstruction algorithm tested here. This result is consistent with results obtained on the spanwise cylinder test.

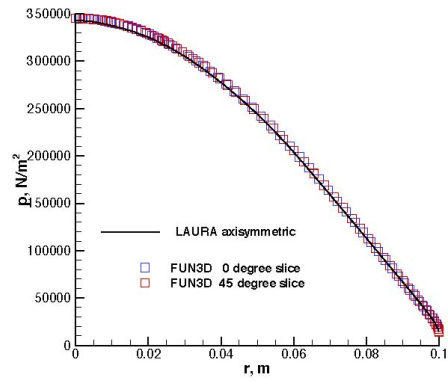
The solution quality for heating and shear are quite different than for pressure. The three sub-figures on the left of Fig. 9 show that the predicted heating in the stagnation region is sensitive to local grid orientation even in the case of a hexahedral grid system (Fig. 9(a)). A cross shaped contour pattern is formed in the stagnation region which transitions to concentric circles at approximately 40% of the maximum radius. (Fig. 9(b)) The standard reconstruction on the tetrahedral grids produces an even worse heating pattern as judged by lack of symmetry (Fig. 9(c)) and comparison to the benchmark solution in Fig. 9(d). These results are consistent with previous investigations. The Option 1 results with 3D reconstruction recover significant improvement in symmetry. A jag in the heating contours persists on the left side of Fig. 9(e) associated with a discontinuous change in the formation of the diagonal from the original hexahedral grid. Comparison with the benchmark solution at the stagnation region is improved compared to the 1D reconstruction on both the hexahedral and tetrahedral grids (Fig. 9(f)). The heating distribution is in good agreement with the benchmark around the entire hemisphere. There is a slight dip in the predicted heating levels at the stagnation point — this behavior is typical of upwind schemes that employ an entropy fix in regions where eigenvalues are near zero and it is accentuated in this test problem with the very fine surface grid around the stagnation point. Note that the hexahedral grid system retains excellent comparison with the benchmark across this same range (Fig. 9(b)). This comparison confirms the observation that edge based reconstruction is acceptable for well designed hexahedral grid systems but that the multi-dimensional reconstruction is required in tetrahedral grid systems.

The quality of the predicted shear follows much the same pattern as the quality of the predicted heating as measured by symmetry (concentric circular contours) and comparisons to the benchmark. The hexahedral results show good symmetry and compare well with the benchmark (Figs. 10(a) and 10(b)); however, shear goes to zero at the stagnation point so it is not a sensitive indicator of solution quality in the stagnation region. The 1D reconstruction on tetrahedral grids provides a very poor simulation of shear by metrics of symmetry and difference from the benchmark. (Figs. 10(c) and 10(d)). The 3D reconstruction shows good symmetry and agreement with the benchmark.

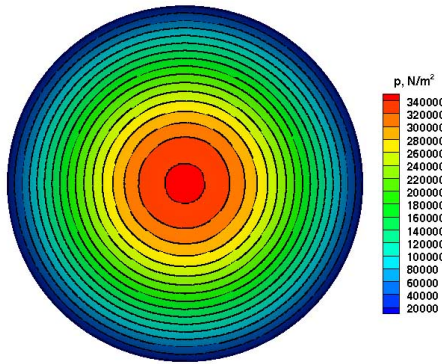
Two cuts are made through the stagnation streamline of the shock layer at 0 degrees and 45 degrees to show temperature contours (Fig. 12). Temperature contours from the 1D reconstruction (Fig. 12(a)) do not overlay in the stagnation region but otherwise show expected symmetry. The temperature contours from the 3D reconstruction (Fig. 12(b)) show excellent symmetry even in the stagnation region.



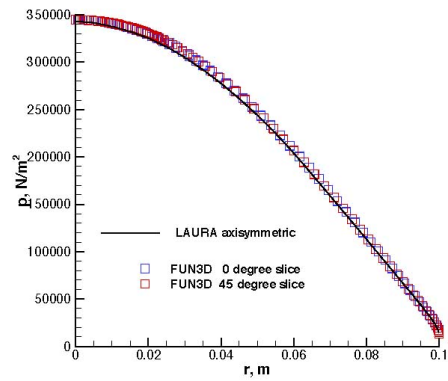
(a) Hex Grid, Surface Contours



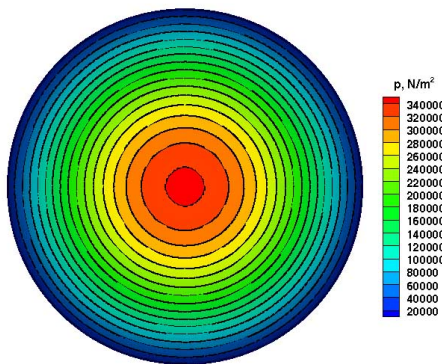
(b) Hex Grid, Line Cuts at 0 and 45 deg.



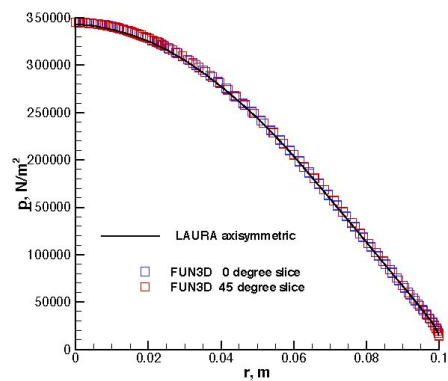
(c) Tet Grid, 1D reconstruction, Surface Contours



(d) Tet Grid, 1D reconstruction, Line Cuts at 0 and 45 deg.

