Weighted Least-squares Cell-Average Gradient Construction Methods For The VULCAN-CFD Second-Order Accurate Unstructured-Grid Cell-Centered Finite-Volume Solver

Jeffery A. White[1]
NASA Langley Research Center, Hampton, VA, 23681, USA
Hioraki Nishikawa[2]
National Institute of Aerospace, Hampton, VA, 23666, USA
Robert A. Baurle[3]
NASA Langley Research Center, Hampton, VA, 23681, USA

The ability to solve the equations governing the hypersonic turbulent flow of a real gas on unstructured grids using a spatially-elliptic, 2nd-order accurate, cell-centered, finite-volume method has been recently implemented in the VULCAN-CFD code. The construction of cell-average gradients using a weighted linear least-squares method and the use of these gradients in the construction of the inviscid fluxes is the focus of this paper. A comparison of least-squares stencil construction methodologies is presented and approaches to augment the number of cells participating in the stencil while preserving accuracy are explored. Due to our interest in hypersonic flow, a robust multidimensional cell-average gradient limiter procedure that is consistent with the stencil used to construct the cell-average gradients is described and investigated. Canonical problems are computed to illustrate the challenges and investigate the accuracy, robustness and convergence behavior of the cell-average gradient methods on unstructured cell-centered finite-volume grids. Finally, thermally perfect, chemically frozen, Mach 8 turbulent flow of air around a blunt wedge is computed to demonstrate the robustness and convergence behavior of the new method for constructing stencils of use in a weighted linear least-squares gradient method for a hypersonic flow.

I. Introduction

The use of computational fluid dynamics (CFD) to characterize the external and internal flows typical of hypersonic vehicles is extremely challenging due to the complex physical modeling required to compute these flows. Nonetheless, over the past two decades, multiple CFD codes have been developed that are capable of computing these types of flows [1-4]. With the notable exception of the VULCAN-CFD code, the codes developed have almost exclusively employed unstructured grid methodologies. For the most part, these unstructured-grid codes provide significantly improved geometric flexibility at the expense of increased computational overhead, usually in the form of an increase in the number of processors required, relative to structured-grid codes. To address this additional overhead, there has been a concerted effort by the CFD community at large to develop unstructured grid codes that scale to "many" thousands of processors so as to either enable computation of "Grand Challenge Problems" or to perform less complex engineering analyses rapidly enough that they are relevant to engineering design time scales. Unfortunately, most engineers still work in a computational environment having finite resources where many programs compete for computational access. This competition naturally creates pressure on resource managers to configure their batch queuing software such that the time spent "in the queue" for jobs requiring "many thousands" of processors can become untenable from an engineering design point of view. This problem is further exacerbated in restricted access computational environments because computational resources are usually severely limited by the nature of the work. Moreover, as the number of processors required to rapidly compute a single "design point" solution increases, the number of processors available to compute other points in the design space decreases linearly, thereby adversely affecting the time required to cover the design space.

Historically, the development strategy for the VULCAN-CFD code has been to develop and implement solution methodologies that are efficient when computing the flows of interest to the scramjet community. This strategy resulted in the development of a "multiregion" framework in VULCAN-CFD [5,6] wherein the user has the ability to decompose the computational domain into multiple spatially-elliptic flow and/or parabolic/hyperbolic flow subdomains or "regions" where the flow solution is computed using the algorithm most appropriate for the flow physics. To date, this multiregion framework, has been instantiated by solving the spatially-elliptic flow regions with a structured-grid implicit time marching scheme and the parabolic/hyperbolic flow regions with a structured-grid implicit space marching

[1]Aerospace Technologist, Computational AeroSciences Branch, MS 128, Associate Fellow, AIAA.
[2]Associate Research Fellow, Research Department, MS 128, Associate Fellow, AIAA.
[3]Aerospace Technologist, High Speed Airbreathing Propulsion Branch, MS 168, Associate Fellow, AIAA.

scheme. The issue of geometric complexity has been addressed via the use of multiblock curvilinear structured grids within each region. However, when geometric complexity becomes too extreme, the time required to generate the multiblock curvilinear structured grids can become prohibitive. Therefore, given the aforementioned computational resource constraints and the maturation of unstructured-grid flow solution technology, the incorporation of an unstructured-grid spatially-elliptic flow solver capability into the VULCAN-CFD multiregion domain decomposition framework was desirable. Consequently, an effort to accomplish this goal was initiated utilizing a code developed as part of a hybrid structured/unstructured grid NASA Research Award (NRA), funded by the Fundamental Aeronautics Program as described by Spiegel et al. [7,8].

The cell-average and cell face gradient methods implemented by Spiegel et al. were based on the Green-Gauss approach for the cell-average gradient and an edge normal augmented approach for the cell-face gradient. However, these methods do not represent the current state of the art, consequently, the best cell-average gradient and cell face gradient construction practices available in the literature for 2nd-order, cell-centered, finite-volume, unstructured-grid flow solvers, were implemented. In addition, to improve convergence of the solver to steady state, the implicit scheme was rewritten to improve the left hand side as compared to the approximations used by the original LU-SGS and matrix-free SGS schemes and to couple the partitions during the linear solve subiterative process. Furthermore, a novel y+ adaptive turbulent wall boundary condition approach was also developed and implemented. This approach allows the turbulent wall boundary condition algorithm to choose between using, a solve-to-the-wall or a wall-matching-function wall boundary condition, for each wall cell face based on the local y^+_{wall}. Finally, all thermodynamic, chemical kinetic and turbulence models, as well as all relevant boundary conditions available in the structured-grid solver, were implemented in the unstructured-grid solver. All of these modifications were recently described in detail in [9].

The current work seeks to understand how the least-squares cell-average gradient stencil affects the stability and robustness of the unstructured solver and to choose a stencil construction methodology that produces stencils that are accurate, efficient and robust. Our previous effort [9] described the face neighbor (*fn1*) and the face neighbor of face neighbors (*fn2*) least-squares stencils. In the current work, a node neighbor (*nn*) stencil is investigated and a new stencil construction method (*symF*), designed to address the short comings of the *fn2* and *nn* stencils, that selectively augments the *fn1* stencil with cells from the *nn* stencil is proposed using a stencil augmentation methodology described in [10]. In addition, due to our emphasis on hypersonic flow applications, a robust multidimensional cell-average gradient limiter procedure that is consistent with the stencil used to construct the cell-average gradients is described and investigated. Canonical problems are computed to investigate the accuracy, efficiency, robustness and convergence behavior of the cell-average gradient weighted linear least-squares stencil schemes. Finally, hypersonic turbulent flow over a backward facing step, turbulent flat plate and a hypersonic inlet blunt leading edge are computed using the *fn2*, *nn* and *symF* stencil methods and compared.

II. Methodology

A. Least-Squares Cell-Average Gradient Construction

Cell-average gradients are perhaps the most important and one of the most difficult quantities to obtain accurately and robustly on irregular, unstructured grids. The cell-average gradients are required to accomplish three things when computing the residual of the discrete equations for each time step/cycle of the solution process: 1) to perform the higher-order reconstruction when computing the inviscid fluxes, 2) to compute the cell-face gradient when computing the viscous fluxes, and 3) to compute the source terms for the turbulence modeling transport equations. Moreover, there is evidence in the literature that a different definition of the cell-average gradient may be required to compute each of these quantities [11].

While no cell-average gradient method has been found to be accurate for all arbitrary polygons, with some caveats [12], the weighted linear least-squares method has been found to be the preferred method when computing cell-average gradients [12,13] for node-centered and cell-centered 2nd-order finite-volume schemes. Therefore, based on the results in the literature [11-15], the Weighted least-squares (WLSQ) method was chosen in [9] to replace the Green-Gauss method originally described in [7,8]. The WLSQ gradient method is based on a polynomial fit over a set of nearby cells. For second-order finite-volume schemes, the gradients need to be at least first-order accurate on general unstructured grids; and thus, it is sufficient to fit a linear polynomial. Suppose we wish to compute the gradient of a solution variable q at a cell i, and have a set $\{g_i\}$ of $N(\geq 3)$ nearby cells (i.e., a gradient stencil) available for fitting the linear polynomial:

$$q_j = q_i + \overline{\partial_x q_i}(x_j - x_i) + \overline{\partial_y q_i}(y_j - y_i) + \overline{\partial_z q_i}(z_j - z_i) \tag{1}$$

where $j \in \{g_i\}$, (x_i, y_i, z_i) and (x_j, y_j, z_j) denote the cell centroid coordinates of cell, i, and the set of neighbor cells, j, respectively, $\overline{\partial_x q_i}$, $\overline{\partial_y q_i}$ and $\overline{\partial_z q_i}$ are the derivatives we wish to compute. As the number of cell neighbors often exceeds three on 3-D unstructured grids, the polynomial fit (1) typically leads to an overdetermined problem:

$$Ax = b, \tag{2}$$

where

$$A=\begin{bmatrix} w_1(x_1-x_i) & w_1(y_1-y_i) & w_1(z_1-z_i) \\ \vdots & \vdots & \vdots \\ w_j(x_j-x_i) & w_j(y_j-y_i) & w_j(z_j-z_i) \\ \vdots & \vdots & \vdots \\ w_N(x_N-x_i) & w_N(y_N-y_i) & w_N(z_N-z_i) \end{bmatrix} \quad , \quad x=\begin{bmatrix} \overline{\partial_x q_i} \\ \overline{\partial_y q_i} \\ \overline{\partial_z q_i} \end{bmatrix} \quad , \quad b=\begin{bmatrix} w_1(q_1-q_i) \\ \vdots \\ w_j(q_j-q_i) \\ \vdots \\ w_N(q_N-q_i) \end{bmatrix} \quad , \tag{3}$$

and w_j is the weight applied to the equation corresponding to the neighbor cell j. The following inverse-distance weight is widely used in finite-volume methods:

$$w_j=\frac{1}{d_j^{p(l)}} \quad , \quad d_j=\sqrt{(x_j-x_i)^2+(y_j-y_i)^2+(z_j-z_i)^2} \quad , \tag{4}$$

where *p(l)* is a parameter ranging from zero (unweighted LSQ) to one (fully weighted LSQ) and l =1, 2 or 3, where 1 refers to the parameter used for the WLSQ gradients used in the inviscid flux reconstruction, 2 refers to the parameter used for the WLSQ gradients used in the construction of the cell face gradients for the viscous flux and 3 refers to the parameter used for the WLSQ gradients used in the construction of turbulence model source terms. The over-determined WLSQ system (2) can be solved in various ways. We chose to use QR factorization via the Householder transformation [16], which directly solves the overdetermined system as

$$x = R^{-1}Qb, \tag{5}$$

where Q is the orthonormal matrix and R is the upper triangular matrix generated from $\boldsymbol{A}$ by the QR factorization. The solution can be expressed in the following form:

$$\begin{bmatrix} \overline{\partial_x q_i} \\ \overline{\partial_y q_i} \\ \overline{\partial_z q_i} \end{bmatrix} = \sum_{j\in\{g_i\}} \begin{bmatrix} c_{ij}^x \\ c_{ij}^y \\ c_{ij}^z \end{bmatrix} (q_j-q_i) \quad , \tag{6}$$

where c^x_{ij}, c^y_{ij} and c^z_{ij} are the WLSQ coefficients to be computed and stored at all cells once for a given stationary grid. From Eq. (6), it is clear that the cost of the gradient calculation is directly proportional to the number of neighbors involved in the gradient stencil.

B. Least-Squares Stencils

1. The fn1, fn2 and nn stencils

In 3-D, the stencil of the linear least-squares average gradient operator must have at least 3 participating cells to be well posed. This condition can usually be met using the face neighbor stencil, (*fn1*), illustrated in the 2-D example shown in Fig. 1. However, on highly skewed grids, the *fn1* stencil may become biased and give rise to instabilities [13,14]. These instabilities can be alleviated by augmenting the stencil to reduce or eliminate the bias [13,14].

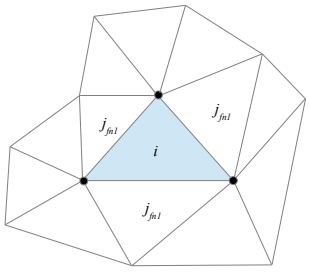


Fig. 1 The *fn1* stencil for computing the WLSQ cell-average gradient on a triangular grid.

Three augmentation approaches have been considered to address this difficulty. The first approach is to augment the *fn1* stencil cells with all of the cells that share a face with the cells of the *fn1* stencil, resulting in the face neighbors of face neighbors stencil, (*fn2*), shown in Fig. 2. The second approach is to augment the *fn1* stencil with all the cells that share a node with the nodes of cell *i,* resulting in the node neighbors stencil, (*nn*), shown in Fig. 2. However, as illustrated in Fig. 2, the *fn*2 stencil can result in gaps in the stencil that do not exist in the *nn* stencil that could potentially cause an instability when the grid is highly skewed [13,14]. Furthermore, Fig. 2 illustrates that, for 2-D triangular grids, the *fn2* stencil is a subset of the *nn* stencil, while the converse is true for quadrilateral grids, i.e., the *nn* stencil is a subset of the *fn2* stencil. Also note that Fig. 3 illustrates that the *nn* stencil is more spatially compact than the *fn2* stencil for quadrilateral grids, and by analogy, prismatic grids. In 3-D, the *nn* stencil of hexahedral and prismatic grids are not a

subset of the *fn2* stencil. Moreover, the *nn* stencil of nonhexahedral grids also has the potential for there to be many more cells in the *nn* stencil than in the *fn2* stencil thereby requiring significantly more storage and operations to compute the gradient.

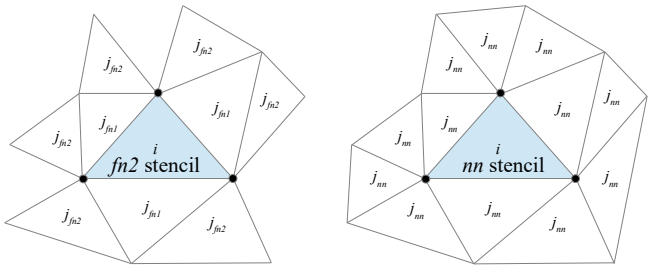


Fig. 2 The *fn2* and *nn* stencils for computing the WLSQ cell-average gradient on a triangular grid.

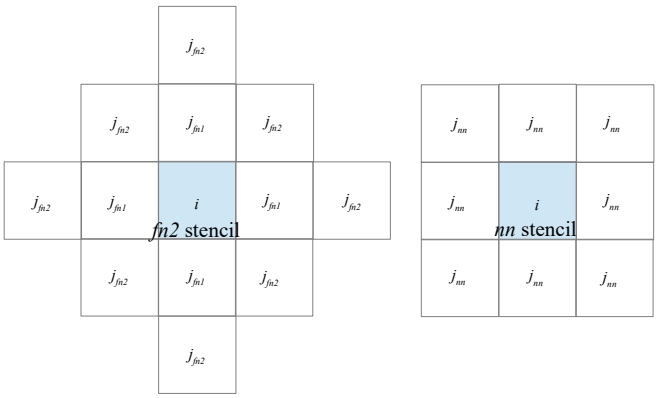


Fig. 3 The *fn2* and *nn* stencils for computing the WLSQ cell-average gradient on a quadrilateral grid.