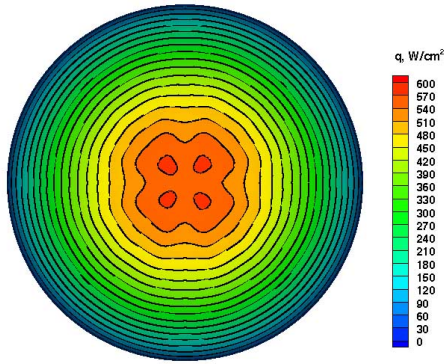


(e) Tet Grid, 3D reconstruction, Surface Contours

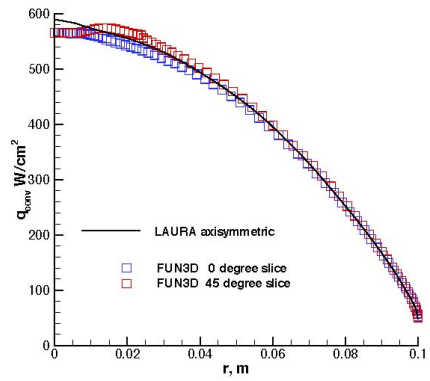


(f) Tet Grid, 3D reconstruction, Line Cuts at 0 and 45 deg.

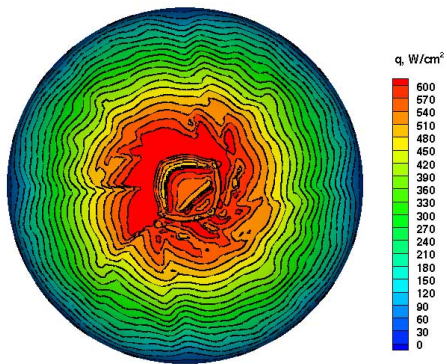
Figure 8. Surface pressure on sphere test problem.



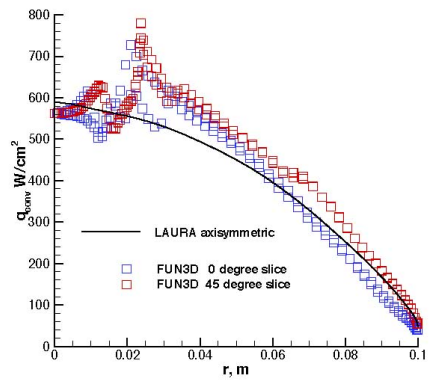
(a) Hex Grid, Surface Contours



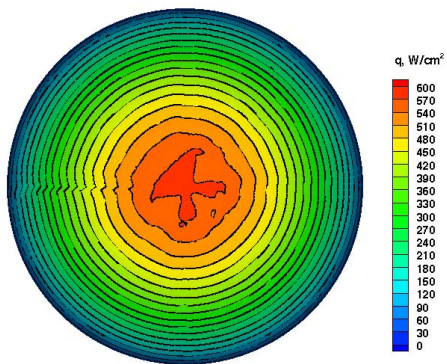
(b) Hex Grid, Line Cuts at 0 and 45 deg.



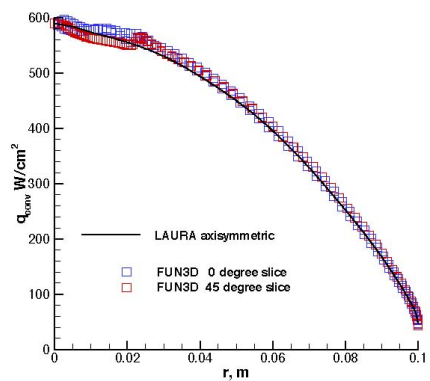
(c) Tet Grid, 1D reconstruction, Surface Contours



(d) Tet Grid, 1D reconstruction, Line Cuts at 0 and 45 deg.

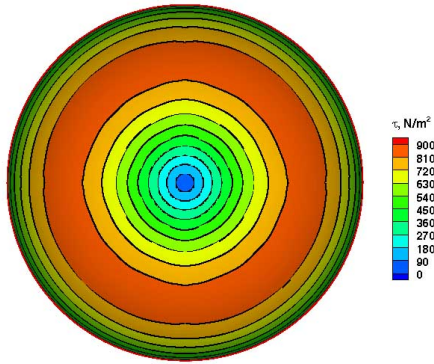


(e) Tet Grid, 3D reconstruction, Surface Contours

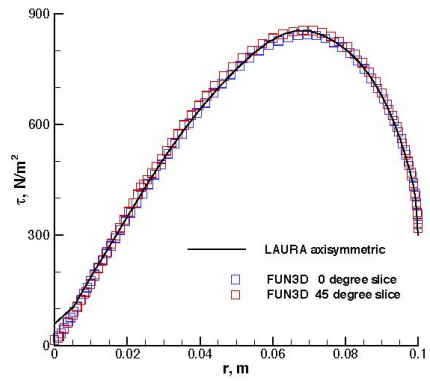


(f) Tet Grid, 3D reconstruction, Line Cuts at 0 and 45 deg.

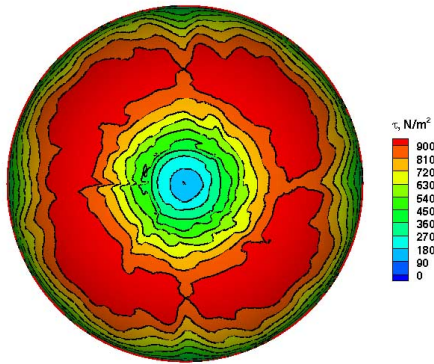
Figure 9. Surface heating on sphere test problem.



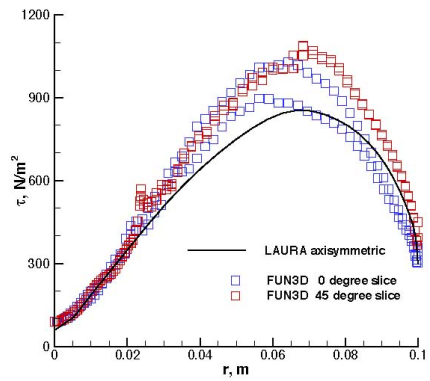
(a) Hex Grid, Surface Contours



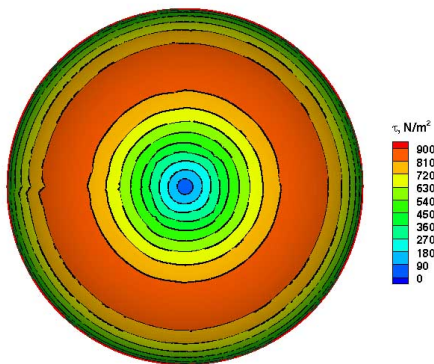
(b) Hex Grid, Line Cuts at 0 and 45 deg.



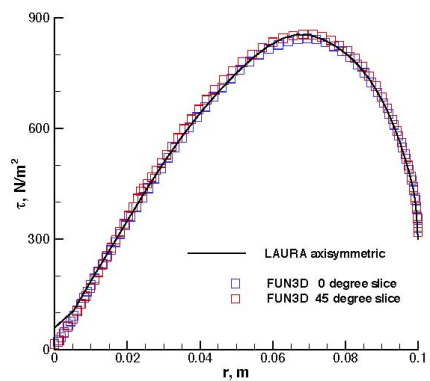
(c) Tet Grid, 1D reconstruction, Surface Contours



(d) Tet Grid, 1D reconstruction, Line Cuts at 0 and 45 deg.

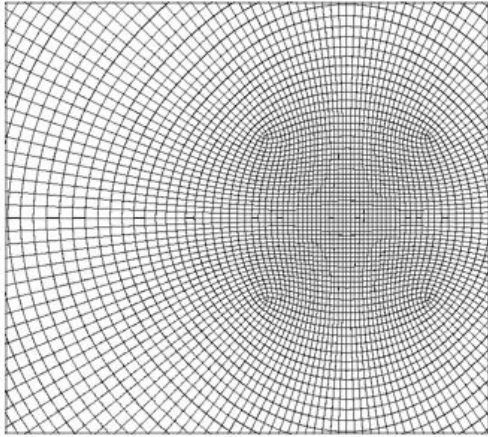


(e) Tet Grid, 3D reconstruction, Surface Contours

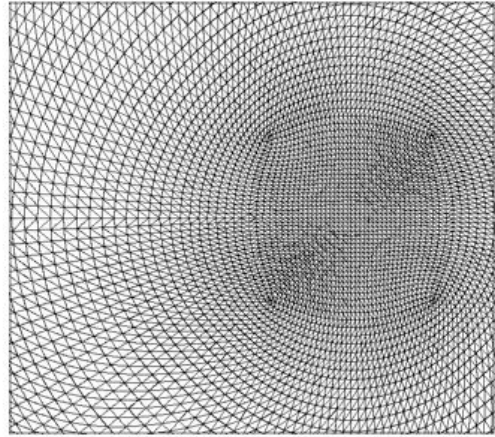


(f) Tet Grid, 3D reconstruction, Line Cuts at 0 and 45 deg.

Figure 10. Surface shear on sphere test problem.

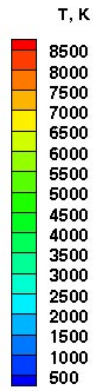
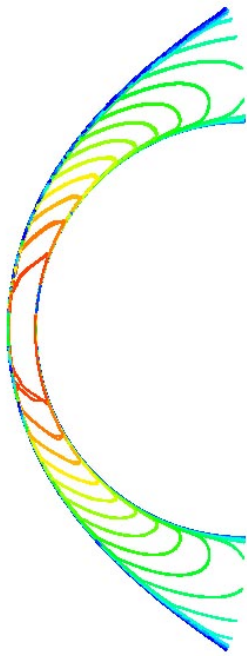


(a) Hex Grid

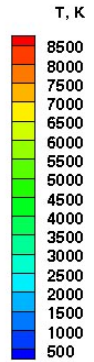
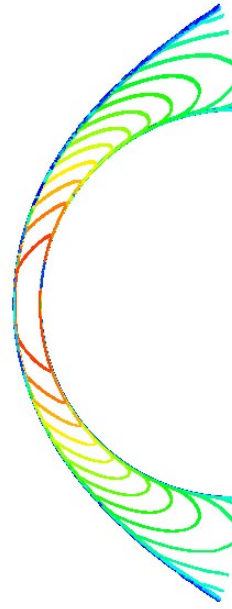


(b) Tet Grid

Figure 11. Surface grids on sphere test problem focused near stagnation point.



(a) Tet Grid, 1D reconstruction



(b) Tet Grid, 3D reconstruction

Figure 12. Shock layer temperatures on slices at 0 and 45 degrees on sphere test problem.

D. Sharp, Double Cone

A series of experiments to measure pressure and heating for code validation involving hypersonic, laminar, separated flows over double cones was conducted at the Calspan-University at Buffalo Research Center (CUBRC) in the Large Energy National Shock (LENS) tunnel.⁷ The configuration (Fig. 13) yields a flow topology with a large separation zone across the intersection of the two cones. An overview of the flow topology is evident in the temperature and pressure contours of Fig. 14. The shock off the forward part of the separation intersects the bow shock over the second cone. A transmitted shock from this intersection reflects off the wall to a contact surface. The extent of separation and the location of pressure maxima downstream of the reattachment point are sensitive to gas chemistry. These features are also a sensitive function of grid convergence. This sensitivity to physical models and grid convergence makes simulation of this flow an excellent code validation test. The experiments were the focus of a blind code validation study.⁷ A single case, Run 28, is featured here with inflow conditions $V_\infty = 2.664$ km/s, $\rho_\infty = 6.55 \cdot 10^{-4}$ kg/m³, $T_\infty = 185.6$ K, and wall temperature $T_w = 293.3$ K. These conditions correspond to $M_\infty = 9.59$ and $Re_\infty = 144,010$ /m in Nitrogen (to minimize effects of chemical reactions). The conditions are the same as originally presented to the study participants. A subsequent evaluation of test conditions was spurred by a consistent disagreement (20%) between experimental data for heating on the front cone prior to separation and multiple, grid converged simulations.²⁰ This study found that: (1) vibrational freezing of flow exiting the facility nozzle; (2) vibrational energy accommodation at the surface; and (3) small levels of flow non-uniformity must be included in the simulations to improve the heating prediction. These corrections to boundary conditions are not included in these results. The intent here is to show that the simulations on a tetrahedral grid system are consistent with the earlier, grid converged, structured-grid simulations.