It has been shown in [14,15] that the set of neighbors, $\{g_j\}$, that define the WSLQ stencil, affects the stability of finite-volume solvers. Ref. [14] formally shows that a finite-volume scheme can be unstable when using the face-neighbor gradient stencil on tetrahedral and hybrid grids, and that adding extra cells to the stencil can cure the instability. Ref. [15] shows that a larger stencil size usually leads to stability. Therefore, this suggests that the most robust stencil should be the vertex stencil on triangular/tetrahedral grids and the union of *nn* and *fn2* stencils on general unstructured grids. However, due to their size, these stencils can noticeably increase the time and memory required to compute the solution to (6), especially in 3-D [17] as well as decreasing the accuracy of the gradients [10]. Recently, Nishikawa [10] explored ways to construct a gradient stencil that achieves robustness and accuracy of the *fn2* and *nn* stencils with a smaller stencil. In [10], Nishikawa proposed two augmentation methods that resulted in a robust, efficient and accurate stencil in 2-D. These methods were the symmetric augmentation of the *fn1* stencil, (*sym*), and the symmetric F-decreasing augmentation of the *fn1* stencil, (*symF*). These stencils, were compared with the *fn1*, *fn2*, and *nn* stencils using a 2-D cell-centered finite-volume Euler solver on four unstructured grids. For a detailed discussion of the test problems and the results obtained, please refer to [10]. The end result of the numerical experiments conducted was that the *symF* stencil was the only stencil that allowed a solution to be obtained for all of the test problems. In the interest of brevity, a brief description of the *sym* and *symF* stencils follows.

2.Construction of the symmetric augmentation stencil, (*sym*)

In [10], the construction of the *sym* stencil, $\{g_i\}_{sym}$, begins with the cells defining the *fn1* stencil, $\{g_i\}_{fn1}$, shown in Fig. 1, for cell *i*, and adds cells to it from the union of the *nn* and *fn2* stencil neighbors $\{v_i\}$ that will symmetrize the stencil as much as possible. The symmetric augmentation begins with one of the face neighbors, $j_{fn1} \in \{g_i\}_{fn1}$, and searches for a cell, $j_{sym} \in \{v_i\}$, located symmetrically opposed with respect to the centroid of the cell *i* as viewed from the centroid of cell j_{sym}. This is repeated to form the set of j_{sym} cells, $\{j_i\}_{sym}$, until the *sym* stencil, $\{g_i\}_{sym}$, is formed from the union of $\{g_i\}_{fn1}$ *and* $\{j_i\}_{sym}$. As shown in Figs. 4 and 5, for triangles and quadrilaterals, respectively, this process is repeated for each cell face neighbor cell until each face neighbor cell has a symmetrically opposed cell that is not a face neighbor. Note that a $\{j_i\}_{sym}$ cannot also be a member of $\{g_i\}_{fn1}$.

3.Construction of the symmetric F-decreasing augmentation stencil, (*symF*)

The *symF* stencil, $\{g_i\}_{symF}$, is constructed using an augmentation of the *sym* stencil based on the magnitude of the gradient. As described in [10], this is accomplished by considering the normal equation:

$$A^T Ax = A^T b \quad , \tag{7}$$

where

$$A^T A=\begin{bmatrix} \sum_{j\in\{g_i\}} w_j^2\Delta x_j^2 & \sum_{j\in\{g_i\}} w_j^2\Delta y_j\Delta x_j & \sum_{j\in\{g_i\}} w_j^2\Delta z_j\Delta x_j \\ \sum_{j\in\{g_i\}} w_j^2\Delta x_j\Delta y_j & \sum_{j\in\{g_i\}} w_j^2\Delta y_j^2 & \sum_{j\in\{g_i\}} w_j^2\Delta z_j\Delta y_j \\ \sum_{j\in\{g_i\}} w_j^2\Delta x_j\Delta z_j & \sum_{j\in\{g_i\}} w_j^2\Delta y_j\Delta z_j & \sum_{j\in\{g_i\}} w_j^2\Delta z_j^2 \end{bmatrix},$$

$$x=\begin{bmatrix} \overline{\partial_x q_i} \\ \overline{\partial_y q_i} \\ \overline{\partial_z q_i} \end{bmatrix}, \quad A^T b=\begin{bmatrix} \sum_{j\in\{g_i\}} w_j^2\Delta x_j\Delta q_j \\ \sum_{j\in\{g_i\}} w_j^2\Delta y_j\Delta q_j \\ \sum_{j\in\{g_i\}} w_j^2\Delta z_j\Delta q_j \end{bmatrix}, \tag{8}$$

and

$$\Delta x_j=x_j-x_i \ , \ \Delta y_j=y_j-y_i \ , \ \Delta z_j=z_j-z_i \ , \ j=1, 2, 3, \cdots, N. \tag{9}$$

Equation (7) is then scaled such that its right hand side is on the order of the typical variation of q over the stencil, i.e., $O(\Delta q_i)$, resulting in,

$$s^{-1} A^T Ax=\tilde{b}, \tag{10}$$

where

$$\tilde{b}=s^{-1} A^T b \ , \text{ and } \ s=\sum_{j\in\{g_i\}} w_j^2 d_j. \tag{11}$$

In [10], it was shown that the magnitude of the gradient is bounded from below and that the lower bound is determined by the measure F defined as

$$F=\frac{s}{\left\| A^T A\right\|_F}, \tag{12}$$

where $\left\| \cdot \right\|_F$ is the Frobenius norm. This measure is used to select cells from $\{v_i\}$ for use in the stencil if their inclusion decreases F thereby decreasing the magnitude of the gradient. The algorithm employing the use of F to construct the *symF* stencil is as follows:

1. Construct the symmetric augmentation stencil.
2. Compute A^TA and s, where $\{g_i\}=\{g_i\}_{sym}$, and then compute F.
3. Let $F_0=F$, $(A^TA)_0=A^TA$, $s_0=s$, and the initial *symF* stencil, $symF_0$, is then *sym* stencil.
4. Let $\{v_i^R\}$ be a subset of cells in $\{v_i\}$ not in the $symF_0$ stencil, and N_R be the number of cells in $\{v_i^R\}$.
5. If $N_R = 0$, no further augmentation is possible, stop, else,
6. for $m=1$ to N_R, perform the following:

   (1) Compute A^TA and s by adding the contributions of the m-th cell at (x_m,y_m,z_m):

$$A^T A=(A^T A)_0+\begin{bmatrix} w_m^2\Delta x_m^2 & w_m^2\Delta y_m\Delta x_m & w_m^2\Delta z_m\Delta x_m \\ w_m^2\Delta x_m\Delta y_m & w_m^2\Delta y_m^2 & w_m^2\Delta z_m\Delta y_m \\ w_m^2\Delta x_m\Delta z_m & w_m^2\Delta y_m\Delta z_m & w_m^2\Delta z_m^2 \end{bmatrix}, \ s=s_0+w_m^2 d_m \ , \tag{13}$$

where

$$\Delta x_m = x_m - x_i \ , \quad \Delta y_m = y_m - y_i \ , \quad \Delta z_m = z_m - z_i \ ,$$

$$d_m = \sqrt{\Delta x_m^2 + \Delta y_m^2 + \Delta z_m^2} \quad \text{and} \quad w_j = \frac{1}{d_j^{p(1)}}. \tag{14}$$

(2) Compute $F = s / \| A^T A \|_F$.

(3) If $F < K_F F_0$, where K_F=0.85, add the m-th cell to $\{symF_i\}$ and set $F_0 = F$, $(A^T A)_0 = A^T A$ and $s_0 = s$.

In [10], the set $\{v_j\}$ was the union of the *nn* and *fn2* stencils, and K_F was chosen to be 0.85 to only accept cells that significantly reduce the stencil F. Some modifications to the algorithm were made when extending it to 3-D and implementing it in VULCAN-CFD. These modifications were as follows:

1. Due to memory considerations, the current algorithm uses the *nn* stencil as the basis of $\{v_j\}$ for tetrahedral, pyramidal and prismatic cells. This modification will have no effect on the stencils constructed for tetrahedral cells because the *fn2* stencil is a subset of the *nn* stencil. However, it will affect the construction of stencils for prismatic and pyramidal cells. This remains an open area of research but testing to date has not shown this modification to be a problem.
2. Hexahedral cells are forced to use the *fn1* stencil. This restriction is based on an examination of the curved, high aspect ratio hexahedral grid numerical experiments presented in [10] where it was shown that the *fn1* stencil had behavior similar to the *symF* stencil even when the *fn2* and *nn* stencils failed.

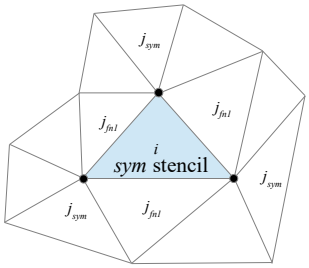


Fig. 4 The *sym* stencil for computing the WLSQ cell-average gradient on a triangular grid.

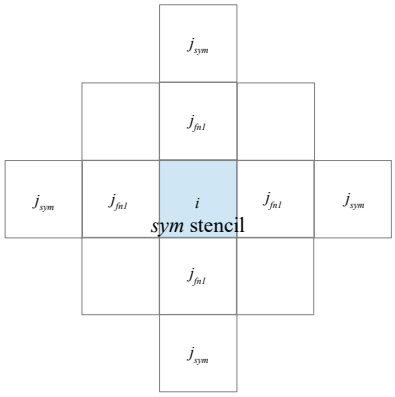


Fig. 5 The *sym* stencil for computing the WLSQ cell-average gradient on a quadrilateral grid.

C. Inviscid Flux Cell Face State Variable Reconstruction

The inviscid fluxes in the unstructured-grid solver are computed using an upwind flux scheme. Currently, either the LDFSS [18] or the HLLC scheme [19] can be selected. Both of these schemes require that the reconstruction variables, q, be specified on the left (L) and right (R) sides of the cell face midpoint, f, as shown in Fig. 6. The reconstruction variables are defined as

$$q = (\frac{\rho_1}{\rho}, \ldots, \frac{\rho_{ncs}}{\rho}, \rho, u, v, w, P, k, \omega) \quad \text{for thermal equilibrium, or}$$

$$q = (\frac{\rho_1}{\rho}, \ldots, \frac{\rho_{ncs}}{\rho}, \rho, u, v, w, T_{ve}, P, k, \omega) \quad \text{for thermal nonequilibrium,}$$

where $\frac{\rho_1}{\rho},\ldots,\frac{\rho_{ncs}}{\rho},\rho,u,v,w,P,T_{ve},k,\omega$ are the chemical species mass fractions, from 1 to the number of species, static density, Cartesian velocity components, vibrational/electronic Temperature, static pressure, turbulent kinetic energy, and specific turbulent dissipation rate, respectively.

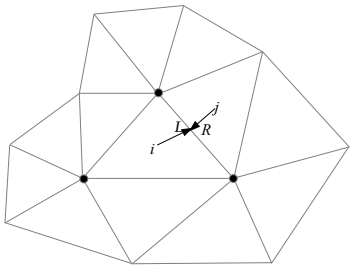


Fig. 6 Higher-order reconstruction of the L and R states to the cell face midpoint.

A 1st-order accurate scheme results when the cell-average values to the left, *i* and right, *j* of the cell face are used. A higher-order accurate scheme results when the *L* and *R* primitive variables are reconstructed to the cell face midpoint with an extrapolation or interpolation method based on the left and right cell-average primitive variables and gradients as given by

$$q_f^L = q_i + \overline{\nabla q_i} \cdot \vec{r}_{if} \tag{14}$$

$$q_f^R = q_j + \overline{\nabla q_j} \cdot \vec{r}_{jf}. \tag{15}$$

where $\vec{r}_{if}$ and $\vec{r}_{jf}$ are the vectors shown in Fig. 6. In addition to the scheme above, which is an unstructured-grid interpretation of Fromm's scheme [20], the higher-order variable extrapolation (or UMUSCL) reconstruction scheme [21] was also implemented to control the dissipation of the scheme further. The UMUSCL scheme can be written as

$$q_f^L = q_i + \frac{\chi}{2}(q_j - q_i) + (1-\chi)\overline{\nabla q_i} \cdot \vec{r}_{if} \tag{16}$$

$$q_f^R = q_j + \frac{\chi}{2}(q_i - q_j) + (1-\chi)\overline{\nabla q_j} \cdot \vec{r}_{jf} \tag{17}$$

where χ is used to control the behavior and the 1-D order of accuracy of the scheme when the flow is smooth.