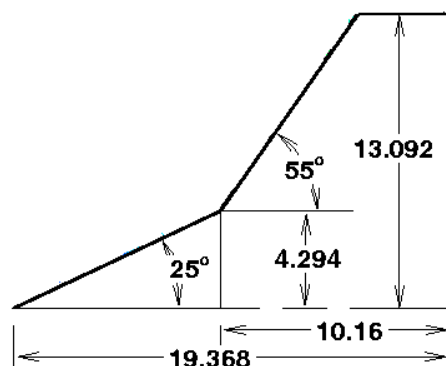


Figure 13. Dimensions (cm) of sharp, double cone.

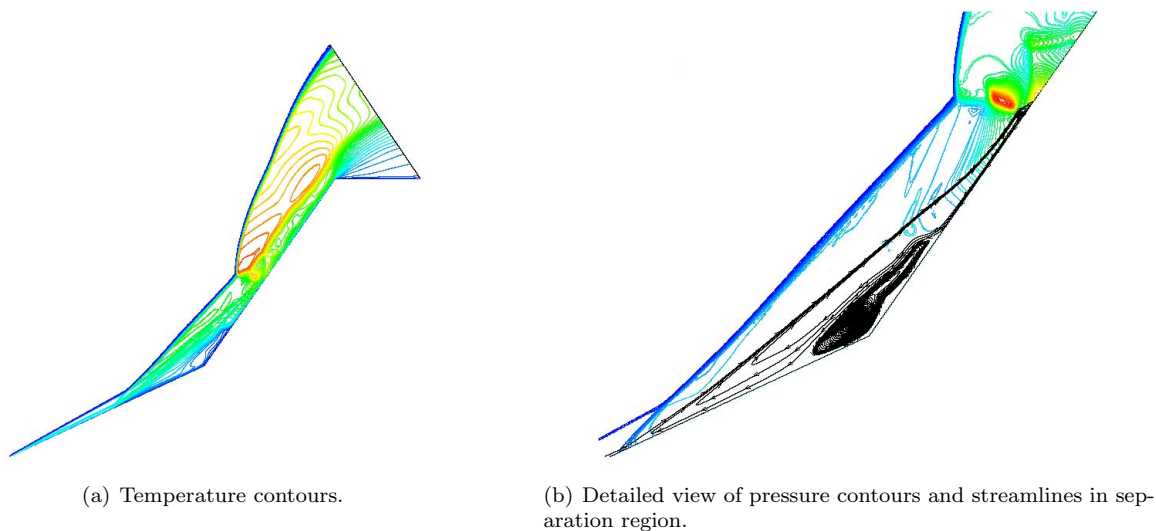


Figure 14. Flowfield over sharp, double cone.

The computed pressure coefficients and Stanton number distributions from three simulations are compared to experimental data in Fig. 15. All three simulations use the same set of nodes and Roe / STVD based inviscid flux formulation. The LAURA solution was generated for the blind validation study (solid blue line)⁸ and uses a cell-centered, structured-grid formulation. The FUN3D simulation on hexahedral grid (small black circles) uses conventional, one-dimensional, edge-based inviscid flux formulation. The FUN3D simulation on tetrahedral grid (small green triangles) uses the new three-dimensional, element-based inviscid

cid flux formulation. The “LAURA Path” descriptor in these figures indicated that FUN3D uses the same modules for thermodynamics and transport properties as the original LAURA code. The 3D reconstruction on tets indicates a slight increase in the extent of separation. The FUN3D hex simulation falls almost exactly under the original LAURA simulation. The primary differences between the two hex simulations is that LAURA is cell-centered and uses a directionally dependent entropy fix while the 1D reconstruction used for hex grids in node centered FUN3D uses a directionally independent entropy fix. The peak pressure appears beneath the transmitted shock at $x/L \approx 1.3$. All three simulations begin the rise to peak pressure slightly after the experimental data (in red squares). This delay is indicative of the slightly larger extent of separation with the original boundary conditions. The reflected waves in all three simulations closely follow each other downstream of the peak pressure and heating. This result demonstrates the consistency of the 3D reconstruction on tetrahedral grids with both structured and unstructured grid simulations using hexahedral grids for a complex flow in which principal directions vary significantly across the domain.

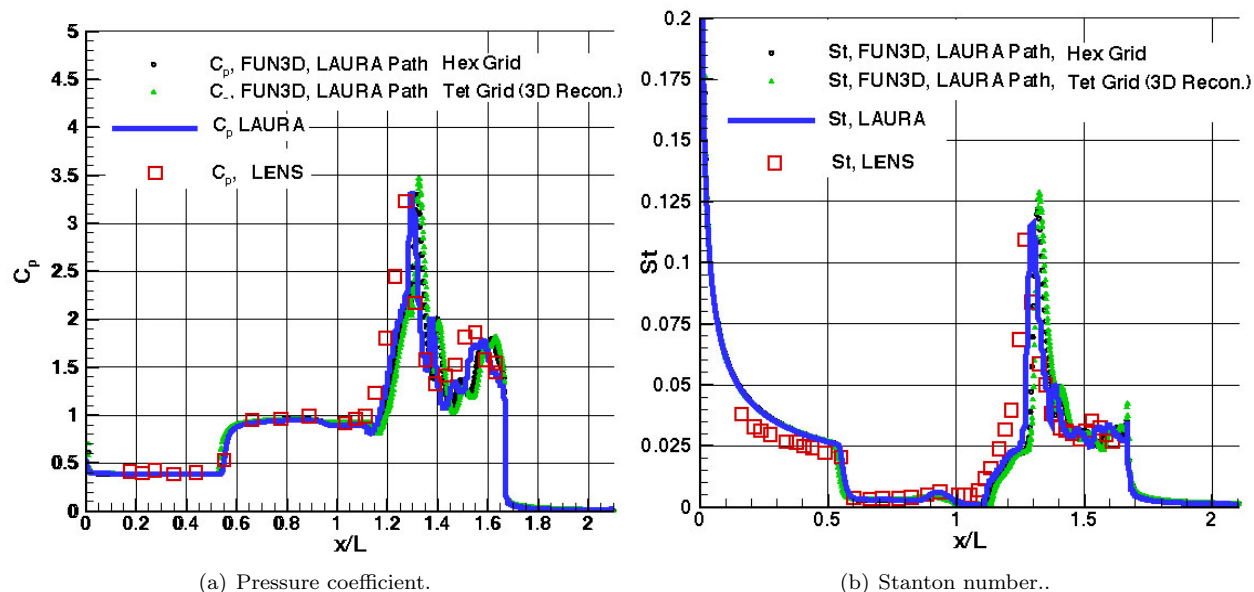


Figure 15. Comparison of experimental data with computation for the sharp, double cone, Run 28.