1. χ = 0, gives Fromm's scheme
2. χ = -1, gives a 2nd-order fully upwind MUSCL-type scheme
3. χ = 1/3, gives a 3rd-order upwind biased MUSCL-type scheme

D. Inviscid Flux Cell-Average Gradient Limiter Construction

When computing hypersonic flow, discontinuities will usually exist somewhere in the computational domain. In the vicinity of these discontinuities, the higher-order reconstruction of the state variables to the cell face used to achieve 2nd-order accuracy of the inviscid flux scheme will produce oscillations in the flow solution, and eventually cause the computation to fail. These oscillations can be suppressed by locally forcing the reconstruction to be 1st-order through the use of some sort of gradient limiter. A gradient limiter can be implemented in two different ways for the UMUSCL scheme; a 1-D "face"-based limiter approach or a multidimensional "stencil"-based limiter approach. A face-based limiter for the UMUSCL scheme can be written as

$$\widetilde{q_f^L} = q_i + \chi\widetilde{\beta^L} + (1-\chi)\widetilde{\alpha^L} \tag{18}$$

$$\widetilde{q_f^R} = q_j + \chi\widetilde{\beta^R} + (1-\chi)\widetilde{\alpha^R} \tag{19}$$

where the limited left and right gradients, $\widetilde{\alpha^{L,R}}$ and $\widetilde{\beta^{L,R}}$ are

$$\widetilde{\alpha^{L,R}} = F(\alpha^{L,R}, \beta^{L,R})_{(limiter)} \quad \text{and} \quad \widetilde{\beta^{L,R}} = F(\beta^{L,R}, \alpha^{L,R})_{(limiter)} \quad ,$$

the unlimited left and right 1-D gradients $\alpha^{L,R}$ and $\beta^{L,R}$ are

$$\alpha^L = \overline{\nabla q_i} \cdot \vec{r}_{if} \quad \text{and} \quad \beta^L = \frac{1}{2}(q_j - q_i) \tag{20}$$

$$\alpha^R = \overline{\nabla q_j} \cdot \vec{r}_{jf} \quad \text{and} \quad \beta^R = \frac{1}{2}(q_i - q_j) \tag{21}$$

and, in the case of the van Leer limiter [22]

$$F(\Delta_1,\Delta_2)_{(van\ Leer)}=\frac{(\Delta_2|(\Delta_1)|+\Delta_1|(\Delta_2)|)}{(|(\Delta_1)|+|(\Delta_2)|+\epsilon)} \tag{22}$$

where ϵ is on the order of $1.0\text{x}10^{-12}$. While the face-based scheme has been found to be reasonably effective on smooth hexahedral grids, its effectiveness deteriorates on truly unstructured grids. Therefore, two stencil-based limiter approaches have been implemented. The first method is a generalization of the approach used to form stencil-based gradient limiters by Barth and Jesperson [23] and later by Venkatakrishnan [24]. The second method is the multidimensional limiter process (MLP) of Park and Kim [25]. Both of these approaches are referred to as stencil-based limiters herein because they use information from all of the cells that make up the stencil that was used to compute the cell-average gradient. Figure 7 presents the cells involved in the *fn2* stencils used to compute the limiter coefficients in cells *i* and *j*, where the cells labeled i_s and j_s only participate in the *i* and *j* cell stencils, respectively, and cells labeled i_s,j_s are cells that participate in both cell stencils. These stencil-based limiter approaches compute cell-limiter coefficients that are used to limit the higher-order reconstruction that, when applied to the UMUSCL higher-order reconstruction scheme, results in equations for the left and right states having the form

$$\widetilde{q_f^L}=q_i+\Phi_i(q_i)[\frac{\chi}{2}(q_j-q_i)+(1-\chi)\overline{\nabla q}_i\ \cdot\ \vec{r}_{if}] \tag{23}$$

$$\widetilde{q_f^R}=q_j+\Phi_j(q_j)[\frac{\chi}{2}(q_i-q_j)+(1-\chi)\overline{\nabla q}_j\ \cdot\ \vec{r}_{jf}] \tag{24}$$

where $\Phi_i(q_i)$ and $\Phi_j(q_j)$ are the cell-limiter coefficients that are used to limit the reconstruction consistently for all faces of the cells *i* and *j,* respectively. Figure 8 presents the *fn2* stencil cells that participate in the computation of the cell *i* limiter coefficient, $\Phi_i(q_i)$. The methods of [23] and [24] compute the cell limiter coefficients for each cell, *i* using

$$\Phi_i(q_i^{fn2})=min\Big(1,\ \begin{cases}\phi_f(\dfrac{q_i^{max(fn2)}-q_i}{q_f-q_i}), \text{if}\ \ (q_f-q_i)>0\\ \phi_f(\dfrac{q_i^{min(fn2)}-q_i}{q_f-q_i}), \text{if}\ \ (q_f-q_i)<0\\ 1 \qquad\qquad\quad \text{if}\ \ (q_f-q_i)=0\end{cases}\Big),\ f=1\rightarrow N_{i,faces}, \tag{25}$$

where $q_i^{max(fn2)}$ and $q_i^{min(fn2)}$ are the maximum and minimum values of *q* of the *fn2* stencil cells, respectively, and q_f is computed at each cell face midpoint by using the unlimited form of UMUSCL

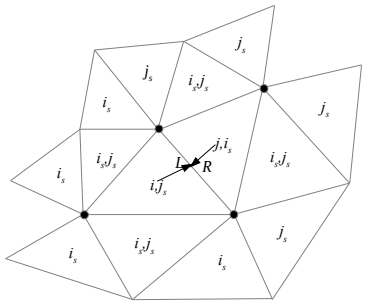


Fig. 7 The cells that participate in the construction of the *fn2* stencil-based limiter coefficients used to reconstruct data to the face shared by cells *i* and *j*.

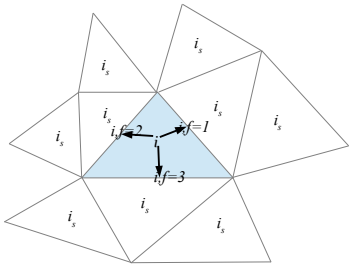


Fig. 8 The *fn2* stencil cells and face midpoints that participate in the construction of the stencil-based limiter coefficients for cell *i* where $N_{i,faces}$ = 3.

$$q_f = q_i + \frac{\chi}{2}(q_j - q_i) + (1-\chi)\overline{\nabla q_i} \cdot \vec{r}_{if} \tag{26}$$

where the value of χ is consistent with the value used in Eq. (15). The face-limiter coefficient, ϕ_f, is computed using a generalization of the form used in [23] and [24]

$$\phi_f\left(\frac{b}{a}\right) = \frac{F(a,b)_{(limiter)}}{a} \tag{27}$$

where, $F(a,b)_{(limiter)}$, can be any limiter function found in the literature. For example, [23] used the minmod limiter function whereas [24] used a modified form of the van Albada limiter function [26]. Currently VULCAN-CFD allows the use of the Sweby [27], van Leer [22], van Albada [26], Venkatakrishnan [24], and Koren [27] limiter functions consistent with the structured-grid solver. However, while we have found that this stencil-based limiter approach improves the shock capturing capability of the code significantly compared to the face-based limiter approach, some oscillations can still occur in the vicinity of very strong shocks. These flows require further augmentation via a heuristic pressure limiter such as proposed by Gnoffo [28].

Adding heuristic limiters, such as the aforementioned pressure limiter, has the potential to adversely affect the flow solution by adding too much dissipation where it is not needed, e.g., in shock boundary layer interaction flows where the physical viscosity should prevent oscillations from occurring such that the pressure limiter is no longer needed. Therefore, as mentioned previously, the multidimensional limiter procedure (*MLP*) of Park and Kim [25] has also been implemented in an attempt to further improve the discontinuity capturing capability and robustness of the unstructured-grid solver near strong shocks, while reducing limiting and thus dissipation in the vicinity of shock boundary layer interactions. The *MLP* limiter is a stencil-based limiter that also attempts to "define and implement monotinicity in multiple dimensions" by strictly enforcing the maximum principle at the corners of the cell. Park and Kim state that the central premise of *MLP* is to "control the distribution of both cell center and cell node physical properties to mimic the multidimensional nature of the flow physics." They state that this can be accomplished based on the observation that a well controlled reconstruction of the cell-centered solution to the nodes can be used to construct a limiting process that is both multidimensional and monotone. For a detailed discussion of the mathematical proof of this concept, see [25].

The implementation of the *MLP* approach proceeds in a manner similar to that of the previously described stencil-based limiting approach with the key difference being that the construction of the cell limiter coefficient uses a reconstruction of the solution to the cell nodes instead of to the cell face midpoints. Figure 9 illustrates the *nn* stencil cells and nodes that participate in the computation of the cell *i, MLP*$_{nn}$ "node-based" limiter coefficient, $\Phi_i(q_i^{MLP_{nn}})$.

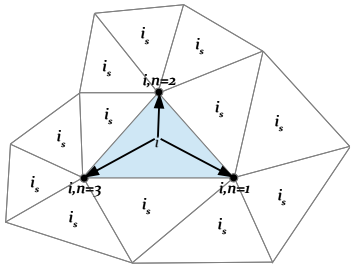


Fig. 9 The *nn* stencil cells and nodes that participate in the construction of the *MLP* ^{nn} limiter coefficients for cell *i* where $N_{i,\,nodes} = 3$.

This results in Eq. (25) being split into two equations

$$\Phi_i(q_i^{MLP_{nn}}) = min\left(1, [\Psi_{i,n}^{nn}(q_{i,n}), n = 1 \rightarrow N_i, nodes]\right) \tag{28}$$

and

$$[\Psi_{i,n}^{nn}(q_{i,n}) = \begin{cases} \phi_{i,n}\left(\dfrac{q_{i,n}^{max(ncn)} - q_i}{q_{i,n} - q_i}\right), \text{if } (q_{i,n} - q_i) > 0 \\ \phi_{i,n}\left(\dfrac{q_{i,n}^{min(ncn)} - q_i}{q_{i,n} - q_i}\right), \text{if } (q_{i,n} - q_i) < 0 \\ 1 \qquad\qquad\qquad \text{if } (q_{i,n} - q_i) = 0 \end{cases}], \tag{29}$$

where $\Phi_i(q_i^{MLP_{nn}})$, is computed in two steps: 1) the cell node limiter coefficient, $\Psi_{i,n}^{nn}(q_{i,n})$, is computed at each node, *n,* that is a vertex of cell *i,* using Eq. (29), 2) the cell-limiter coefficient is computed as the minimum of those cell node limiter coefficients using Eq. (28). In Eq. (29), the quantities, $q_{i,n}^{max(ncn)}$ and $q_{i,n}^{min(ncn)}$ are the maximum and minimum values of the node cell-neighbor stencil, *ncn*, illustrated in Fig. 10, and $q_{i,n}$ is the reconstruction of *q* to each node, *n*, of cell, *i*, based on an unlimited form of Fromm's scheme, i.e.,

$$q_{i,n} = q_i + \overline{\nabla q}_i \cdot \vec{r}_{in} \tag{30}$$

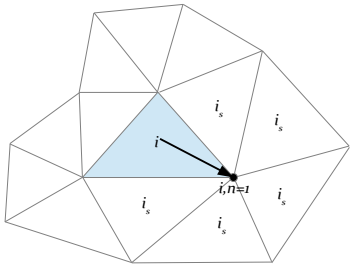


Fig. 10 The node cell-neighbor stencil of cells sharing the n=1 node used to compute $q_{i,n}^{max(ncn)}$ and $q_{i,n}^{min(ncn)}$.

1. where $\phi_{i,n}$, is computed at each node, *n,* of cell *i*, by using Eq. (29), instead of at each face midpoint, *f*, of cell *i*. However, since the cell limiter coefficient, $\Phi_i(q_i^{MLP_{nn}})$, was derived for use with the *nn* stencil, it must be modified for use with a general stencil $\{g_j\}$ that is a subset of the *nn* stencil, such as the *fn2* or the *symF* stencils, $\Phi_i(q_i^{MLP_{\{g_j\}}})$, as illustrated in Fig. 11,

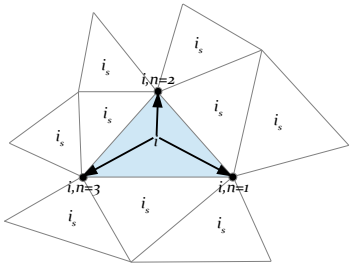


Fig. 11 The fn2 stencil cells and nodes that participate in the construction of the $\Phi_i(q_i^{MLP_{\{g_j\}}})$, cell-limiter coefficients for the cell i where $N_{i,\,nodes} = 3$.

where the terms $q_{i,n}^{max(ncn)}$ and $q_{i,n}^{min(ncn)}$ of Eq. (29) have been replaced with $q_i^{max(\{g_j\})}$ and $q_i^{min(\{g_j\})}$, respectively, to yield

$$\Phi_i(q_i^{MLP_{\{g_j\}}}) = min(1, [\Psi_{i,n}^{\{g_j\}}(q_{i,n}), n=1 \rightarrow N_i, nodes]) \tag{31}$$

and

$$[\Psi_{i,n}^{\{g_j\}}(q_{i,n}) = \begin{cases} \phi_{i,n}\left(\dfrac{q_i^{max(\{g_j\})} - q_i}{q_{i,n} - q_i}\right), \text{if } (q_{i,n} - q_i) > 0 \\ \phi_{i,n}\left(\dfrac{q_i^{min(\{g_j\})} - q_i}{q_{i,n} - q_i}\right), \text{if } (q_{i,n} - q_i) < 0 \\ 1 \qquad\qquad\qquad \text{if } (q_{i,n} - q_i) = 0 \end{cases}]. \tag{32}$$

The resulting gradient limiter approach of Eq.s (31-32), $\Phi_i(q_i^{MLP_{\{g_j\}}})$ while not a strict implementation of the $\Phi_i(q_i^{MLP_{nn}})$ limiter described in Eq.s (28-30), has been found to be much more robust than the face-based limiter originally implemented and less dissipative then the heuristic pressure limiter augmented stencil-based limiter approach of Eq.s (25-27). In addition, we have replaced the Venkatakrishnan limiter with the MLP-u2 limiter of [25]. The MLP-u2 limiter is a modified form of Venkatakrishnan's limiter. Venkatakrishnan's limiter has the form

$$\phi_{i,n}\left(\frac{\Delta_2}{\Delta_1}\right)=\frac{1}{\Delta_1}\left[\frac{(\Delta_2^2+\epsilon^2)\Delta_1+2\Delta_1^2\Delta_2}{\Delta_2^2+2\Delta_1^2+\Delta_1\Delta_2+\epsilon^2}\right] \tag{33}$$

where

$$\Delta_1=q_i^{min\,or\,max}-q_i,\ \ \Delta_2=\overline{\nabla q}_i\cdot\vec{r}_{in}\ \text{ and }\ \epsilon^2=(K_L|\vec{r}_{in}|)^3 \tag{34}$$

where K_L is a O(1) user definable constant that should be "tuned" for each computation. Park and Kim construct the MLP-u2 limiter by retaining Venkatakrishnan's limiter function and redefining ϵ^2 to be

$$\epsilon^2=\frac{K_1}{1+\theta}\Delta q_{i,n}^2\ \text{ where }\ \Delta q_{i,n}=q_{i,n}^{max}-q_{i,n}^{min},\ \theta=\frac{\Delta q_{i,n}}{K_2|\vec{r}_{in}|^{K_3}}\ \text{ and }\ K_1=5,\ K_2=5,\ K_3=1.5. \tag{35}$$

We have found this limiter to be robust, well behaved and relatively insensitive to the K_1, K_2, and K_3 parameters. A final limiter modification that has also been found to improve robustness when computing complex hypersonic flows using the *fn2, nn* or the *symF* stencil is to compute the *MLP* limiter coefficient as the minimum of the pressure limiter coefficient and each of the primitive variable limiter coefficients, i.e.,

$$\Phi_i(q_i^{MLP_{(g_j)}})=min\left[\Phi_i(P_i^{MLP_{(g_j)}}),\Phi_i(q_i^{MLP_{(g_j)}})\right] \tag{36}$$