All three simulations employed local-time-stepping (constant CFL). When starting from a uniform flow this process produces a fully attached flow (no separation) with shocks set close to the surface in the early stages of the simulation. As the relaxation process continues, the shocks move out from the body and the primary separation bubble forms at the intersection of the two cones and slowly grows. In both the LAURA and FUN3D hexahedral grid simulations the separation bubble continues to grow until it reaches its converged state. In the 3D reconstruction on tetrahedral grids the growth of the separation bubble followed much the same history. However, when the leading edge of the separation bubble reached $x/L \approx 0.7$ the bubble shape became unsteady, giving way to a pulsing that could not be quenched with any variation of CFL or entropy fix. A steady solution could only be recovered by engaging a time accurate simulation with local sub-iterations employed between time steps. The use of time accuracy had been required by LAURA for a related flow condition at a higher Reynolds number (Run 24) of the original study. The 3D reconstruction appears to be more sensitive to this phenomena.

E. STS - 2

CFD simulations of heating over the shuttle orbiter (STS - 2) including effects of finite surface catalysis have been reported for the multi-block, structured grid solver LAURA.²¹ Some of these solutions were redone with a refined grid and updated physical models in support of the Columbia accident investigation and Return to Flight activities.²² The original simulations used a grid with approximately 1 million cells and a 7-species air model with an early model for surface catalysis. The more recent simulations on a refined

grid with approximately 8 million cells used a more current catalysis model²³ for reaction-cured glass (RCG) and a 5-species air model (NO^+ and e^- were ignored to save computational time with focus on relative effects due to damage). Two new simulations are run with FUN3D for a Mach 24 case:²¹ $V_\infty = 6.920$ km/s, $\rho_\infty = 5.750 \cdot 10^{-5}$ kg/m³, $T_\infty = 202$ K, $\alpha = 39.4$ deg with wall temperature defined assuming radiative equilibrium with a surface emissivity of 0.89. The structured grid was converted to FUN3D format and a simulation was executed using edge-based reconstruction on hexahedral elements. The hexahedral grid was converted into a pure tetrahedral system with identical nodes and a second FUN3D simulation was executed — this time using element-based reconstruction on the biased, tetrahedral-grid system.

Two cuts along the windside of the shuttle orbiter are illustrated in Fig. 16 over a flooded contour plot of surface heating. Heating levels from the original LAURA simulation from 1993, an updated LAURA²⁴ simulation, and the FUN3D simulation on a hexahedral grid and a tetrahedral grid are compared in Fig. 17 along these cut lines. The updated LAURA solution for heating is in excellent agreement with the FUN3D simulation on a hexahedral grid. These two simulations use identical modules for thermodynamics, transport properties, chemical kinetics, and surface catalysis. The heating distribution with these new simulations are generally in good agreement with the experimental data. These simulations did not include a deflected body flap. Though not shown here, comparisons along other cuts across the body (windside, leeward, circumferential) show equivalent trends. The older LAURA results over-predict both the flight data and the newer simulations along these windside cuts. It is thought that the updated model for surface catalysis is the primary reason for this difference though a systematic study has not been executed to check this supposition. The main point of this comparison is to demonstrate that the node based FUN3D provides equivalent results to the cell based LAURA when identical grids and physical models are employed.

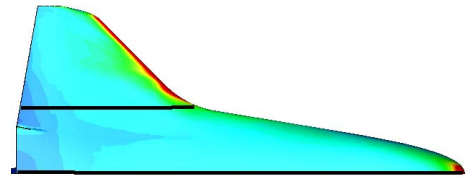


Figure 16. Cut lines over windside of shuttle orbiter showing surface heating contours.

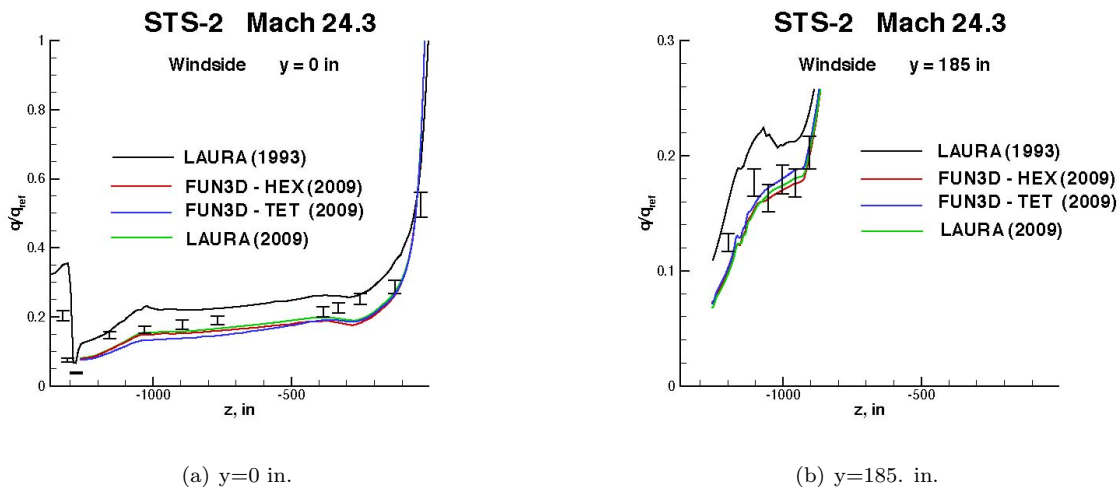


Figure 17. Comparison of heating on windside shuttle test case among original LAURA (black line), updated LAURA version 5 (green line), FUN3D on hex grid (red line), FUN3D on tet grid with 3D reconstruction (blue line), and experimental data (black error bars).