E. Viscous Flux Cell Face Gradient Construction

The computation of the viscous flux requires that the cell-face average and the cell-face average gradient of the primitive variables, $\overline{\nabla q}_f$, be computed, where the primitive variables are,

$$q\ =\ (\frac{\rho_1}{\rho},\ldots,\frac{\rho_{ncs}}{\rho},\rho,u,v,w,T,k,\omega)\quad\text{for thermal equilibrium, or}$$

$$q\ =\ (\frac{\rho_1}{\rho},\ldots,\frac{\rho_{ncs}}{\rho},\rho,u,v,w,T_{ve},T,k,\omega)\quad\text{for thermal nonequilibrium}$$

where T and T_{ve} are the static temperature and vibrational/electronic temperature, respectively. Hasselbacher [30] observed that computing $\overline{\nabla q}_f$ as a simple average of the face neighbor cell-average gradients, i.e.,

$$\overline{\nabla q}_f=\frac{(\overline{\nabla q}_i+\overline{\nabla q}_j)}{2} \tag{37}$$

leads to odd-even decoupling causing him to introduce face-derivative augmentation. Hasselbacher suggested two methods to accomplish this augmentation: the so-called, edge-normal (*EN*) and face-tangent (*FT*) cell-face gradient methods. More recently, Nishikawa [31] proposed a new approach to deriving a scheme for a second-derivative diffusion term, where a diffusion scheme is derived from an advection scheme applied to a hyperbolic diffusion model. As demonstrated in Ref. [31], the use of the hyperbolic model leads to robust diffusion schemes for various discretization methods, including finite-volume and finite-element methods. The resulting finite-volume diffusion scheme has a consistent approximation term and an adjustable high-frequency damping term with a coefficient alpha, and thus is often called the alpha-damping diffusion scheme. It is a generalized finite-volume diffusion scheme that includes the edge-normal and face-tangent methods as special cases. The edge-normal augmented cell-face gradient method is the method originally implemented in the NRA-supplied code. However, when the edge-normal and face-tangent augmented cell-face gradient methods were studied in [11,31,32], the face-tangent method was found to be preferable to the edge-normal method. Moreover, in [11], the observation was made that, in many cases, a converged solution could only be obtained when the face-tangent augmented face-gradient method was used. However, insight can be gained by comparing the edge-normal, face-tangent and hyperbolic-reconstruction cell-face gradient methods. Therefore, for the sake of completeness, the edge-normal method is also described. The edge-normal augmented cell-face gradient method, as defined by Hasselbacher, is

$$\widehat{\nabla q}_f^{EN}=\overline{\nabla q}_f-[\overline{\nabla q}_f\cdot\hat{e}_{ij}-\frac{(q_j-q_i)}{|\vec{e}_{ij}|}]\ \hat{e}_{ij} \tag{38}$$

where, referring to Fig. 1, $\vec{e}_{ij}$ is a vector drawn from cell-center *i* to cell-center *j* and $\hat{e}_{ij}$ is its unit vector. The face-tangent augmented cell-face gradient method, as defined by Hasselbacher, is

$$\widehat{\nabla q_f}^{FT} = \overline{\nabla q_f} - [\overline{\nabla q_f} \cdot \hat{e}_{ij} - \frac{(q_j - q_i)}{|\vec{e}_{ij}|}](\frac{\hat{n}_f}{\hat{n}_f \cdot \hat{e}_{ij}}). \tag{39}$$

As mentioned above, Nishikawa considered the augmentation terms (the bracketed terms in Eq. (38) and (39) to be damping terms, and further observed that the face-tangent method damping term leads to a more robust scheme on highly skewed meshes due to the dependence on $1/(\hat{n}_f \cdot \hat{e}_{ij})$. The increased robustness results from the fact that as skewness increases, $\hat{n}_f \cdot \hat{e}_{ij}$ decreases, thereby increasing damping. When Nishikawa's hyperbolic diffusion-based approach is applied to a cell-centered, finite-volume scheme, it results in a reconstruction-based cell-face average gradient, $\widehat{\nabla q_f}^{HR}$, that includes a damping term that arises naturally due to an upwind method being used to discretize the construction of the cell-face average gradient. This reconstruction based cell-face average gradient method, i.e., the alpha-damping scheme, has the form:

$$\widehat{\nabla q_f}^{HR} = \overline{\nabla q_f} + \alpha(\frac{\hat{n}_f}{|\vec{e}_{ij} \cdot \hat{n}_f|})\ (q_f{}^R - q_f{}^L) \tag{40}$$

where α is a damping coefficient and $q_f{}^L$ and $q_f{}^R$ are the left and right higher-order-reconstructed viscous face state variables. These state variables are reconstructed using Fromm's scheme where

$$q_f{}^L = q_i + \overline{\nabla q_i} \cdot \vec{r}_{if} \tag{41}$$

$$q_f{}^R = q_j + \overline{\nabla q_i} \cdot \vec{r}_{jf}. \tag{42}$$

The first term in Eq. (40) is the consistent term approximating the face gradient, and the second term is the adjustable damping term. Ref. [31] shows that $\alpha = 4/3$ leads to fourth-order accuracy on regular quadrilateral grids. Observe that the alpha-damping scheme reduces to the *FT* method if the reconstruction is performed halfway between the two centroids across the face (not necessarily on the face); and the absolute sign is removed from the skewness measure in the denominator. Recently, Jalai et al. [32] analyzed the stability and accuracy of nine cell-centered, finite-volume, cell-face gradient methods and reported that Nishikawa's scheme, with $\alpha = 4/3$, was the preferred scheme for computing the cell-face gradient, based on stability and accuracy analyses.

III. Verification

A 2nd-order accurate cell-centered scheme requires derivatives that are 1st-order accurate. Therefore, we seek to verify the relative accuracy and order of accuracy of the linear least-squares cell gradient operator for each of the candidate stencils, *fn2, nn* and, *symF.* For completeness, we also verify the relative accuracy and order of accuracy of the linear least-squares cell gradient operator for the *fn1* stencil. These experiments were accomplished through the use of a 2-D cell-centered code developed to compute the cell-average gradient of a variable *u(x,y)* using a weighted linear least-squares method with a weighting parameter, *p(l)*, of *0.25*, on severely perturbed triangle- and quadrilateral-based unstructured grids around a 2-D cylinder for the function

$$u(x, y) = \sin(0.1r) + 0.5\sin(\theta). \tag{43}$$

The baseline triangle and quadrilateral perturbed grids, shown in Fig. 12a and Fig. 13a, respectively, were created by perturbing the nodes of a regular grid, and, in the case of the triangle based grid, swapping the edges. Close up views of the grids near the inner boundary, presented in Figs. 12b and 13b, illustrate the level of distortion achieved. Four finer grids were then generated by repeating the perturbation process on grids that were refinements of the baseline grid. Equation (43) was then used to compute *u* at each cell center. The linear least-squares cell-average gradient was then computed for each cell of the grid using the *fn1*, *fn2, nn* and *symF* stencils and the gradient error was computed as the difference between the analytical value and the linear least-squares value. Figures 14 and 15 present the gradient L1 error as a function of the effective cell size for each of the stencil types for the interior cells, i.e., cells that do not have a face on a boundary. Figure 14 demonstrates that all stencils types produce linear least-squares cell-average gradients that are 1st-order accurate and have similar relative error on the triangle-based grid. Figure 15a demonstrates that all stencil types produce linear least-squares gradients that are 1st-order accurate. Fig. 15b shows that, on a quadrilateral grid, that the *fn1* and *nn,* and *fn2* have similar accuracy as the grid is refined. Figure 15b also shows that the *symF* stencil is slightly less accurate than the *fn1*, *fn2*, and *nn* stencils.

Stencil size statistics for the *fn1*, *fn2, nn* and *symF* interior cell stencils for the triangle-based grid shown in Fig. 12 are presented in Fig. 16a. Based on the definitions of the *fn1* and *fn2* stencils shown in Fig. 1 and 2, as expected, Fig. 16a indicates the *fn1* stencil has a minimum, maximum and mean stencil size of 3 for all grids. Figure 16a also shows that the *fn2* stencil has a minimum stencil size of 6, a maximum stencil size of 9 and a mean stencil size of

essentially 8 for all grids. The minimum *fn2* stencil size is 6 due to a lack of ghost cells on the boundaries. It is important to note, that the size of these face-neighbor-based stencils are driven by the cell topology and as such when the cell topology is constant, the cell stencil statistics will be constant. In other words, as shown in Figs. 2 and 3, a grid made up of cells having 3 faces will have a *fn1* stencil size of 3 and an *fn2* stencil of 9 and a grid made up of cells having 4 faces will have a *fn1* stencil size of 4 and a *fn2* stencil of 12. However, Fig. 16a also reveals that the *nn* stencil has a minimum stencil size of 6, a maximum stencil size of 24-28, and a mean stencil size of 12-14. It is very important to note that the increase in *nn* stencil size is due to the size of this type of stencil being driven by node connectivity instead of cell topology. Moreover, since node connectivity is only a constant for structured grids or regular unstructured grids, the *nn* stencil size will vary spatially as a function of the node connectivity for all but the simplest unstructured grids. As shown in Fig. 2, for a 2-D unstructured triangle-based grid, the *nn* stencil will always be bigger than the *fn2* stencil. Figure 2 also shows that for this particular hypothetical grid, the *fn2* stencil has 9 members and the *nn* stencil has 12 members. Moreover, as illustrated in Fig. 16a, the *nn* stencil can be significantly bigger then the *fn2* stencil where the grid connectivity is irregular/complex. However, Fig. 16a shows that the *symF* stencil has a minimum size of 4, a maximum size of 9 and a mean size of 6. Therefore, in 2-D on this particular grid, the *symF* stencil is approximately the same size as the *fn2* stencil and nearly a factor of 3 smaller than the *nn* stencil. Therefore, in 3-D, where tetrahedral, prismatic and pyramidal cells will also have a *nn* stencil size that is determined by node connectivity, the *fn1* and *fn2* stencils will be, and the *symF* stencil should be, significantly smaller than the *nn* stencil. These observations have important implications with regard to stencil size behavior on general unstructured grids, especially in 3-D, where stencils sizes in excess of 100 cells have been observed. This is why it is preferable to use the *fn1*, *fn2* or *symF* stencils over the *nn* stencil for a general unstructured grid.

Stencil size statistics for the *fn1*, *fn2, nn* and *symF* interior cell stencils for the quadrilateral-based grid shown in Fig. 13 are presented in Fig. 16b. Consistent with the *fn1*, *fn2* and *nn* stencil definitions shown in Fig. 3, the *fn1*, *fn2* stencil and *nn* maximum stencil sizes shown in Fig. 16b are 4, 8 and 12, respectively. The minimum and mean stencil sizes are different from the maximum size due to a lack of boundary ghost cells. The extension of the 2-D definition of the *fn1*, *fn2* and *nn* stencils to 3-D results in their stencils being 8, 24 and 26 cells, respectively. Figure 16b also shows that the maximum *symF* stencil size was 8, which is the same as the maximum of the nn stencil. This means that the augmentation process used in the *symF* stencil construction "recovered" the *nn* stencil for some of the cells in the quadrilateral grid. It should be noted that for quadrilateral and hexahedral cells the *fn1*, *fn2*, *nn* and *symF* maximum stencil size are determined by cell topology, not node connectivity, and are therefore always bounded.

The end result of these observations regarding stencil size and accuracy on 2-D grids and the extension to 3-D is that, due to accuracy and cost, we always want to use the smallest possible stencil that is stable. Moreover, the *fn1, fn2,* and *symF* stencils are preferable from a parallel processing point of view because they have less stencil variability thereby making the cost of the least-squares more homogeneous across the processors. This would naturally lead one to decide to use the *fn1* stencil wherever possible. However, it has also been observed that:

1. For tetrahedral cells, the *fn1* stencil must be augmented in some manner [13,14,15].
2. For triangular grids, the *symF* stencil was the only stencil that was stable for all the numerical tests performed in [10].
3. For quadrilateral cells, the *fn2* and *nn* stencils can be unstable on high aspect ratio highly curved grids and the *fn1* and *symF* stencils were the only stencils that were stable for all the numerical tests performed in [10].
4. For distorted quadrilateral cells, the *symF* stencil is less accurate than the *fn1* stencil, as shown in Fig. 15.

Therefore, we propose the following strategy for the construction of stencils for a general 3-D grid made up of tetrahedral, pyramidal, prismatic and hexahedral cells:

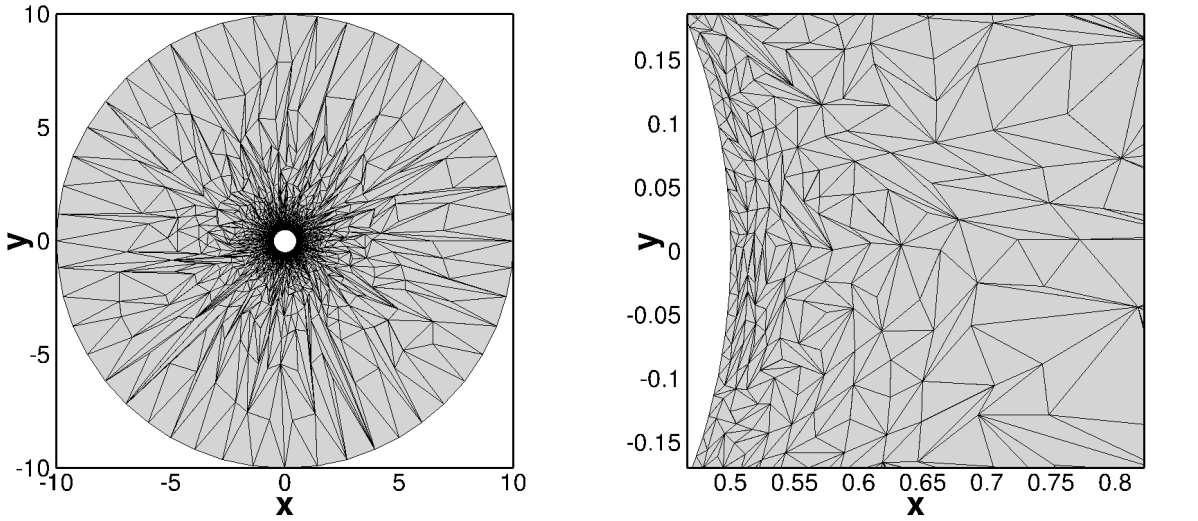
1. Use the modified *symF* stencil proposed in section II.B.2 for tetrahedral cells due to its robustness, cost and accuracy.
2. Use the modified *symF* stencil proposed in section II.B.2 for prismatic and pyramidal cells due to its cost.
3. Use the *fn1* stencil for hexahedral cells due to its simplicity, robustness, cost and accuracy.

In the following section, numerical experiments will be conducted to examine the proposed stencil construction strategy utilizing three hypersonic turbulent flow canonical problems:

1. 2-D hypersonic turbulent flow of air over a backward facing step using a 2-D grid made up of prismatic and hexahedral cells.
2. 2-D hypersonic turbulent flow of air over a flat plate using a 3-D grid made up of pyramidal, prismatic and tetrahedral cells.