The next step is to demonstrate that the FUN3D simulation on tetrahedral grids with 3D reconstruction provides equivalent surface heating as the hexahedral grid simulation. This step has been obstructed by a sensitivity to the formulation of the outflow boundary condition off the trailing edge of the leeward side of the wing and vertical tail. The boundary condition has been revised to vacuum type boundary (enforce low pressure on exit) if flow reversal is detected through the outflow boundary. The intent is to nudge a reverse flow to exit the outflow boundary. This condition restricts the time steps allowed for stability. A second issue involves the algorithm for defining the principal direction. A modification to smoothly transition the

principal direction from orthogonal to strong gradients to streamline aligned appeared to be engaging while still in the upper portion of the boundary layer. This condition engages an unacceptably high value for the eigenvalue limiter crossing the boundary layer and results in a turbulent-like wedge of heating near the symmetry plane on the windside. Both of these issues have been addressed and the results are shown as solid blue lines in Fig. 17. The interplay of eigenvalue limiting as a function of the principal direction is a suspected cause of the remaining disagreement between the hexahedral and tetrahedral predictions of heating along the centerline of the shuttle. Comparisons of heating away from the centerline show very good agreement, as shown for example in Fig. 17(b).

V. Concluding Remarks

The ultimate goal for any flowfield simulation is to achieve a grid converged answer to a user specified target function (lift, drag, heat load) within a user-specified uncertainty in a fully automated algorithm. Automated grid adaptation through localized enrichment, coarsening, and movement is a necessary component of such a simulation capability. The present work is motivated by a vision that an underlying unstructured grid system composed of simplex tetrahedra in three dimensions provides the greatest flexibility to adapt to shocks and shear layers in a hypersonic flow. However, it is recognized that hypersonic simulations on such a grid system produce poor quality heating and shear — associated with algorithms that are sensitive to element edges skewed to critical flow structures. The present paper introduces a three-dimensional flux reconstruction algorithm that is less dependent on edge orientation and produces improved results for heating and shear, even on highly biased, tetrahedral grids.

Four test problems have been investigated: (1) a hypersonic flow over a three-dimensional cylinder, including resolution in the spanwise direction; (2) hypersonic flow over a three-dimensional hemisphere; (3) hypersonic flow over an axisymmetric, sharp, double cone; and (4) hypersonic flow over the shuttle orbiter. The tetrahedral cells used in the simulations are derived from a structured grid where cell faces are bisected across the diagonal resulting in a consistent pattern of diagonals running in a biased direction across the domain. Such a grid is known to accentuate problems in both shock capturing and stagnation region heating encountered with conventional, quasi-one-dimensional inviscid flux reconstruction algorithms. The test problems therefore provide a sensitive response to algorithm on heating.

The three-dimensional reconstruction shows significant improvement in symmetry (less than 3% dispersion) of heating with some penalty of ringing at the captured shock. Furthermore, it shows good agreement with benchmark solutions for pressure, heating, and shear obtained on structured grids. The cost for inviscid reconstruction is almost a factor 3 more work than the baseline algorithm using equivalent storage requirements. The net penalty on the complete simulation is just over a factor of two in the cylinder test problem. Some cost can be offset by the ability to combine the inviscid and viscous formulations in the same loop to make best use of data in cache. Also, a conventional reconstruction algorithm may be engaged to start the simulation and the three-dimensional reconstruction may be engaged at the end to provide acceptable results for heating.

The sharp, double cone test case demonstrates that the 3D reconstruction algorithm works well in an environment with complex flow topology (reflected shocks, free shear layer, large separation zone). A time accurate simulation was required in this case to obtain a steady solution. Subsequent tests show that the 1st-order time-accurate algorithm using sub-iterations within a time step is more robust than local time stepping for complex flow simulations.

Simulations on the shuttle orbiter (STS-2) demonstrate that the unstructured algorithm on hexahedral grids yields good agreement with both flight data and structured grid simulations for surface heating. Heating results with 3D reconstruction on tetrahedra are in good agreement with the structured grid simulations except for an under-prediction of heating along the windside symmetry line starting about 1/3 of the way down the body. This disagreement is thought to be associated with the eigenvalue limiting algorithm engaging near the boundary-layer edge. Research into a solution to this issue is ongoing.

References

¹Candler, G., Barnhardt, M., Drayna, T., Nompelis, I., Peterson, D., and Subbareddy, P., “Unstructured Grid Approaches for Accurate Aeroheating Simulations,” AIAA Paper 2007-3959, June 2007.

²Gnoffo, P. A., “Simulation of Stagnation Region Heating in Hypersonic Flow on Tetrahedral Grids,” AIAA Paper 2007-3960, June 2007.

- ³Gnoffo, P. A., “Multi-Dimensional, Inviscid Flux Reconstruction for Simulation of Hypersonic Heating on Tetrahedral Grids,” AIAA Paper 2009-0599, Jan. 2009.
- ⁴Cheatwood, F. M. and Gnoffo, P. A., “User’s Manual for the Langley Aerothermodynamic Upwind Relaxation Algorithm (LAURA),” NASA TM 4674, April 1996.
- ⁵Gnoffo, P. A., Gupta, R. N., and Shinn, J. L., “Conservation Equations and Physical Models for Hypersonic Air Flows in Thermal and Chemical Nonequilibrium,” NASA TP 2867, Feb. 1989.
- ⁶Wright, M. J., Candler, G. V., and Bose, D., “Data-Parallel Line Relaxation Method for the Navier-Stokes Equations,” *AIAA J.*, Vol. 36, No. 9, 1998, pp. 1603–1609.
- ⁷Holden, M. and Harvey, J., “Comparisons Between Experimental Measurements over Cone/Cone and Cylinder/Flare Configurations and Predictions Employing DSMC and Navier-Stokes Solvers,” AIAA Paper 2001-1031, January 2001.
- ⁸Gnoffo, P. A., “CFD Validation Studies for Hypersonic Flow Prediction,” AIAA Paper 2001-1025, Jan. 2001.
- ⁹Candler, G. V., Nompelis, I., Druguet, M.-C., Holden, M. S., Wadhams, T. P., Boyd, I. D., and Wang, W.-L., “CFD Validation for Hypersonic Flight - Hypersonic Double-Cone Flow Simulations,” AIAA Paper 2002-0581, January 2002.
- ¹⁰Nompelis, I., Drayna, T. W., and Candler, G. V., “Development of a Hybrid Unstructured Implicit Solver for the Simulation of Reacting Flows Over Complex Geometries,” AIAA Paper 2004-2227, June 2004.
- ¹¹Gnoffo, P. A., “Computational Fluid Dynamics Technology for Hypersonic Applications,” AIAA Paper 2003-3259, July 2005.
- ¹²Paciorri, R. and Bonfiglioli, A., “Numerical Simulation of Shock-Shock Interactions with an Unstructured Shock-Fitting Technique,” Sixth European symposium on aerothermodynamics for space vehicles, Nov. 2008.
- ¹³Barter, G. E., “Shock Capturing with PDE-Based Artificial Viscosity for an Adaptive, Higher-Order Discontinuous Galerkin Finite Element Method,” Mit dissertation, May 2008.
- ¹⁴Kontinos, D. A. and McRae, D. S., “Rotated Upwind Strategies for Solution of the Euler Equations,” Tech. rep.
- ¹⁵Dadone, A. and Grossman, B., “Characteristic-Based, Rotated Upwind Scheme for the Euler Equations,” *AIAA J.*, Vol. 30, No. 9, 1992, pp. 2219–2226.
- ¹⁶Anderson, W. K. and Bonhaus, D. L., “An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids,” *Comp. and Fluids*, Vol. 23, No. 1, Jan. 1994, pp. 1–21.
- ¹⁷Barth, T. J. and Jespersen, D. C., “The Design and Application of Upwind Schemes on Unstructured Meshes,” AIAA Paper 89-0366, Jan. 1989.
- ¹⁸White, J. A. and Morrison, J. H., “A Pseudo-Temporal Multi-Grid Relaxation Scheme for Solving the Parabolized Navier-Stokes Equations,” AIAA Paper 99-3360, June 1999.
- ¹⁹Gnoffo, P. A. and White, J. A., “Computational Aerothermodynamic Simulation Issues on Unstructured Grids,” AIAA Paper 2004-2371, June 2004.
- ²⁰Nompelis, I., Candler, G., and Holden, M., “Effect of Vibrational Nonequilibrium on Hypersonic Double-Cone Experiments,” *AIAA J.*, Vol. 41, No. 11, Nov. 2003, pp. 2162–2169.
- ²¹Gnoffo, P. A., Weilmuenster, K. J., and Alter, S. J., “Multiblock Analysis for Shuttle Orbiter Re-Entry Heating from Mach 24 to Mach 12,” Tech. Rep. 3, May 1994.
- ²²Reuther, J., Thompson, R., Pulsonetti, M., and Campbell, C., “Computational Aerothermodynamic Analysis for the STS-107 Accident Investigation,” AIAA Paper 2004-1384, Jan. 2004.
- ²³Stewart, D. A., “Surface Catalysis and Characterization of Proposed Candidate TPS for Access-to-Space Vehicles,” NASA TM 112206, July 1997.
- ²⁴Mazaheri, A., Gnoffo, P. A., Johnston, C. O., and Kleb, B., “LAURA Users Manual: 5.1-41601,” NASA TM 2009, 2009.

VI. Appendix A - Three-Dimensional Algorithm

The flux components are defined using nodal weights based on the Green-Gauss formulation for derivatives in the component directions. Thus,

$$\frac{\partial \mathbf{f}_A}{\partial x'} = \frac{1}{\Omega_A} \sum_{k=faces} \bar{\mathbf{f}}_k A_k n_{k,x'} \quad (31)$$

where $\bar{\mathbf{f}}_k$ is the average value of \mathbf{f} across all nodes defining surface k of element A . Following in parallel the development of the two-dimensional algorithm and defining node 4 as the additional node required to define the tetrahedral element one obtains

$$\begin{aligned} \frac{\partial \mathbf{f}_{x',A}}{\partial x'} &= \alpha_{3,x'} (\mathbf{f}_{2,x'} + \mathbf{f}_{1,x'} + \mathbf{f}_{4,x'}) + \alpha_{1,x'} (\mathbf{f}_{3,x'} + \mathbf{f}_{2,x'} + \mathbf{f}_{4,x'}) \\ &+ \alpha_{2,x'} (\mathbf{f}_{1,x'} + \mathbf{f}_{3,x'} + \mathbf{f}_{4,x'}) + \alpha_{4,x'} (\mathbf{f}_{1,x'} + \mathbf{f}_{2,x'} + \mathbf{f}_{3,x'}) \end{aligned} \quad (32)$$

where

$$\alpha_{k,x'} = \frac{A_k n_{k,x'}}{3\Omega_A} \quad (33)$$

and k is a face index identified by the number of the opposite node. The sum $\sum_{k=faces} \alpha_{k,x'} = 0$ for any closed element. It is convenient to reorder and scale the coefficients as follows.

$$\begin{aligned} \frac{\partial \mathbf{f}_A}{\partial x'} &= (\alpha_{2,x'} + \alpha_{3,x'} + \alpha_{4,x'}) \mathbf{f}_1 + (\alpha_{3,x'} + \alpha_{1,x'} + \alpha_{4,x'}) \mathbf{f}_2 \\ &+ (\alpha_{1,x'} + \alpha_{2,x'} + \alpha_{4,x'}) \mathbf{f}_3 + (\alpha_{1,x'} + \alpha_{2,x'} + \alpha_{3,x'}) \mathbf{f}_4 \end{aligned} \quad (34)$$

$$\frac{\partial \mathbf{f}_A}{\partial x'} = \frac{1}{\Delta x'} [\beta_{1,x'} \mathbf{f}_1 + \beta_{2,x'} \mathbf{f}_2 + \beta_{3,x'} \mathbf{f}_3 + \beta_{4,x'} \mathbf{f}_4] \quad (35)$$

Note that $\Delta x'$ can be interpreted as a distance across the element in x' direction. It is defined

$$\frac{2}{\Delta x'} = \sum_{n=nodes} (|\alpha_{n-1,x'} + \alpha_{n+1,x'} + \alpha_{n+2,x'}|) \quad (36)$$

where a cyclic indexing is assumed. Note that the original formulation³ is corrected here with a factor 2 in the numerator which provides the distance between virtual nodes. The reordering and scaling yield the following relations for $\beta_{n,x'}$.

$$\beta_{n,x'} = \Delta x' (\alpha_{n-1,x'} + \alpha_{n+1,x'} + \alpha_{n+2,x'}) \quad (37)$$

$$\sum_n \beta_{n,x'} = 0 \quad (38)$$

$$\sum_n |\beta_{n,x'}| = 2 \quad (39)$$

A. Option 1 - Virtual Node Averaging in 3D

The equations developed in the original section for two-dimensions are unchanged for three dimensions except that the sum over nodes includes a fourth node required to define a tetrahedral element.