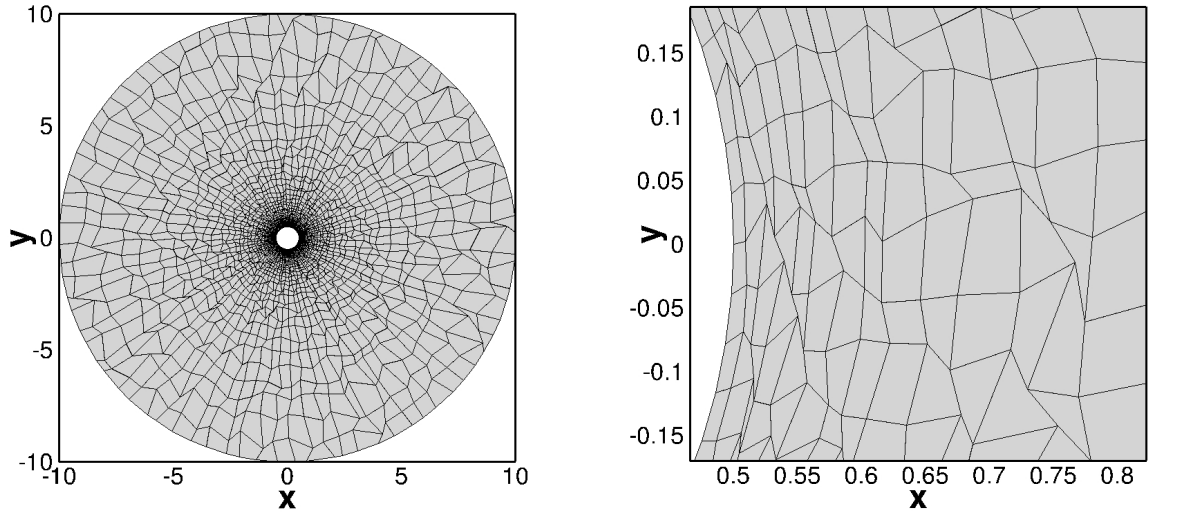
3. 2-D hypersonic turbulent flow of air over a blunt wedge using a 3-D grid made up of tetrahedral, pyramidal, prismatic and hexahedral cells.

The convergence behavior, relative cost, skin friction and heat transfer of the proposed stencil construction method will by compared to results obtained using the *fn2* and *nn* stencils on the same grid.



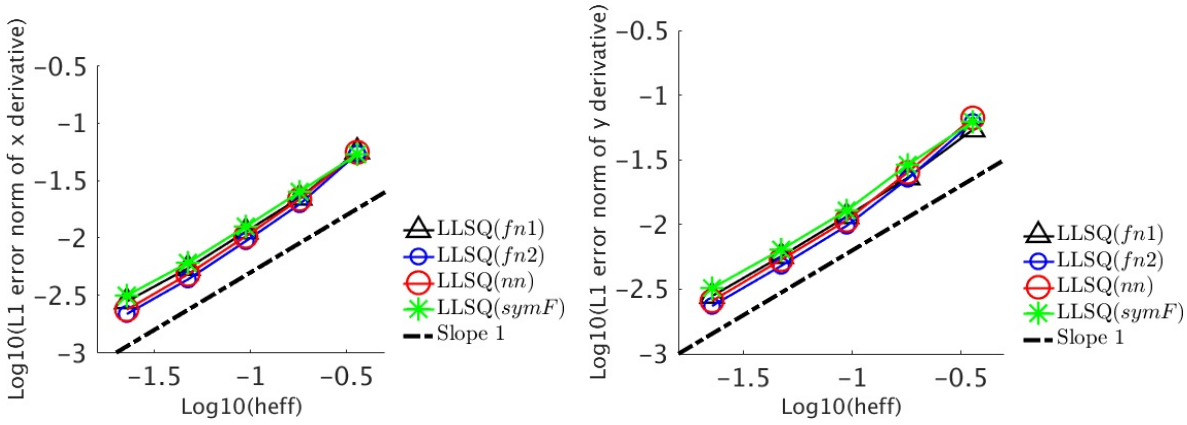
a) Overall view **b) Close up view**

Fig. 12 A very bad triangle-based unstructured grid for the verification of the accuracy of interior weighted linear least-squares cell-average gradients using the *fn1*, *fn2, nn* and *symF* stencils.



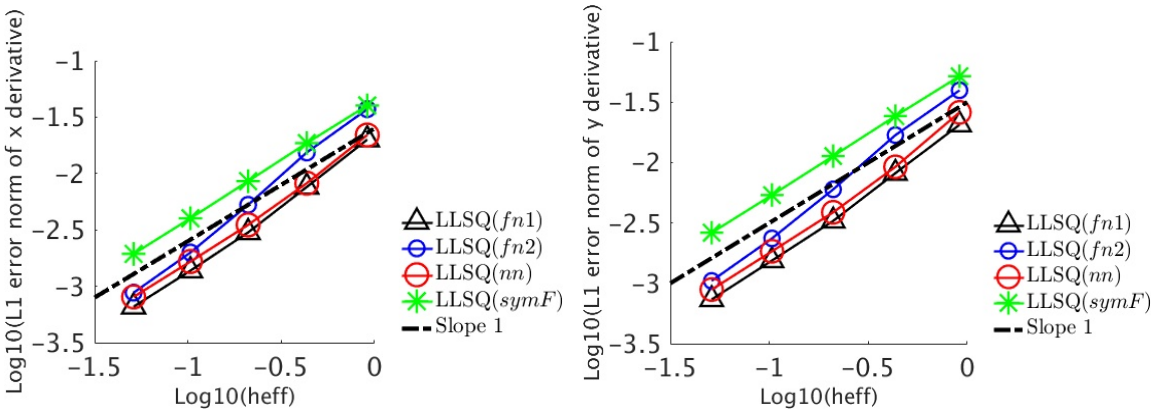
a) Overall view **b) Close up view**

Fig. 13 A very bad quadrilateral-based unstructured grid for the verification of the accuracy of interior weighted linear least-squares cell-average gradients using the *fn1*, *fn2, nn* and *symF* stencils.



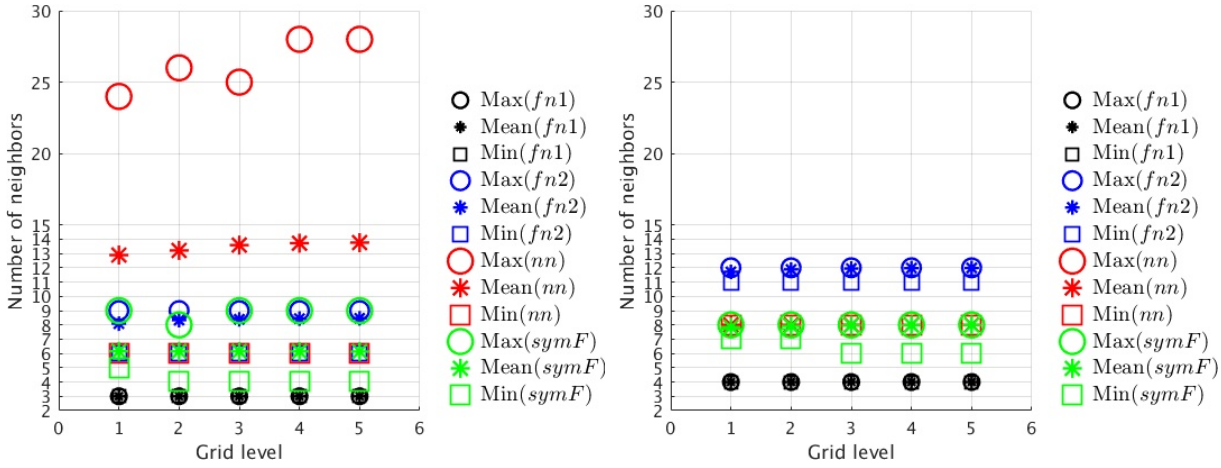
a) $\partial u/\partial x$ **b)** $\partial u/\partial y$

Fig. 14 The accuracy of the interior weighted linear least-squares cell-average gradients using the *fn1*, *fn2, nn* and *symF* stencils on the triangle-based unstructured grid.



a) $\partial u/\partial x$ **b)** $\partial u/\partial y$

Fig. 15 The accuracy of the interior weighted linear least-squares cell-average gradients using the *fn1*, *fn2, nn* and *symF* stencils on the quadrilateral-based unstructured grid.



a) Triangle-based grid **b) Quadrilateral-based grid**

Fig. 16 Stencil size statistics of the *fn1*, *fn2, nn* and *symF* interior cell stencils on the triangle- and quadrilateral-based grids.

IV. Numerical Results and Discussion

A. Hypersonic Turbulent Flow Over a 2-D Backward Facing Step Using Prismatic and Hexahedral Cells.

The first numerical experiment was conducted by computing hypersonic calorically perfect, chemically frozen, turbulent flow of air over a 2-D backward facing step with freestream conditions of, Mach 6.356, static pressure, P_{ref} = 50,662.58 Pascals, static temperature, T_{ref} = 1297.75 Kelvin, ratio of specific heats, γ_{ref} = 1.4, and unit Reynolds number, Re_{ref} = 1.2891x10^7/m, with the wall treated as isothermal (1172.6 Kelvin), using a turbulent wall matching boundary condition [34]. The Wilcox (2006) $k-\omega$ two-equation turbulence model [35] was used to compute the Reynolds stresses and Reynolds heat flux (Pr_t=0.9) and the turbulence model production term was based on the magnitude of the vorticity. The cell-average gradients were computed using weighted linear least-squares with the *fn1*, *fn2*, *nn*, and *symF* stencils. The inviscid fluxes were computed using the HLLC scheme with the higher-order cell-face states constructed using UMUSCL, $\kappa = 1/3$, with the cell-average gradients limited using the $\Phi_i(q_i^{MLP_{|s,|}})$ gradient limiter and the Park and Kim MLP-u2 limiter function. The viscous fluxes were computed using the Nishikawa cell face gradient method. The governing equations were solved implicitly using the Symmetric Gauss-Seidel (SGS) scheme described in [9], with local time stepping and the CFL number linearly varied from 0.1 to 250 over time steps 1 to 500. Convergence was achieved by "freezing" the gradient limiter after 15000 time steps to prevent convergence stalling due to limiter "ringing". The computations were stopped when the residual L_2 norm had dropped 6 orders of magnitude from its initial value. The 2-D geometry was discretized to form a 3-D computational domain using the Pointwise® unstructured grid generator. The resulting grid consisted of triangular and quadrilateral 2-D cells, as shown in Fig. 17, extruded in the Z-direction to form a 3-D grid of 15,781 prismatic and 8,168 hexahedral cells for a total of 23,949 cells. The boundary conditions were: 1) reflection of all variables at the min. and max. Z-direction boundary cell faces (Symmetry Boundary), 2) specification of all variables on the min. X-direction boundary cell faces (Inflow Boundary), 3) 1st-order extrapolation of all variables at the max. X and Y-direction boundary cell faces (Outflow Boundary) and 4) isothermal no-slip wall-matching construction of all variables on the min. Y-direction wall boundary cell faces (No-slip Isothermal Wall). The computations were performed using parallel processing on 6 partitions. A computation was performed using each stencil type with all other input parameters being unchanged. For each computation, the stencil statistics, convergence behavior, contour plots of the flow solution and the X distribution of wall heat transfer were extracted and used to compare the *fn1*, *fn2*, *nn* and *symF* stencils.

Table 1 presents the stencil statistics, i.e., the min., max., and mean stencil sizes as well as the standard deviation, σ, of the stencil size and the relative augmentation cost. This grid, due to its quasi 2-D nature, and due to it consisting of hexahedral and prismatic cells, has statistics such that the min. stencil size has been determined by the topology of the hexahedral cells and the max. stencil size has been determined by the connectivity of the prismatic cells. The min. stencil size of all 3 stencil types are smaller than expected based on a 3-D extension of Fig. 3 for a hexahedral cell due to a boundary effect. This effect is caused by the exclusion of all boundary ghost cells from the no-slip boundary adjacent hexahedral cell stencils. The key point of Table 1 is that the *symF* stencil is both smaller and varies less over the computational domain than the *fn2* and *nn* stencils by a significant amount. The relative augmentation cost was computed using the equation (A_{cost} -*fn1*$_{cost}$)/(*fn2*$_{cost}$-*fn1*$_{cost}$), where A_{cost} is the cost of a residual evaluation using the *fn2*, *nn* or *symf* stencil to compute the gradients and *fn1*$_{cost}$ is the cost of a residual evaluation using the *fn1* stencil. Using this metric the *symF* stencil can be seen to be 5.3 and 10.7 times less expensive than the *fn2* and *nn* stencils, respectively.

Table 1 Stencil statistics and relative cost of the *fn2*, *nn* and *symF* stencils for the backward facing step grid.

Stencil Augmentation Method	Minimum (min.) Stencil Size	Mean Stencil Size	Maximum (max.) Stencil Size	Stencil Standard Deviation (σ)	Relative Augmentation Cost
fn2	13	18.6	23	2.39	1
nn	13	33.8	53	6.24	2.04
symF	5	8.75	12	2.04	0.19

Figure 18 presents a contour plot of Mach no. (filled) contours and the static pressure (black lines) contours using the *fn2* stencil and is typical of the result obtained using all three stencil types. The flow solution can be seen to be nearly oscillation free with the incident shock caused by the reattachment of the separation bubble being captured without apparent difficulty. Figure 19 presents a comparison of the convergence history of the reduction of the L_2 norm of the residual for the computations performed using the *fn1*, *fn2*, *nn* and *symF* stencils, showing that the four stencils gave very similar convergence behavior with the *fn2* stencil convergence being slightly better than the other stencils. Figure 20 presents a comparison of the wall heat flux versus X and shows that the maximum difference in heat transfer between stencil types was approximately 3% at the X location where the incident shock caused by the reattachment of the separation bubble initiates.

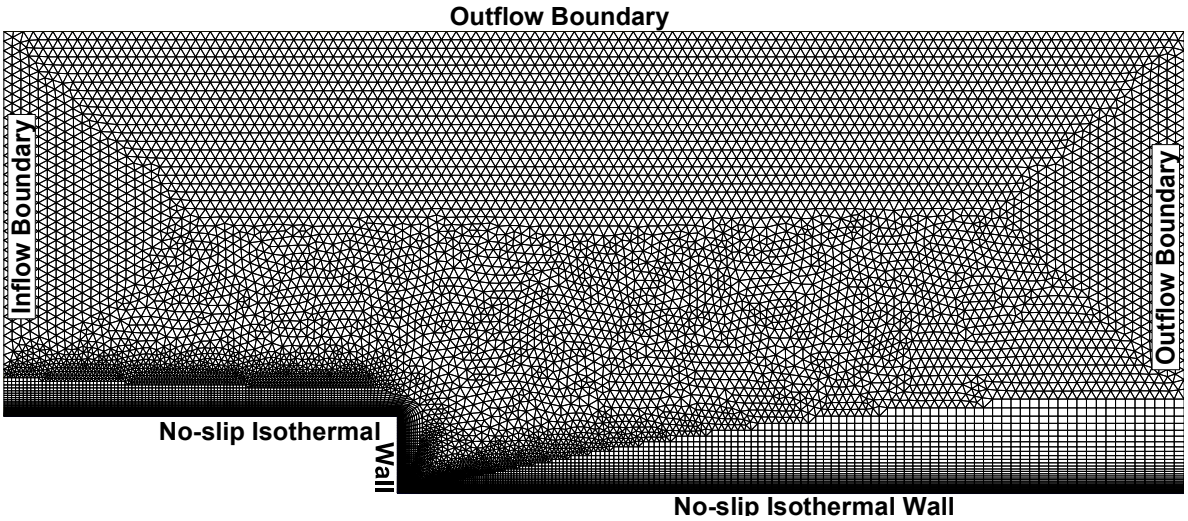


Fig. 17 Computational grid and boundary conditions for hypersonic flow over a 2-D backward facing step.

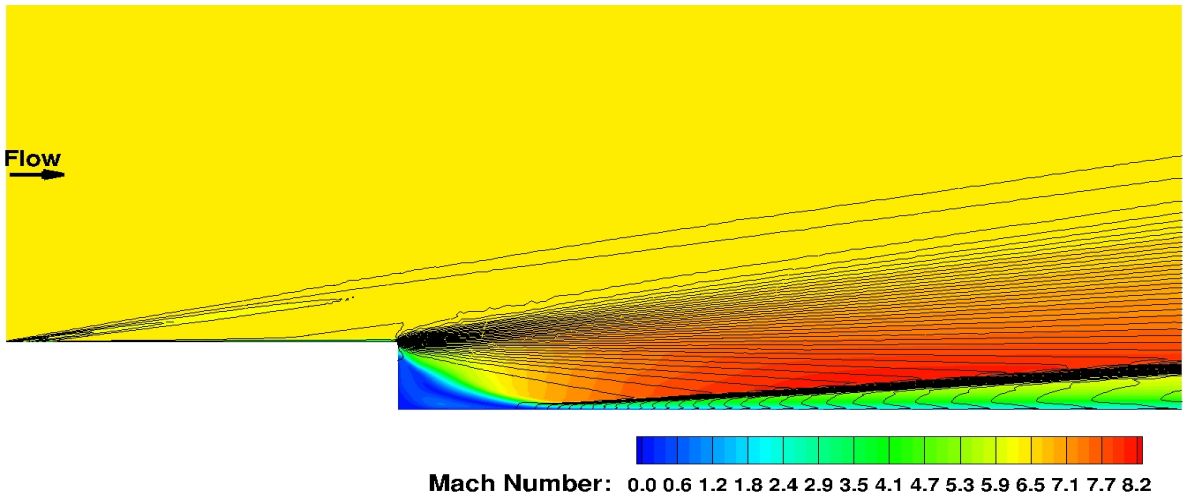


Fig. 18 Mach No. and static pressure contour plot of hypersonic flow over a 2-D backward facing step using the *fn2* stencil.

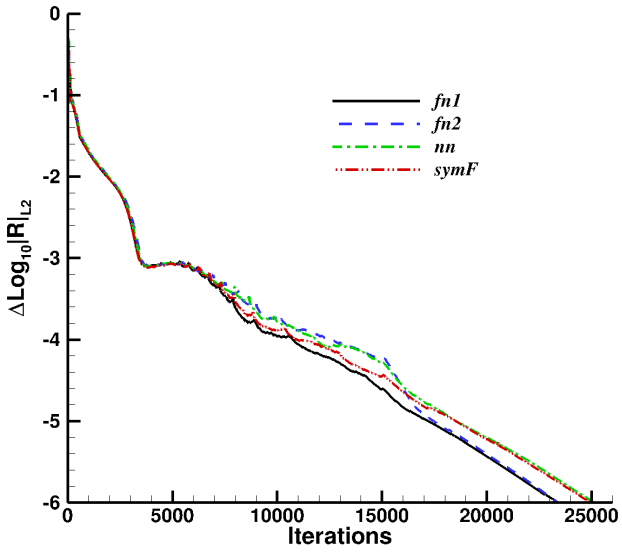


Fig. 19 A comparison of the residual convergence behavior for the computation of hypersonic flow over a 2-D backward facing step using the *fn1*, *fn2*, *nn*, and *symF* stencils.

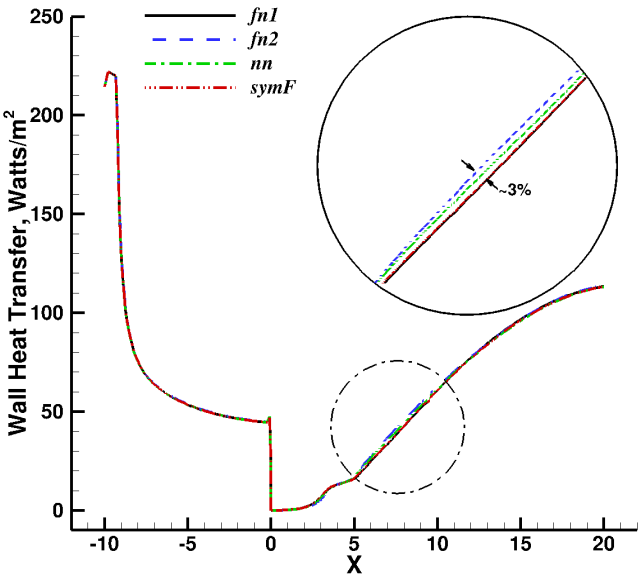


Fig. 20 A comparison of the axial distribution of wall heat transfer for the computation of hypersonic flow over a 2-D backward facing step using the *fn1*, *fn2*, *nn*, and *symF* stencils.

B. Hypersonic Turbulent Flow Over a 2-D Flat Plate Using a Grid Containing Tetrahedral, Pyramidal and Hexahedral Cells.

The second numerical experiment was conducted by computing hypersonic thermally prefect, chemically frozen, turbulent flow of air over a 2-D flat plate with freestream conditions of, Mach 6, static pressure, P_{ref}= 2100.0 Pascals, static temperature, T_{ref}= 63.01 Kelvin, and unit Reynolds number, Re_{ref}= 2.64x10^7/m, with the wall treated as an isothermal (335.83 Kelvin), no-slip, solve-to-the-wall boundary condition. The governing equations were solved in a fully coupled manner, with local time stepping and the CFL number linearly varied from 0.1 to 250 over time steps 1 to 500. The Wilcox (1998) $k-\omega$ two-equation turbulence model [36] was used to compute the Reynolds stresses and Reynolds heat flux (Pr_t=0.9), and the turbulence model production term was based on the magnitude of the vorticity. The cell-average gradients were computed using weighted linear least-squares with the *fn1*, *fn2, nn and symF* stencils. The inviscid fluxes were computed using the LDFSS scheme with the higher-order cell-face states constructed using UMUSCL, $\kappa=0$, with the cell-average gradients limited using the $\Phi_i\left(q_i^{MLP_{(g)}}\right)$ gradient limiter and the van Leer function. The viscous fluxes were computed using the Nishikawa cell-face gradient method. Convergence was achieved by "freezing" the gradient limiter after 200 time steps to prevent convergence stalling due to limiter "ringing". The computations were stopped when the residual L_2 norm had dropped 6 orders of magnitude. The 2-D geometry was discretized to form a 3-D computational domain using the Pointwise® unstructured grid generator. The resulting grid consisted of quadrilaterals on the surface of the plate that were extruded in the Y-direction to form a layer of hexahedrals in the near wall that transitioned into tetrahedral cells via a layer of pryamidal cells to form a 3-D grid of 98,928 hexahedral, 9,160 pyramidal and 65,085 tetrahedral cells for a total of 173,173 cells as shown in Fig. 21. The boundary conditions were: 1) reflection of all variables at the min. and max. Z-direction boundary cell faces (Symmetry), 2) specification of all variables on the min. X-direction boundary cell faces (Inflow Boundary), 3) 1st-order extrapolation of all variables at the max. X- and Y-direction boundary cell faces (Outflow Boundary) and 4) isothermal no-slip solve-to-the-wall on the min. Z-direction wall boundary cell faces (No-slip Isothermal Wall). The computations were performed using parallel processing on 6 partitions. A computation was performed using each stencil type with all other input parameters being unchanged. For each computation, the stencil statistics, convergence behavior, contour plots of the flow solution and the X distribution of wall heat transfer were extracted and used to compare the *fn1*, *fn2*, *nn* and *symF* stencils.

Table 2 presents the stencil statistics and the relative augmentation cost. This grid, due to its 3-D nature, and due to it consisting of hexahedral, pyramidal and tetrahedral cells has statistics such that the min. stencil size has been determined by the topology of the hexahedral cells and the max. stencil size has been determined by the connectivity of the tetrahedral cells. The min. stencil size of all 3 stencil types are similar to the 3-D extension of the stencils illustrated in Fig. 3 for a hexahedral cell. In this case, unlike the previous computations that used the wall matching no-slip boundary condition, the solve-to-the-wall no-slip boundary condition allows the inclusion of ghost cells in the no-slip boundary adjacent hexahedral cell stencils. Once again, the key point of Table 2 is that the *symF* stencil is both smaller and varies less over the computational domain than the *fn2* and *nn* stencils by a significant amount and the relative augmentation cost metric indicates that the *symF* stencil was 3.5 and 6.3 times less expensive than the *fn2* and *nn* stencils, respectively.

Table 2 Stencil statistics and relative cost for the *fn2*, *nn* and *symF* stencils for the flat plate grid.

Stencil Augmentation Type	Minimum (min,) Stencil Size	Mean Stencil Size	Maximum (max.) Stencil Size	Stencil Standard Deviation (σ)	Relative Augmentation Cost
fn2	9	19.8	27	4.52	1
nn	19	42.6	107	20.3	1.82
symF	6	9.41	21	4.07	0.29

Figure 22 presents a contour plot of Mach no. contours using the *nn* stencil and is typical of the results obtained using all three stencil types. The flow solution can be seen to be nearly oscillation free with the weak leading edge shock caused by the rapid growth in the displacement thickness during the initial boundary layer formation being preserved even in the tetrahedral cell part of the grid. Figure 23 presents a comparison of the convergence history of the reduction of the L_2 norm of the residual for the *fn1*, *fn2*, *nn* and *symF* computations showing that the stencils gave nearly identical convergence behavior. Figure 24 presents a comparison of the wall heat flux versus X and shows that all stencil methods produced very similar results and that the maximum difference in heat transfer between stencil types was approximately 1.44% at the trailing edge of the plate.

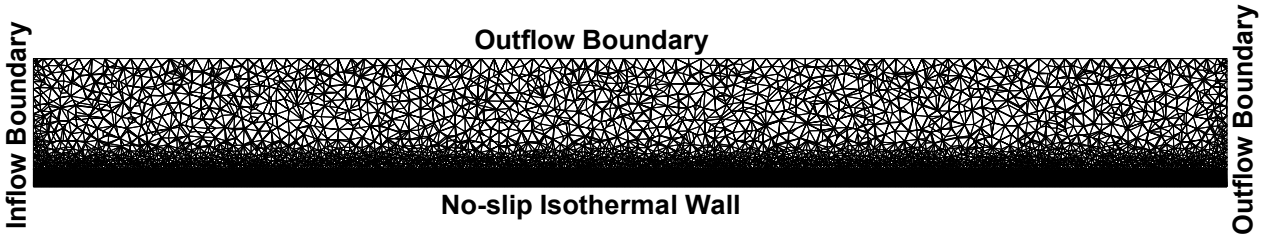


Fig. 21 Computational grid and boundary conditions for hypersonic flow over a flat plate.

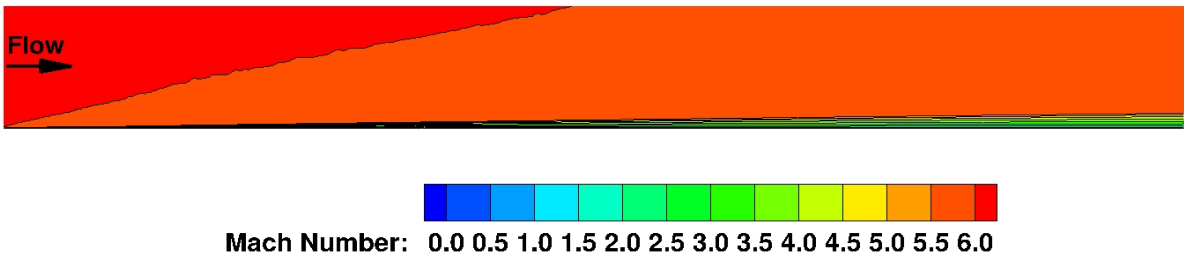


Fig. 22 Mach no. contour plot of hypersonic flow over a 2-D backward facing step using the *nn* stencil.

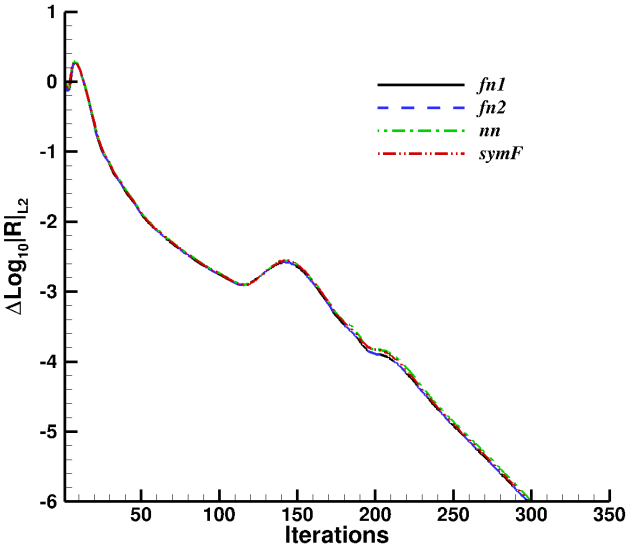


Fig. 23 A comparison of the residual convergence behavior for the computation of hypersonic flow over a 2-D flat plate using the *fn1*, *fn2*, *nn*, and *symF* stencils.