B. Option 2 - Weighted Average of Edges to Principal Node in 3D

Select the largest of $|\beta_{n,x'}|$ to identify the principal node for construction of $f_{x'}$ and express it as a function of the remaining coefficients. Assume node 3 is the principal node. Then

$$\begin{aligned} df_{x'} &= \beta_{1,x'} f_1 + \beta_{2,x'} f_2 + \beta_{3,x'} f_3 + \beta_{4,x'} f_4 \\ &= \beta_{1,x'} f_1 + \beta_{2,x'} f_2 + \beta_{4,x'} f_4 - (\beta_{1,x'} + \beta_{2,x'} + \beta_{4,x'}) f_3 \\ &= \beta_{1,x'} (f_1 - f_3) + \beta_{2,x'} (f_2 - f_3) + \beta_{4,x'} (f_4 - f_3) \end{aligned} \quad (40)$$

The reconstructed flux in the direction x' , $\mathbf{f}_{x'}$, is computed as a weighted average of surrounding edges. Furthermore, it is noted that if any of the surrounding edges are parallel to x' then the weight of that edge will equal one and the weight of the other edges will equal zero.

$$\mathbf{f}_{x'} = |\beta_{1,x'}| \mathbf{f}_{1-3,x'} + |\beta_{2,x'}| \mathbf{f}_{2-3,x'} + |\beta_{4,x'}| \mathbf{f}_{4-3,x'} \quad (41)$$

$$\mathbf{f}_{1-3,x'} = \frac{1}{2} \left[\mathbf{f}_{1,x'} + \mathbf{f}_{3,x'} - \text{sign}(\beta_{1,x'}) \mathbf{R}_{1-3,x'}^{-1} |\mathbf{\Lambda}_{lim,1-3,x'}| (d\tilde{\mathbf{q}}_{1-3,x'} - d\tilde{\mathbf{q}}_{1-3,x',lim}) \right] \quad (42)$$

$$\mathbf{f}_{2-3,x'} = \frac{1}{2} \left[\mathbf{f}_{2,x'} + \mathbf{f}_{3,x'} - \text{sign}(\beta_{2,x'}) \mathbf{R}_{2-3,x'}^{-1} |\mathbf{\Lambda}_{lim,2-3,x'}| (d\tilde{\mathbf{q}}_{2-3,x'} - d\tilde{\mathbf{q}}_{2-3,x',lim}) \right] \quad (43)$$

$$\mathbf{f}_{4-3,x'} = \frac{1}{2} \left[\mathbf{f}_{4,x'} + \mathbf{f}_{3,x'} - \text{sign}(\beta_{4,x'}) \mathbf{R}_{4-3,x'}^{-1} |\mathbf{\Lambda}_{lim,4-3,x'}| (d\tilde{\mathbf{q}}_{4-3,x'} - d\tilde{\mathbf{q}}_{4-3,x',lim}) \right] \quad (44)$$

where

$$d\tilde{\mathbf{q}}_{1-3,x'} = \mathbf{R}_{1-3,x'} (\mathbf{q}_1 - \mathbf{q}_3) \quad (45)$$

$$d\tilde{\mathbf{q}}_{2-3,x'} = \mathbf{R}_{2-3,x'} (\mathbf{q}_2 - \mathbf{q}_3) \quad (46)$$

$$d\tilde{\mathbf{q}}_{4-3,x'} = \mathbf{R}_{4-3,x'} (\mathbf{q}_4 - \mathbf{q}_3) \quad (47)$$

$$d\tilde{\mathbf{q}}_{1-3,x',lim} = \text{minmod} \left[2d\tilde{\mathbf{q}}_{1,x'}, 2d\tilde{\mathbf{q}}_{1-3,x'}, 2d\tilde{\mathbf{q}}_{3,x'}, \frac{1}{2}(d\tilde{\mathbf{q}}_{1,x'} + d\tilde{\mathbf{q}}_{3,x'}) \right] \quad (48)$$

$$d\tilde{\mathbf{q}}_{2-3,x',lim} = \text{minmod} \left[2d\tilde{\mathbf{q}}_{2,x'}, 2d\tilde{\mathbf{q}}_{2-3,x'}, 2d\tilde{\mathbf{q}}_{3,x'}, \frac{1}{2}(d\tilde{\mathbf{q}}_{2,x'} + d\tilde{\mathbf{q}}_{3,x'}) \right] \quad (49)$$

$$d\tilde{\mathbf{q}}_{4-3,x',lim} = \text{minmod} \left[2d\tilde{\mathbf{q}}_{4,x'}, 2d\tilde{\mathbf{q}}_{4-3,x'}, 2d\tilde{\mathbf{q}}_{3,x'}, \frac{1}{2}(d\tilde{\mathbf{q}}_{4,x'} + d\tilde{\mathbf{q}}_{3,x'}) \right] \quad (50)$$

$$d\tilde{\mathbf{q}}_{n,x'} = \Delta x' \mathbf{R}_{n,x'} \nabla_{\mathbf{q}_n} \tilde{\mathbf{n}}_{x'} \quad (51)$$

C. Component Assembly on Faces in 3D

The flux across each equivalent face An in 3D is constructed from the component flux in the principal direction and orthogonal to the principal direction.

$$\mathbf{f}_{An} = n_{An,x'} \mathbf{f}_{x'} + n_{An,y'} \mathbf{f}_{y'} + n_{An,z'} \mathbf{f}_{z'} \quad (52)$$

where An refers to face separating node n from other nodes in the element.