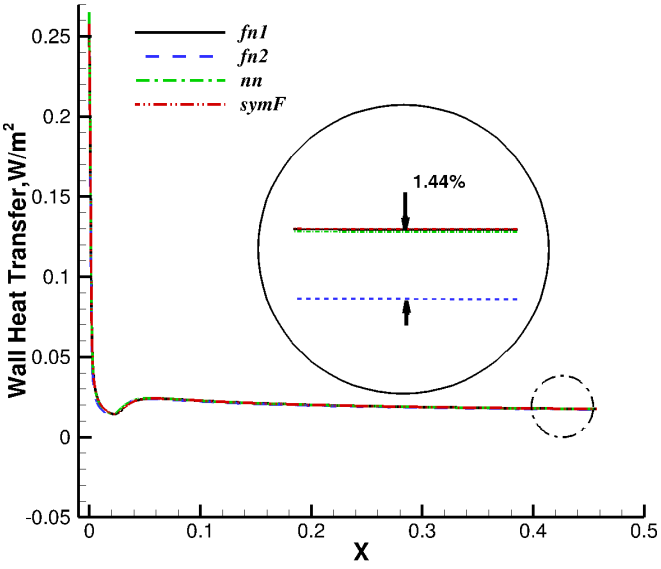


Fig. 24 A comparison of the axial distribution of wall heat transfer for the computation of hypersonic flow over a 2-D flat plate using the *fn1*, *fn2*, *nn*, and *symF* stencils.

C. Hypersonic 2-D Turbulent Flow Over a 2-D Blunt Wedge Using a Grid Containing Tetrahedral, Pyramidal, Prismatic and Hexahedral Cells.

The third numerical experiment was conducted by computing hypersonic, thermally prefect, chemically frozen, turbulent flow over a blunt wedge representing a scramjet flowpath forebody or cowl leading edge (this particular geometry uses the same diameter as the REST inlet cowl presented in [37]). The freestream conditions were: P_{ref}= 1675.0 Pascals, T_{ref}= 226.7 Kelvin, and Mach number, M_{ref}= 8. A thermally perfect air gas mixture was used to simulate the test gas, which at the given conditions yields a unit Reynolds number of Re_{ref}= 2.9467x10^6/m. The wall surface was treated as a no-slip, isothermal (300.0 Kelvin) wall, using a turbulent wall matching boundary condition. The Menter Baseline two-equation turbulence model [38] was used to compute the Reynolds stresses and Reynolds heat flux (Pr_t=0.9), and the turbulence model production term was based on the magnitude of the vorticity. The inviscid fluxes were computed using the LDFSS scheme with the higher-order cell-face states reconstructed using the UMUSCL, $\kappa = 0$, scheme with the cell-average gradients limited using the $\Phi_i(q_i^{MLP_{(g_i)}})$ gradient limiter and the van Leer function. The viscous fluxes were computed using the Nishikawa cell-face gradient method. Convergence was achieved by "freezing" the gradient limiter after 2000 time steps to prevent convergence stalling due to limiter "ringing". The governing equations were solved implicitly using the SGS scheme, with local time stepping and the CFL number linearly varied from 0.1 to 10 over time steps 1 to 1000. The computations were stopped when the residual L_2 norm had dropped 6 orders of magnitude relative to its initial value. The blunt wedge geometry was discretized to form a 3-D computational domain using the Pointwise® unstructured grid generator. The resulting grid consisted of quadrilaterals on the surface of the wedge that were extruded normal to the surface to form layers of hexahedral cells in the near wall, which were transitioned to tetrahedral cells via a layer of pryamidal cells to form a 3-D grid of 364,380 hexahedral, 14,840 prismatic, 67,519 pyramidal and 647,424 tetrahedral cells for a total of 1,094,163 cells presented in Figs. 25 and 26. The boundary conditions were: 1) reflection of all variables at the min. and max. Y-direction boundary cell faces (Symmetry Boundaries), 2) specification of all variables on the parabolic shaped surface boundary cell faces (Inflow Boundary), 3) 1st-order extrapolation of all variables at the max. X-direction boundary cell faces (Outflow Boundary) and 4) isothermal wall-matching on the wall boundary cell faces (No-slip Isothermal Wall Boundary). The computations were performed using parallel processing on 24 partitions. A computation was performed using each stencil stencil type with all other input parameters being unchanged. For each computation, the stencil statistics, convergence behavior, contour plots of the flow solution and the X distribution of wall heat transfer were extracted and used to compare the *fn2*, *nn* and *symF* stencils. However, results were not obtained for the *fn1* stencil because it was found to be unstable.

Table 3 presents the stencil statistics and the relative augmentation cost. This grid, due to its 3-D nature, and due to it consisting of hexahedral, tetrahedral, pyramidal and tetrahedral cells has statistics such that the min. stencil size has been determined by the topology of the hexahedral cells and the max. stencil size has been determined by the connectivity of the tetrahedral cells. The min. stencil size of all 3 stencil types are smaller than expected based on a 3-D extension of Fig. 3 for a hexahedral cell due the same boundary effect described previously. Again, the key point of Table 3 is that the *symF* stencil is both smaller and varies less over the computational domain than the *fn2* and *nn* stencils by a significant amount, and the relative augmentation cost metric indicates that the *symF* stencil was 3.5 and 6.3 times less expensive than the *fn2* and *nn* stencils, respectively.

Table 3 Stencil statistics and relative cost of the *fn2*, *nn* and *symF* stencils for the blunt wedge grid.

Stencil Augmentation Type	Minimum (min.) Stencil Size	Mean Stencil Size	Maximum (max.) Stencil Size	Stencil Standard Deviation (σ)	Relative Augmentation Cost
fn2	13	18.6	32	4.39	1
nn	8	54.5	133	22.2	2.91
symF	5	11.1	24	3.82	0.55

Figures 27 and 28 present contour plots of Mach using the *symF* stencil and are typical of the results obtained using all three stencil types. Figure 27 shows that the flow solution can be seen to be nearly oscillation free with the bow shock being well captured. Figure 28, which presents a close up of the blunt body bow shock on the cutting plane described in Fig. 25, is carbuncle free and well captured in approximately three cells in the hexahedral cell part of the grid in the vicinity of the stagnation streamline. Figure 29 presents a comparison of the convergence history of the reduction of the L_2 norm of the residual for computations using the three stencils, showing that once the limiter was frozen the stencils gave nearly identical convergence behavior. Figure 30 presents a comparison of the normalized wall heat flux versus normalized X-distance showing that, in this case, all three stencil methods produced results that are similar in shape. However, the *nn* stencil results are approximately 12% higher than the *fn2* and *symF* results and the maximum difference in heat transfer between the *fn2* and *symF* stencil types was approximately 2% near the outflow boundary. The cause of this disagreement is not known at this time but may be related to the high aspect ratio and high streamwise curvature nature of the grid and its affect on least square gradients reported in Ref. [10,13,32].

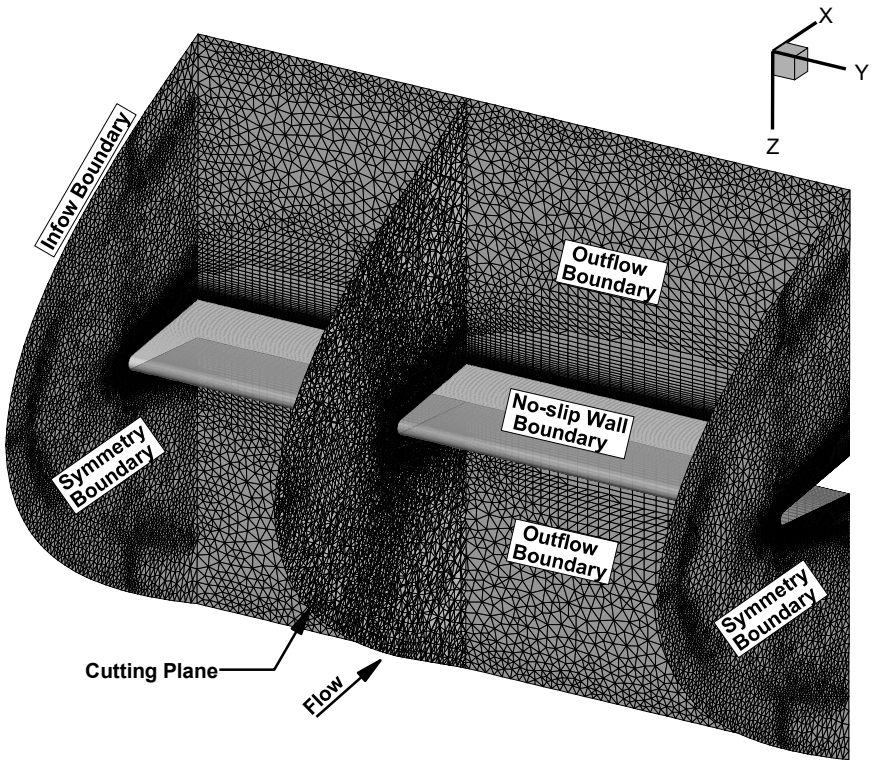


Fig. 25 3-D computational grid and boundary conditions for hypersonic flow over a 2-D blunt wedge.

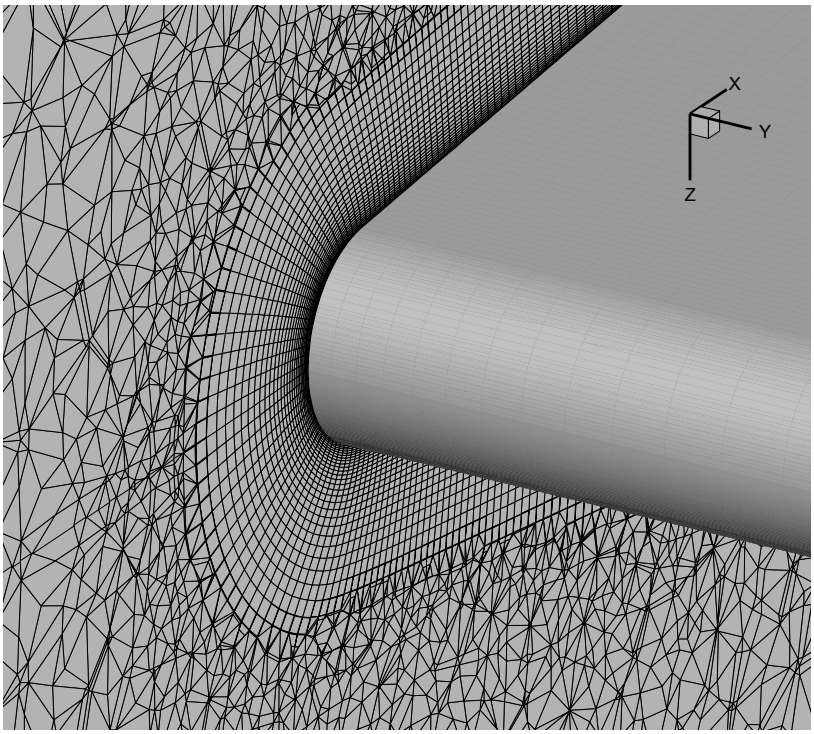


Fig. 26 Close up view of the leading edge grid on cutting plane for hypersonic flow over a 2-D blunt wedge.

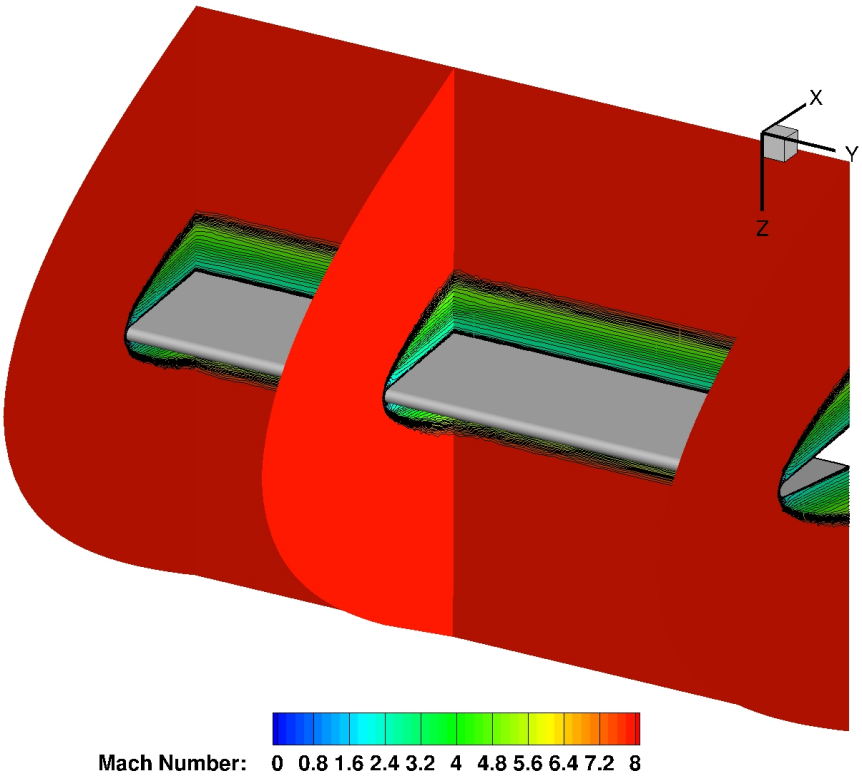


Fig. 27 Mach number contours for hypersonic flow over a 2-D blunt wedge step computed using the *symF* stencil.

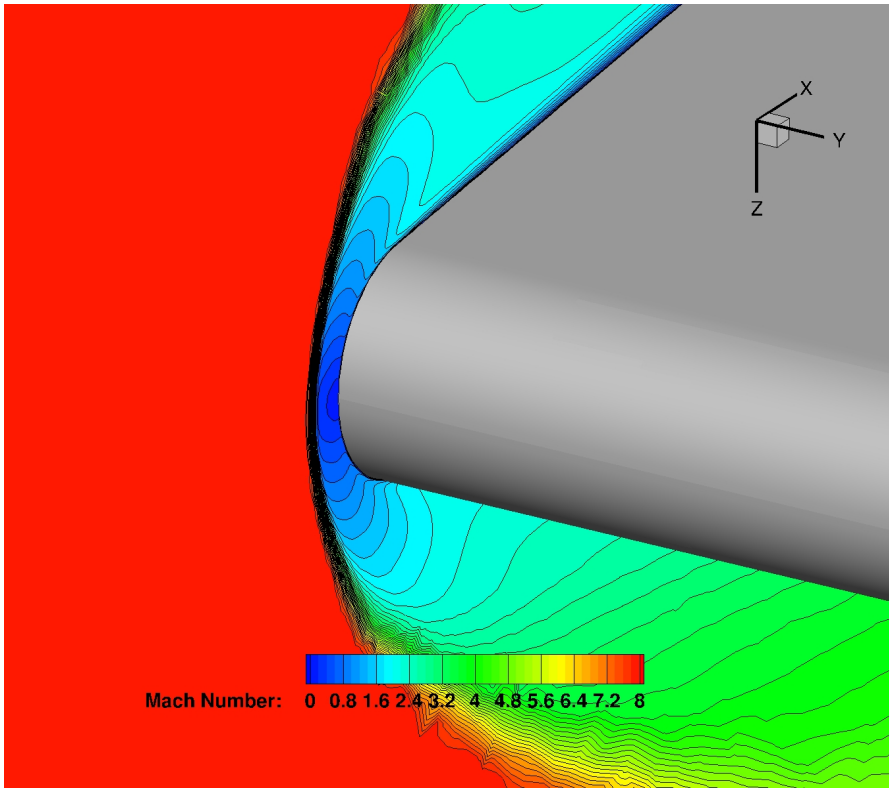


Fig. 28 Close up of leading edge Mach number contours for hypersonic flow over a 2-D blunt wedge computed using the *symF* stencil.

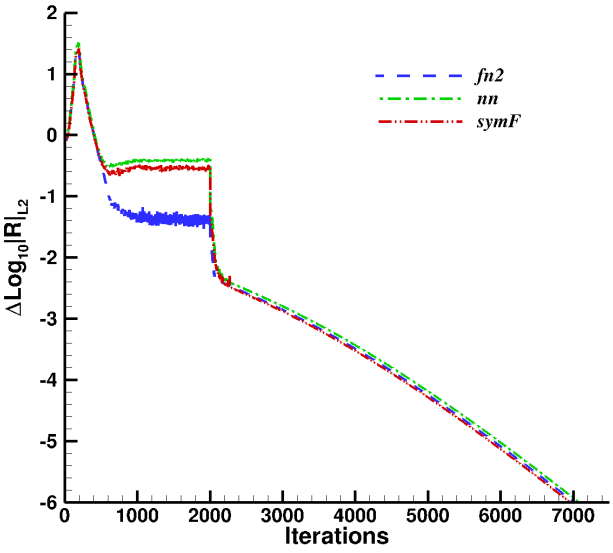


Fig. 29 A comparison of the residual convergence behavior for the computation of hypersonic flow over a 2-D blunt wedge using the *fn2*, *nn*, and *symF* stencils.

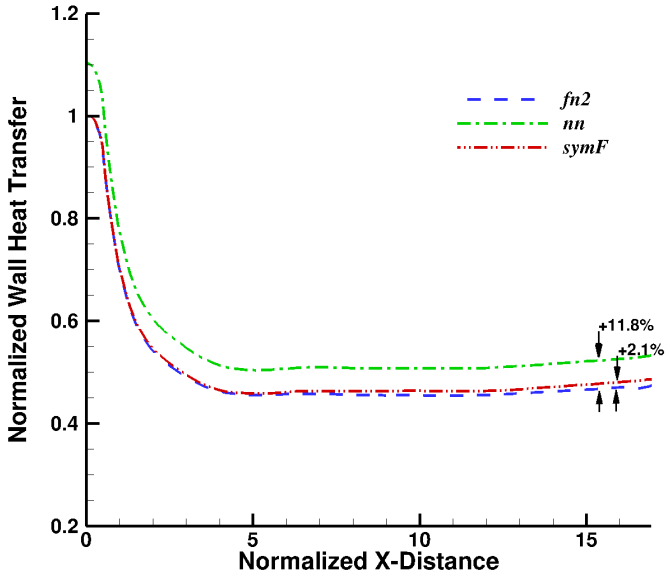


Fig. 30 A comparison of the axial distribution of wall heat transfer for the computation of hypersonic flow over a 2-D blunt wedge using the *fn2*, *nn*, and *symF* stencils.

V. Summary And Conclusions

The methods used in the VULCAN-CFD code to construct the cell-average and cell face gradients were described. Particular attention was paid to the weighted linear least-squares methods for the construction of cell-average gradients. Multiple least-squares stencil construction methodologies were investigated, compared and evaluated. Three of the stencil construction methods were based on the current state of the art, the *fn1*, *fn2* and *nn* methods and a fourth method, *symF*, is based on a modification of a Frobenius-norm minimization-based approach recently introduced by Nishikawa [9]. In addition, a modified form of the MLP cell-average gradient limiter method of Park and Kim [25] that can be applied to a general stencil was introduced. The extension of Nishikawa's *symF* method to 3-D turbulent viscous flow for use in the VULCAN-CFD code was described in detail. The modifications to Nishikawa's *symF* stencil construction method were based on 1) the results obtained from a cell-average gradient accuracy verification study conducted in 2-D using severely distorted grids and 2) inherent implementation constraints currently in the VULCAN-CFD code. The accuracy of the *fn2*, *nn* and modified *symF* stencil construction methods were compared against analytical results for a 2-D canonical problem using severely distorted quadrilateral and triangular cell grids. This verification effort revealed that the *symF* method produced lower accuracy results on the distorted quadrilateral grids. This conclusion, and the aforementioned implementation constraints, led to a modification of the construction methodology proposed by Nishikawa where the *fn1* stencil is used for hexahedral cells and a *symF* stencil, obtained by augmenting the *fn1* stencil using the set of cells belonging to the *nn* stencil using the *symF* minimization process, is used for prismatic, pyramidal and tetrahedral cells.

The robustness and accuracy of the fn1, *fn2*, *nn* and modified *symF* stencil construction methods for viscous flow were then examined by computing three canonical turbulent hypersonic flows with each of the stencils. The stencil size statistics, convergence behavior and wall heat transfer of the three flows for each of the stencil methods were examined and compared. The modified *symF* stencil was found to give stencils having the fewest member cells and to produce the most uniform stencil distribution on fully 3-D unstructured grids, resulting in the modified *symF* stencil being the least expensive method by at least a factor of 3 compared to the *nn* stencil for the blunt wedge case. Moreover, the modified *symF* stencil was found to give similar convergence behavior for all three flows, and was not found to exhibit any robustness issues. In agreement with the literature, the *fn1* stencil was found to sometimes exhibit instabilities in 3-D when using grids containing tetrahedral cells. The *fn1* stencil was found to be stable for the backward facing step and flat plate numerical experiments but it was found to be unstable for the blunt wedge numerical experiment. Moreover, the wall heat transfer comparison for the blunt wedge numerical experiment showed that the *nn* stencil produced heat transfer results that were approximately 12% higher than the *fn2* and *symF* stencils. The root cause of this difference is not currently understood but may be due to the high aspect ratio high curvature nature of the grid around the blunt body portion of the grid and will be the subject of further study. In addition, the robustness and accuracy of the *fn2*, *nn* and *symF* construction methods will be explored further for fully 3-D flow through typical scramjet flowpath components.

Acknowledgments

This work was supported by the Hypersonic Technology Project, through the Hypersonic Airbreathing Propulsion Branch of the NASA Langley Research Center. In addition, we would like to thank Dr. Boris Diskin, of the National Institute of Aerospace, for his many helpful suggestions.

References

1. Gnoffo, P. A., Wood, W. A., Kleb, B., Alter, S. J., Glass, C., Padilla J., Hammond, D. and White, J. A., "Functional Equivalence Acceptance Testing of FUN3D for Entry, Descent, and Landing Applications," *AIAA Paper 2013-2558*, June 2013.

2. Nompelis, I., Drayna, T. and Candler, G., "Development of a Hybrid Unstructured Implicit Solver for the Simulation of Reacting Flows Over Complex Geometries," *AIAA Paper 2004-2227*, June 2004.

3. Luke, E, "On Robust and Accurate Arbitrary Polytope CFD Solvers," *AIAA Paper 2007-3956*, June 2007.

4. Goldberg, U., "Hypersonic Turbulent Flow Predictions Using CFD++," *AIAA Paper 2005-3214*, May 2005.

5. White, J. A. and Morrison, J. H., "A Psuedo-Temporal Multi-Grid Relaxation Scheme for Solving the Parabolized Navier-Stokes Equations," *AIAA Paper 1999-3360*, June 1999.

6. Litton, D. K., Edwards J. R., and White, J. A., "Algorithmic Enhancements to the VULCAN Navier-Stokes Solver," *AIAA Paper 2003-3979*, June 2003.

7. Spiegel, S. C., Stefanski, D. L., Luo, H., and Edwards, J. R., "A Cell-Centered Finite Volume Method for Chemically Reacting Flows on Hybrid Grids," *AIAA Paper 2010-1083*, July 2010.

8. Spiegel, S. C., Stefanski, D. L., Luo, H., and Edwards, J. R., "A Regionally Structured/Unstructured Finite Volume Method for Chemically Reacting Flows," *AIAA Paper 2011-3048*, July 2011.

9. White, J. A., Baurle, R. A., Passe, B. J., Spiegel, S. C. and Nishikawa, H., "Geometrically Flexible and Efficient Flow Analysis of High Speed Vehicles Via Domain Decomposition, Part 1, Unstructured-grid Solver for High Speed Flows," JANNAF 48th CS / 36th APS / 36th EPSS / 30th PSHS Joint Subcommittee / Programmatic and Industrial Base Meeting, Newport News, VA, Dec., 2017.

10. Nishikawa, H., "Efficient Gradient Stencils for Robust Implicit Finite-Volume Solver Convergence on Distorted Grids," *Journal of Computational Physics*, submitted for publication, Aug., 2018.

11. Schwöppe, A. and Diskin, B., "Accuracy of the Cell-Centered Grid Metric in the DLR TAU-Code," *New Results in Numerical and Experimental Fluid Mechanics VIII. Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, Vol. 121, 2013, Springer Berlin Heidelberg, pp. 429-437.

12. Sozer, E., Brehm, C. and Kiris, C. C., "Gradient Calculation Methods on Arbitrary Polyhedral Unstructured Meshes for Cell-Centered CFD Solvers," *AIAA Paper 2014-1440*, 52nd AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Jan. 2014.

13. Diskin, B. and Thomas, J. L., "Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discetizations: Inviscid Fluxes," *AIAA Paper 2010-1079*, Jan. 2010.

14. Haider, F., Croiselle, J.-P., and Courbet, B., "Stability Analysis of the Cell Centered Finite-volume MUSCL Method on Unstructured Grids," *Numerische Mathematik*, Vol. 113, 2009, Springer Berlin Heidelberg, pp. 555-600.

15. Zangeneh, R. and Ollivier-Gooch, C. F., "Reconstruction Map Stability Analysis for Cell Centered Finite Volume Methods on Unstructured Meshes", *AIAA Paper 2017-0734*, Jan. 2017.

16. Strang, G., *Linear Algebra and Its Applications*. Academic Press, 2nd ed., 1980.

17. Cary, A. W., Dorgan, A. J. , and Mani, M., "Towards Accurate Flow Predictions Using Unstructured Meshes," 39th AIAA Fluid Dynamics Conference, AIAA Paper 2009-3650, June, 2009.

18. Edwards, J. R., "A Low-diffusion Flux-splitting Scheme for Navier-Stokes Calculations," *Computers & Fluids,* Vol. 26, No. 6, 1997, pp. 635–659.

19. Toro, E. F., Spruce, M., and Speares, W., "Restoration of the Contact Surface in the HLL-Riemann Solver," *Shock Waves,* Vol. 4, 1994, pp. 25–34.

20. Fromm, J. E. , "A Method for Reducing Dispersion in Convective Difference Schemes," *Journal of Computational Physics*, Vol. 3, 1968, pp. 176–189.

21. Burg, C. O. E., "Higher-Order Variable Extrapolation for Unstructured Finite Volume RANS Flow Solvers," *AIAA Paper 2005-4999*, June 2005.

22. van Leer, B., "Towards the Ultimate Conservative Difference Scheme. V. A Second-order Sequel to Godunov's Method," *Journal of Computational Physics,* Vol. 32, 1979, pp. 101–136.

23. Barth, T. J. and Jespersen, D., "The Design and Application of Upwind Schemes on Unstructured Meshes," *AIAA Paper 1989-0366*, Jan. 1989.

24. Venkatakrishnan, V., "On the Accuracy of Limiters and Convergence to Steady State Solutions," *AIAA Paper 1993-0880*, Jan. 1993.

25. Park, J. S. and Kim, C., "Multi-dimensional Limiting Process for Finite Volume Methods on Unstructured Grids," *Computers & Fluids*, Vol. 65, Elsevier Ltd, 2012, pp. 8–24.

26. van Albada, G. D., van Leer, B., and Roberts, W. W., "A Comparative Study of Computational Methods in Cosmic Gas Dynamics," *Astronomy and Astrophysics*, Vol. 108, 1982, pp. 76–84.

27. Sweby, P. K. "High Resolution Schemes Using Flux-limiters for Hyperbolic Conservation Laws," *SIAM Journal of Numerical Analysis*, Vol. 21, 1984, pp. 995–1011.

28. Koren, B., “A Robust Upwind Discretization Method for Advection, Diffusion and Source Terms,” *Numerical Methods for Advection–Diffusion Problems*, Braunschweig: Vieweg, 1993, p. 117.

29. Gnoffo, P. A., “Updates to Multi-Dimensional Flux Reconstruction for Hypersonic Simulations on Tetrahedral Grids,” *AIAA Paper 2010-1271*, Jan. 2010.

30. Hasselbacher, A. and Blazek, J., “On the Accurate and Efficient Discretization of the Navier-Stokes Equations on Mixed Grids,” *AIAA Paper 1998-0612*, Jan. 1998.

31. Nishikawa, H., “Beyond Interface Gradient: A General Principle for Constructing Diffusion Schemes,” *AIAA Paper 2010-5093*, June 2010.

32. Diskin, B., Thomas, J. L., Nielsen, E., Nishikawa, H. and White, J. A., “Comparison of Node-Centered and Cell-Centered Unstructured Finite Volume Discetizations: Viscous Fluxes,” *AIAA Journal*, Vol. 48, No. 7, 2010, pp. 1326–1339.

33. Jalai, A., Sharbtdar, M. and Ollivier-Gooch C., “Accuracy Analysis of Unstructured Finite Volume Discretization Schemes For Diffusive Fluxes,” *Computers & Fluids*, Vol. 101, 2014, pp. 220–232.

34. Wilcox, D. C., “Wall Matching, a Rational Alternative to Wall Functions”, *AIAA paper 89-0611*, Jan. 1989.

35. Wilcox, D. C., "Formulation of the k-omega Turbulence Model Revisited," *AIAA Journal*, Vol. 46, No. 11, 2008, pp. 2823-2838.

36. Wilcox, D. C., *Turbulence Modeling for CFD*, 2nd ed, DCW Industries, Inc., La Canada, CA, 1998, pp. 121-122.

37. Chan, Y. K., Razzaqi, S. Smart, M. K. and Wise, D., “Freejet Testing of the 75%-scale HIFiRE 7 REST Scramjet Engine,” *AIAA Paper 2014-2931*, June 2014.

38. Menter, F. R., “Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications”, *AIAA Journal,* Vol. 32, No. 8, 1994, pp. 1598–160.